Distributed Optimization of Nonconvex Functions over Time-Varying Graphs ⋆

Bryan Van Scoy [*] **Laurent Lessard** [*,**]

[*] *Wisconsin Institute for Discovery*
[**] *Department of Electrical Engineering*
University of Wisconsin–Madison, Madison, WI 53706, USA
{vanscoy,laurent.lessard}@wisc.edu

Abstract: We consider the distributed optimization problem where a group of agents seeks to cooperatively compute the optimizer of the average of local functions over a time-varying directed communication network. To solve this problem, we propose a novel algorithm which adjusts the ratio between the number of communications and computations to achieve fast convergence. In particular, the iterates of our algorithm converge to the optimizer at the same rate as those of centralized gradient descent in terms of the number of computations. We compare our algorithm with other known algorithms on a distributed target localization problem.

Keywords: Distributed optimization, nonconvex, time-varying, fixed-point iteration, gossip, decentralized, gradient method

1. INTRODUCTION

We consider the distributed optimization problem

$$\underset{x\in\mathbb{R}^d}{\text{minimize}} \;\; f(x) \quad \text{where} \quad f(x) := \frac{1}{n}\sum_{i=1}^{n} f_i(x). \qquad (1)$$

Associated with each agent $i \in \{1,\dots,n\}$ is the local objective function $f_i : \mathbb{R}^d \to \mathbb{R}$ where n is the number of agents and d is the dimension of the problem. The goal is for the agents to calculate the global optimizer using only local communications and computations. This problem has received significant attention recently due to its numerous applications in distributed machine learning, distributed estimation, and resource allocation.

Many new algorithms have been proposed to solve the distributed optimization problem. Some examples include distributed gradient descent by Nedić and Ozdaglar (2009), EXTRA by Shi et al. (2015), AugDGM by Xu et al. (2015), NIDS by Li et al. (2017), DIGing by Nedić et al. (2017) and Qu and Li (2018), and Exact Diffusion by Yuan et al. (2019) among others. For each of these algorithms, every agent does the following at each iteration:

(i) communicate state variables with local neighbors,
(ii) compute the local gradient (i.e., evaluate ∇f_i),
(iii) update local state variables.

Additionally, there have been recent efforts to unify the design and analysis of such algorithms. For example, Jakovetić (2019) provides a unifying view of EXTRA and DIGing, while Sundararajan et al. (2018) construct a canonical form for any method where agents have two local state variables. Furthermore, Sundararajan et al. (2017) provide a systematic approach for proving linear convergence rates of such algorithms.

In each of the algorithms mentioned, agents perform a single communication and computation at each iteration. This approach, however, inefficiently utilizes the available resources in some situations. For example, when the communication network is sparse and the objective function is well-conditioned, one expects the algorithm to need significantly more communications than computations. In this case, the algorithm should exploit this by increasing the ratio between the number of communications and computations in order to effectively use the available resources. Such an approach has been used recently by Scaman et al. (2017) to construct an algorithm with optimal complexity in the case when the communication network is fixed and the objective function of each agent is smooth and strongly convex. Here, the iteration complexity is measured in terms of the time it takes for the algorithm to obtain a solution with a given precision, where both communication and computation are assumed to require a specified amount of time.

Main contributions. In this work, we propose a novel decentralized algorithm for solving (1). Similar to that of Scaman et al. (2017), our algorithm sets the ratio between the number of communications and computations so that communications and computations are used effectively. We do so, however, while making weak assumptions on both the local objective functions and the communication network. In particular, our algorithm has the following properties:

- The worst-case convergence rate in terms of number of computations (evaluations of $\{\nabla f_i\}$) is identical to that of centralized gradient descent.
- We only require the local objective functions to be *one-point convex* with respect to the global optimizer.
- The communication network may be both directed and time-varying as long as it is sufficiently connected at each time step.

We prove linear convergence of our algorithm and provide the corresponding rate. A surprising result is that the rate of the decentralized algorithm is actually the same as that of centralized gradient descent. While this could be achieved with a large amount of communication at each iteration (so that every agent obtains the iterates of every other agent at each iteration), we show that this rate can also be achieved with significantly less communication. In particular, there is a certain amount of mixing among agents which must be done at each iteration so that this rate is achieved, and our algorithm uses the least amount of communication possible to achieve this.

The remainder of the paper is organized as follows. We first setup the distributed optimization problem along with our assumptions in Section 2, and then present our algorithm along with its main convergence result in Section 3. We then compare our algorithm with several others on a distributed target localization problem in Section 4, and conclude in Section 5. To simplify the presentation, we defer the main convergence proof to Appendix A.

Notation. We use subscript i to denote the agent and superscript k to denote the iteration. We denote the all-ones vector by $\mathbb{1} \in \mathbb{R}^n$, the identity matrix by $I_n \in \mathbb{R}^{n\times n}$, and the 2-norm by $\|\cdot\|$.

2. PROBLEM SETUP

Consider a network of n agents where a local function $f_i : \mathbb{R}^d \to \mathbb{R}^d$ is associated with each agent $i \in \{1,\dots,n\}$. The goal is for each agent to compute the minimizer of the average of the local function by communicating with neighboring agents and performing local computations.

Similar to the distributed optimization problem is the *distributed fixed-point problem*. Here, agent $i \in \{1,\dots,n\}$ has a local operator $T_i : \mathbb{R}^d \to \mathbb{R}^d$ and the goal is for the agents to compute a fixed-point of the average operator

$$T := \frac{1}{n}\sum_{i=1}^{n} T_i. \tag{2}$$

This problem has been studied for the past several decades, at least since the work of Bertsekas (1983) on asynchronous distributed fixed-point iterations.

Suppose we define the local operators as $T_i := I - \alpha\nabla f_i$ where $\alpha > 0$ is the stepsize. Then any minimizer $x^\star \in \mathbb{R}^d$ of the distributed optimization problem (1) satisfies $\nabla f(x^\star) = 0$ and is therefore also a fixed-point of the average operator T in (2). In other words, fixed-points of T correspond to first-order stationary points of f. Throughout the rest of the paper, we consider the distributed fixed-point problem to both simplify notation and highlight the properties of our algorithm with respect to the contraction factor of T.

To compute the fixed-point, it is well-known that if T is contractive, then the sequence generated by the *fixed-point iteration*

$$x^{k+1} = T(x^k) \tag{3}$$

for $k \geq 0$ converges to the unique fixed-point by the Banach Fixed-Point Theorem. Note that this is equivalent to (centralized) gradient descent when $T = I - \alpha\nabla f$. Implementing (3) directly, however, requires agents to exchange their information with every other agent between each iteration which is computationally expensive when the network is large.

To solve the distributed fixed-point problem, we make the following assumptions on the set of fixed-point operators $\{T_i\}$ and the communication network.

2.1 Fixed-point operators

Assumption 1. (Fixed-point). There exists a fixed-point $x^\star \in \mathbb{R}^d$ of the average operator T in (2), in other words,

$$x^\star = \frac{1}{n}\sum_{i=1}^{n} T_i(x^\star). \tag{4}$$

Furthermore, there exists a scalar $\rho \in (0,1)$, called the *contraction factor*, such that

$$\|T_i(x) - T_i(x^\star)\| \leq \rho\,\|x - x^\star\| \tag{5}$$

for all $x \in \mathbb{R}^d$ and all $i \in \{1,\dots,n\}$.

Note that this assumption implies that

$$\|T(x) - T(x^\star)\| \leq \rho\,\|x - x^\star\|$$

for all $x \in \mathbb{R}^d$. In other words, the global operator T is a contraction with respect to the fixed-point $x^\star$ with contraction factor ρ. Since the fixed-point $x^\star$ is not known a priori, it may be difficult to verify (5). In this case, a useful sufficient condition is that each T_i is a *contraction* with contraction factor ρ, meaning that

$$\|T_i(x) - T_i(y)\| \leq \rho\,\|x - y\|$$

for all $x, y \in \mathbb{R}^d$.

2.2 Communication network

To characterize the communication among agents, we use a gossip matrix defined as follows.

Definition 1. (Gossip matrix). We say that a matrix $W = \{w_{ij}\} \in \mathbb{R}^{n\times n}$ is a *gossip matrix* if $w_{ij} = 0$ whenever agent i does not receive information from agent j. We define the *spectral gap* $\sigma \in \mathbb{R}$ of a gossip matrix W as

$$\sigma := \|W - \tfrac{1}{n}\mathbb{1}\mathbb{1}^\mathsf{T}\|. \tag{6}$$

Furthermore, we say that W is *row-stochastic* if $W\mathbb{1} = \mathbb{1}$ and *column-stochastic* if $\mathbb{1}^\mathsf{T}W = \mathbb{1}^\mathsf{T}$. Finally, we say W is *doubly-stochastic* if it is both row- and column-stochastic.

One way to obtain a gossip matrix is to set $W = I - L$ where L is the (possibly weighted) graph Laplacian. We make the following assumption on the gossip matrix.

Assumption 2. (Communication network). There exists a scalar $\sigma \in (0,1)$ such that each agent $i \in \{1,\dots,n\}$ has access to the i^{th} row of a doubly-stochastic gossip matrix W with spectral gap σ at each iteration of the algorithm.

This assumption allows the communication network to be both directed and time-varying. At each iteration, however, the gossip matrix must be doubly stochastic with a known upper bound on its spectral gap. See Xiao et al. (2007) for how to optimize the weights of the gossip matrix to minimize the spectral gap.

As we will see, the convergence properties of our algorithm depend on the contraction factor ρ and spectral gap σ.

3. MAIN RESULTS

We now introduce our algorithm for solving the distributed fixed-point problem.

Algorithm

Parameters: $\rho, \sigma \in (0,1)$

Inputs: Fixed-point operator $T_i : \mathbb{R}^d \to \mathbb{R}^d$ on agent $i \in \{1,\dots,n\}$, gossip matrices $\{w_{ij}^{k\ell}\}$ at iteration k and communication round ℓ.

Initialization:

- Each agent $i \in \{1,\dots,n\}$ chooses $x_i^0, y_i^0 \in \mathbb{R}^d$ such that $\sum_{i=1}^n y_i^0 = 0$ (for example, $y_i^0 = 0$).
- Define the number of communications per iteration

$$m := \underset{r \geq \rho,\, s \geq \sigma}{\text{minimize}} \left\lceil \log_s\left(\tfrac{\sqrt{1+r}-\sqrt{1-r}}{2}\right)\right\rceil$$

as well as the parameter $\lambda := \sqrt{1-\rho^2}$.

for iteration $k = 0, 1, 2, \dots$ **do**
  for agent $i \in \{1,\dots,n\}$ **do**
    $v_{i,0}^k = x_i^k$
    for communication round $\ell = 1,\dots,m$ **do**
      $v_{i,\ell}^k = \sum_{j=1}^n w_{ij}^{k\ell}\, v_{j,\ell-1}^k$ (local communication)
    end for
    $u_i^k = T_i(v_{i,m}^k)$ (local computation)
    $y_i^{k+1} = y_i^k + x_i^k - v_{i,m}^k$ (local state update)
    $x_i^{k+1} = u_i^k - \lambda\, y_i^{k+1}$ (local state update)
  end for
end for
return $x_i^k \in \mathbb{R}^d$ is the estimate of $x^\star$ on agent i at iteration k

At each iteration of the algorithm, each agent $i \in \{1,\dots,n\}$ first communicates with its local neighbors m times using the gossip matrices $\{W^{k,\ell}\}_{\ell=1}^m$, then computes their local fixed-point operator T_i at the point resulting from the communication, and finally updates its local state variables x_i^k and y_i^k. The output of the algorithm is x_i^k which is the estimate of the fixed-point $x^\star$ of the global operator T. Note that agents are required to know the global parameters ρ and σ so that they can calculate the number of communication rounds m and the parameter λ.

For a given contraction factor ρ and spectral gap σ, agents perform m consecutive rounds of communication at each iteration where

$$m := \underset{r \geq \rho,\, s \geq \sigma}{\text{minimize}} \left\lceil \log_s\left(\tfrac{\sqrt{1+r}-\sqrt{1-r}}{2}\right)\right\rceil. \tag{7}$$

Since only one computation is performed per iteration, this adjusts the ratio between the number of communications and computations as shown in Figure 1. In particular, the algorithm uses a single communication per computation when the network is sufficiently connected (σ small) and the fixed-point operator is ill-conditioned (ρ large). As the network becomes more disconnected and/or the fixed-point operator becomes more well-conditioned, the algorithm uses more communications per computation in order to keep the ratio at the optimal operating point.

We now present our main result which states that the iterates of each agent converge to the fixed-point linearly with rate equal to the contraction factor of the fixed-point operators. We prove the result in Appendix A.

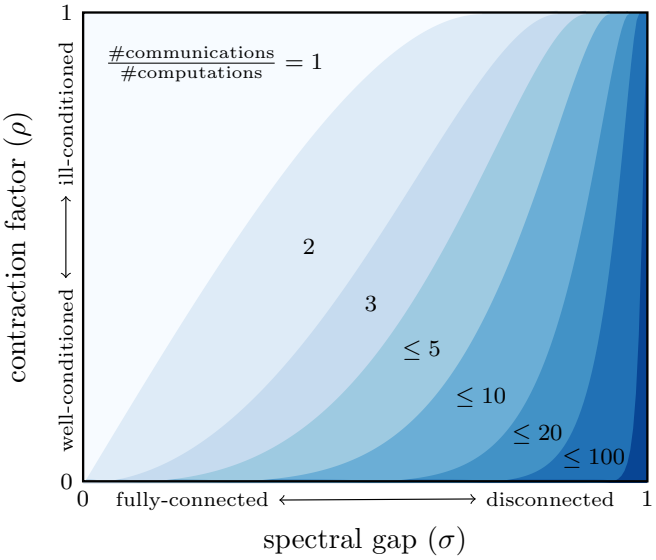

Fig. 1. Ratio between the number of communications and computations as a function of the spectral gap σ and the contraction factor ρ. The color indicates the ratio from light (small ratio) to dark (large ratio).

Theorem 1. (Main result). Suppose Assumptions 1 and 2 hold for some $x^\star \in \mathbb{R}^d$ and $\rho, \sigma \in (0,1)$. Then the iterate sequence $\{x_i^k\}_{k\geq 0}$ of each agent $i \in \{1,\dots,n\}$ in our algorithm converges to the fixed-point $x^\star$ linearly with rate ρ. In other words,

$$\|x_i^k - x^\star\| = \mathcal{O}(\rho^k) \quad \text{for all } i \in \{1,\dots,n\}. \tag{8}$$

Theorem 1 shows that the algorithm computes the fixed-point in a decentralized manner at the *same* rate as the centralized fixed-point iterations in (3) in terms of the number of computations. In other words, the algorithm converges just as fast (in the worst case) as if each agent had access to the information of all other agents at every iteration. Instead of communicating all this information, however, it is sufficient to only perform m rounds of communication where m is defined in (7). Note that this is the minimum number of communication rounds required so that the spectral gap of the m-step gossip matrix $\prod_{\ell=1}^m W^{k,\ell}$ at iteration k is no greater than

$$\tfrac{\sqrt{1+\rho}-\sqrt{1-\rho}}{2}. \tag{9}$$

The convergence rate in Theorem 1 may be misleading since it describes the convergence per iteration, but m rounds of communication are performed per iteration. However, we now show that the convergence rate is fast even when we account for this extra communication. To do so, we define a *cycle* as the amount of time it takes to communicate with local neighbors and/or compute the local operators $\{T_i\}$. In particular, our algorithm performs m cycles per iteration where the first $m-1$ cycles consist only of communication and the last cycle uses both communication and computation. Then the error decreases by a factor of $\gamma = \rho^{1/m}$ per cycle which is plotted in Figure 2 as a function of the contraction factor ρ and spectral gap σ.

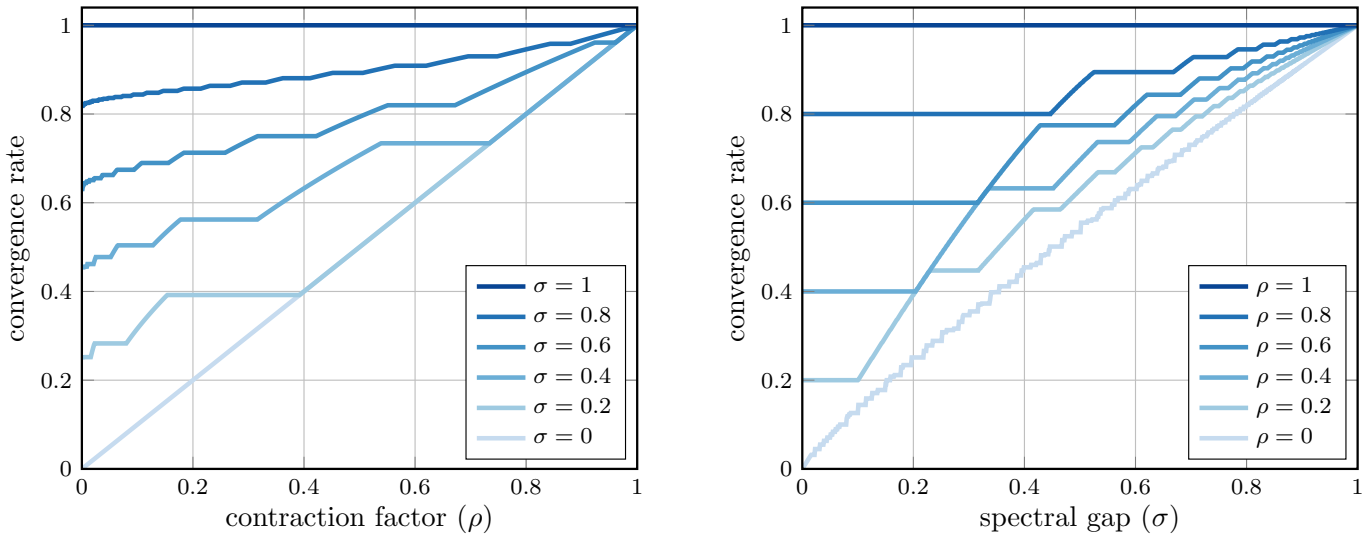

Fig. 2. Convergence rate as a function of the contraction factor ρ and spectral gap σ. Here, we define a *cycle* as the amount of time it takes to communicate with local neighbors and/or compute the local operators $\{T_i\}$. We then define the convergence rate as the scalar $\gamma \in (0,1)$ such that the norm of the iterates from the fixed-point of each agent decreases by a factor of γ at each cycle. Our algorithm performs m cycles per iteration, so $\gamma = \rho^{1/m}$.

4. APPLICATION: TARGET LOCALIZATION

To illustrate our results, we use our algorithm to have a group of agents solve a target localization problem as illustrated in Figure 3. We assume each agent can measure its distance (but not angle) to the target and can communicate with local neighbors.

Suppose agents are located in a two-dimensional plane where the location of agent $i \in \{1,\dots,n\}$ is given by $(p_i, q_i) \in \mathbb{R}^2$. Each agent knows its own position but *not* the location of the target, denoted by $x^\star = (p^\star, q^\star) \in \mathbb{R}^2$. Agent i is capable of measuring its distance to the target,

$$r_i = \sqrt{(p_i - p^\star)^2 + (q_i - q^\star)^2}.$$

The objective function $f_i : \mathbb{R}^2 \to \mathbb{R}$ associated to agent i is

$$f_i(p,q) = \tfrac{1}{2}\left(\sqrt{(p_i - p)^2 + (q_i - q)^2} - r_i\right)^2.$$

Then in order to locate the target, the agents cooperate to solve the distributed nonlinear least-squares problem

$$\underset{p,q\in\mathbb{R}}{\text{minimize}} \ \frac{1}{n}\sum_{i=1}^{n} f_i(p,q). \tag{10}$$

This is equivalent to the distributed fixed-point problem where $T_i = I - \alpha\,\nabla f_i$ is the fixed-point operator associated with agent i and $\alpha > 0$ is the stepsize.

Agents can communicate with local neighbors as shown in Figure 3. To simulate randomly dropped packets from agent 4 to agent 1, the gossip matrix at each iteration is randomly chosen from the set

$$W \in \left\{ \begin{bmatrix} 0 & \frac{3}{8} & \frac{1}{4} & 0 & \frac{3}{8} \\ \frac{1}{8} & 0 & \frac{3}{4} & \frac{1}{8} & 0 \\ 0 & \frac{5}{8} & 0 & \frac{3}{8} & 0 \\ \frac{3}{8} & 0 & 0 & 0 & \frac{5}{8} \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 \end{bmatrix}, \begin{bmatrix} 0 & \frac{1}{2} & \frac{1}{4} & 0 & \frac{1}{4} \\ \frac{1}{4} & 0 & \frac{3}{4} & 0 & 0 \\ 0 & \frac{1}{2} & 0 & \frac{1}{2} & 0 \\ \frac{1}{4} & 0 & 0 & 0 & \frac{3}{4} \\ \frac{1}{2} & 0 & 0 & \frac{1}{2} & 0 \end{bmatrix} \right\}.$$

Both gossip matrices satisfy Assumption 2 with maximum spectral gap $\sigma \approx 0.7853$.

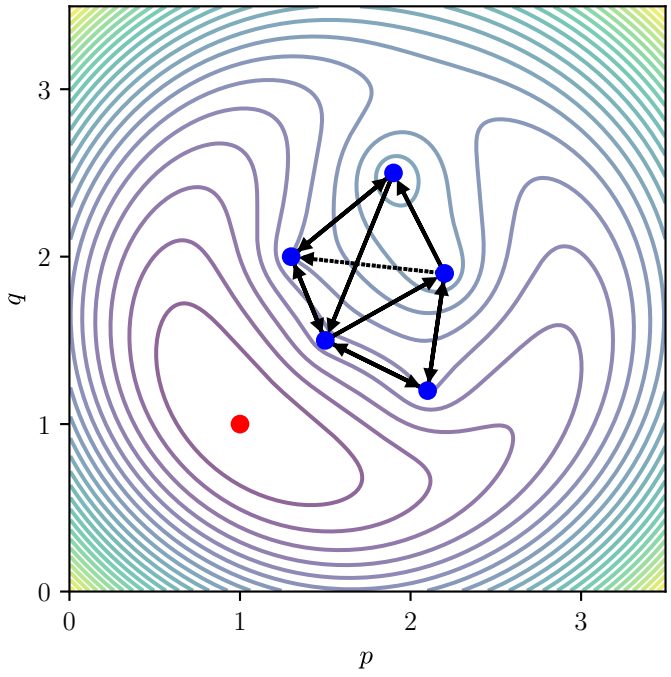

Fig. 3. Setup of the target localization problem. The position $(p_i, q_i) \in \mathbb{R}^2$ of agent $i \in \{1,\dots,5\}$ is denoted by a blue circle with the position of the target in red at $(p^\star, q^\star) = (1,1)$. The black arrows indicate the flow of information, with an arrow from agent i to j if agent j receives information from agent i. The dashed arrow indicates the link that varies in time. The smooth curves are the contour lines of the objective function for the distributed nonlinear least-squares problem in (10). Note that the problem is nonconvex since the level sets are nonconvex.

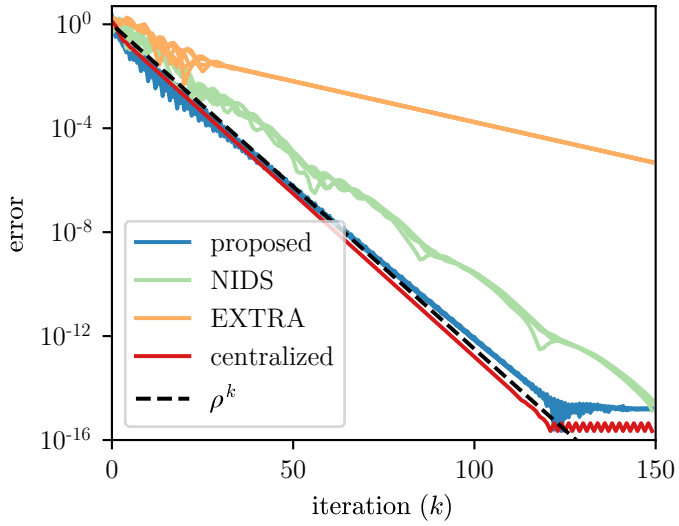

Fig. 4. Plot of the error as a function of the iteration for the target localization problem. The blue lines indicate the error $\|x_i^k - x^\star\|$ for each of the five agents computed using our proposed distributed algorithm while the red line indicates the error using the centralized fixed-point iteration (3). Note that our algorithm performs one computation and $m = 6$ communications per iteration.

We choose the stepsize to optimize the asymptotic rate of convergence. In particular, the estimate of each agent becomes arbitrarily close to the target as $k \to \infty$, so the optimal stepsize is $\alpha = \frac{2}{\lambda_1+\lambda_2}$ where λ_1 and λ_2 are smallest and largest eigenvalues of the Hessian matrix evaluated at the target, in other words, $\nabla^2 f(p^\star, q^\star)$. Since the objective function is two-dimensional, the sum of its smallest and largest eigenvalues is equal to its trace, so

$$\lambda_1 + \lambda_2 = \operatorname{trace}(\nabla^2 f) = \frac{1}{n}\sum_{i=1}^{n} \operatorname{trace}(\nabla^2 f_i)$$

where the trace of the Hessian of the objective function on agent i is

$$\operatorname{trace}(\nabla^2 f_i) = 2 - \frac{r_i}{\sqrt{(p_i-p)^2+(q_i-q)^2}}.$$

At the target, the trace is equal to one which gives an optimal stepsize of $\alpha = 2$. Since NIDS and EXTRA are unstable with this stepsize, we instead use $\alpha = 1$ and $\alpha = 0.5$, respectively, for these algorithms in the simulation.

We choose the contraction factor as the convergence rate of the centralized fixed-point iterations which is $\rho \approx 0.75$. Then our algorithm performs $m = 6$ communication rounds per iteration. We have each agent initialize its states with its position $x_i^0 = (p_i, q_i) \in \mathbb{R}^2$ and $y_i^0 = (0, 0) \in \mathbb{R}^2$.

In Figure 4, we plot the error of each agent as a function of the iteration. The error converges to zero at the same rate as the centralized fixed-point iterations (3) as expected from Theorem 1. Note that the nonconvexity of the objective function affects the initial transient behavior, but once the iterates are close to the optimizer, the objective function is approximately quadratic. Also, our algorithm uses $m = 6$ communications per iteration while NIDS and EXTRA use only one; while our algorithm is more computationally efficient, it also uses more communications than NIDS to obtain a solution with a given precision.

5. CONCLUSION

We developed a new algorithm for distributed fixed-point computation, or equivalently, distributed optimization. Our algorithm converges with the same rate as centralized fixed-point iterations (or equivalently, gradient descent) in terms of the number of computations. Furthermore, it uses the minimum number of communications necessary to do so. Such an algorithm is particularly useful when computations are expensive relative to the cost of communication.

REFERENCES

Bertsekas, D.P. (1983). Distributed asynchronous computation of fixed points. *Mathematical Programming*, 27(1), 107–120.

Jakovetić, D. (2019). A unification and generalization of exact distributed first-order methods. *IEEE Trans. Signal and Information Processing over Networks*, 5(1), 31–46.

Li, Z., Shi, W., and Yan, M. (2017). A decentralized proximal-gradient method with network independent step-sizes and separated convergence rates. *arXiv:1704.07807*.

Nedić, A., Olshevsky, A., and Shi, W. (2017). Achieving geometric convergence for distributed optimization over time-varying graphs. *SIAM Journal on Optimization*, 27(4), 2597–2633.

Nedić, A. and Ozdaglar, A. (2009). Distributed subgradient methods for multi-agent optimization. *IEEE Transactions on Automatic Control*, 54(1), 48–61.

Qu, G. and Li, N. (2018). Harnessing smoothness to accelerate distributed optimization. *IEEE Transactions on Control of Network Systems*, 5(3), 1245–1260.

Scaman, K., Bach, F., Bubeck, S., Lee, Y.T., and Massoulié, L. (2017). Optimal algorithms for smooth and strongly convex distributed optimization in networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 3027–3036.

Shi, W., Ling, Q., Wu, G., and Yin, W. (2015). EXTRA: An exact first-order algorithm for decentralized consensus optimization. *SIAM Journal on Optimization*, 25(2), 944–966.

Sundararajan, A., Hu, B., and Lessard, L. (2017). Robust convergence analysis of distributed optimization algorithms. In *Allerton Conference on Communication, Control, and Computing*, 1206–1212.

Sundararajan, A., Van Scoy, B., and Lessard, L. (2018). A canonical form for first-order distributed optimization algorithms. In *arXiv:1809.08709*.

Xiao, L., Boyd, S., and Kim, S.J. (2007). Distributed average consensus with least-mean-square deviation. *Journal of parallel and distributed computing*, 67(1), 33–46.

Xu, J., Zhu, S., Soh, Y.C., and Xie, L. (2015). Augmented distributed gradient methods for multi-agent optimization under uncoordinated constant stepsizes. In *IEEE Conference on Decision and Control*, 2055–2060.

Yuan, K., Ying, B., Zhao, X., and Sayed, A.H. (2019). Exact diffusion for distributed optimization and learning—Part I: Algorithm development. *IEEE Transactions on Signal Processing*, 67(3), 708–723.

Appendix A. PROOF OF THEOREM 1

We now prove linear convergence of the iterates of our algorithm to the fixed-point of the average operator.

Average and disagreement operators. To simplify the notation, we define the *average operator* $\mathsf{avg} : \mathbb{R}^{nd} \to \mathbb{R}^{nd}$ with

$$\mathsf{avg}(\mathbf{x}) := (\tfrac{1}{n}\mathbb{1}\mathbb{1}^\mathsf{T} \otimes I_d)\,\mathbf{x}$$

along with the *disagreement operator* $\mathsf{dis} : \mathbb{R}^{nd} \to \mathbb{R}^{nd}$ with

$$\mathsf{dis}(\mathbf{x}) := \big((I_n - \tfrac{1}{n}\mathbb{1}\mathbb{1}^\mathsf{T}) \otimes I_d\big)\,\mathbf{x}$$

where $\otimes$ denotes the Kronecker product. Note that any point can be decomposed into its average and disagreement components since $\mathsf{avg} + \mathsf{dis} = I$. Also, the operators are orthogonal in that $\mathsf{avg}(\mathbf{x})^\mathsf{T}\mathsf{dis}(\mathbf{y}) = 0$ for all $\mathbf{x}, \mathbf{y} \in \mathbb{R}^{nd}$.

Vectorized form. We can then write our algorithm in vectorized form as

$$\mathbf{v}^k = \mathcal{W}^k(\mathbf{x}^k) \tag{A.1a}$$
$$\mathbf{u}^k = \mathcal{T}(\mathbf{v}^k) \tag{A.1b}$$
$$\mathbf{y}^{k+1} = \mathbf{y}^k + \mathbf{x}^k - \mathbf{v}^k \tag{A.1c}$$
$$\mathbf{x}^{k+1} = \mathbf{u}^k - \lambda\,\mathbf{y}^{k+1} \tag{A.1d}$$

with $\mathsf{avg}(\mathbf{y}^0) = 0$ where the concatenated vectors are

$$\mathbf{u}^k := \begin{bmatrix} u_1^k \\ \vdots \\ u_n^k \end{bmatrix},\ \mathbf{v}^k := \begin{bmatrix} v_1^k \\ \vdots \\ v_n^k \end{bmatrix},\ \mathbf{x}^k := \begin{bmatrix} x_1^k \\ \vdots \\ x_n^k \end{bmatrix},\ \mathbf{y}^k := \begin{bmatrix} y_1^k \\ \vdots \\ y_n^k \end{bmatrix},$$

and the m-step consensus operator $\mathcal{W}^k : \mathbb{R}^{nd} \to \mathbb{R}^{nd}$ and global fixed-point operator $\mathcal{T} : \mathbb{R}^{nd} \to \mathbb{R}^{nd}$ are defined as

$$\mathcal{W}^k := \prod_{\ell=1}^{m} \big(W^{k,\ell} \otimes I_d\big) \quad \text{and} \quad \mathcal{T} := \begin{bmatrix} T_1 & & 0 \\ & \ddots & \\ 0 & & T_n \end{bmatrix}.$$

Fixed-point. Define the points $\mathbf{u}^\star, \mathbf{v}^\star, \mathbf{x}^\star, \mathbf{y}^\star \in \mathbb{R}^{nd}$ as

$$\mathbf{v}^\star = \mathbf{x}^\star = \mathbb{1} \otimes x^\star, \quad \mathbf{u}^\star = \mathcal{T}(\mathbf{v}^\star), \quad \mathbf{y}^\star = \tfrac{1}{\lambda}(\mathbf{u}^\star - \mathbf{x}^\star).$$

Then $(\mathbf{u}^\star, \mathbf{v}^\star, \mathbf{x}^\star, \mathbf{y}^\star)$ is a fixed-point of the concatenated system (A.1) since the gossip matrix is row-stochastic at each iteration. Also, $\mathsf{avg}(\mathbf{y}^\star) = 0$ since $x^\star$ satisfies (4).

Error system. To analyze the algorithm, we use a change of variables to put it in error coordinates. The error vectors

$$\bar{\mathbf{u}}^k := \mathbf{u}^k - \mathbf{u}^\star \qquad \bar{\mathbf{x}}^k := \mathbf{x}^k - \mathbf{x}^\star$$
$$\bar{\mathbf{v}}^k := \mathbf{v}^k - \mathbf{v}^\star \qquad \bar{\mathbf{y}}^k := \mathbf{y}^k - \mathbf{y}^\star$$

satisfy the iterations

$$\bar{\mathbf{y}}^{k+1} = \bar{\mathbf{y}}^k + \bar{\mathbf{x}}^k - \bar{\mathbf{v}}^k \tag{A.2a}$$
$$\bar{\mathbf{x}}^{k+1} = \bar{\mathbf{u}}^k - \lambda\,\bar{\mathbf{y}}^{k+1} \tag{A.2b}$$

for $k \geq 0$.

Fixed-point operator. From Assumption 1, the global fixed-point operator $\mathcal{T}$ satisfies

$$\mathbf{x}^\star = \mathsf{avg}\big(\mathcal{T}(\mathbf{x}^\star)\big) \tag{A.3}$$

and

$$\|\mathcal{T}(\mathbf{x}) - \mathcal{T}(\mathbf{x}^\star)\| \leq \rho\,\|\mathbf{x} - \mathbf{x}^\star\| \tag{A.4}$$

for all $\mathbf{x} \in \mathbb{R}^{nd}$. In other words, $\mathcal{T}$ is a contraction with respect to the point $\mathbf{x}^\star$ with contraction factor ρ.

Consensus operator. From Assumption 2 along with the definition of m, the consensus operator $\mathcal{W}^k$ satisfies

$$\|\mathsf{dis}\big(\mathcal{W}^k(\mathbf{x})\big)\| \leq \sigma^m\,\|\mathsf{dis}(\mathbf{x})\| \leq \sigma_0\,\|\mathsf{dis}(\mathbf{x})\| \tag{A.5}$$

for all $\mathbf{x} \in \mathbb{R}^{nd}$ and all $k \geq 0$ where

$$\sigma_0 := \tfrac{\sqrt{1+\rho}-\sqrt{1-\rho}}{2}. \tag{A.6}$$

Consensus direction. We now derive some properties of the average error vectors. Using the assumption that the gossip matrix is column-stochastic, we have

$$\mathsf{avg}(\bar{\mathbf{x}}^k) = \mathsf{avg}(\bar{\mathbf{v}}^k) \quad \text{for all } k \geq 0. \tag{A.7}$$

The iterates are initialized such that $\mathsf{avg}(\bar{\mathbf{y}}^0) = 0$ (recall that $\mathsf{avg}(\mathbf{y}^\star) = 0$). Taking the average of (A.1c), we have that the average is preserved. In other words, we have that $\mathsf{avg}(\bar{\mathbf{y}}^{k+1}) = \mathsf{avg}(\bar{\mathbf{y}}^k)$ for all $k \geq 0$. Then by induction,

$$\mathsf{avg}(\bar{\mathbf{y}}^k) = 0 \quad \text{for all } k \geq 0. \tag{A.8}$$

Lyapunov function. To prove convergence, we will show that the function $V : \mathbb{R}^{nd} \times \mathbb{R}^{nd} \to \mathbb{R}$ defined by

$$V(\bar{\mathbf{x}}, \bar{\mathbf{y}}) := \|\mathsf{avg}(\bar{\mathbf{x}})\|^2 + \begin{bmatrix} \mathsf{dis}(\bar{\mathbf{x}}) \\ \mathsf{dis}(\bar{\mathbf{y}}) \end{bmatrix}^\mathsf{T} \left(\begin{bmatrix} 1 & \lambda \\ \lambda & \lambda \end{bmatrix} \otimes I_{nd} \right) \begin{bmatrix} \mathsf{dis}(\bar{\mathbf{x}}) \\ \mathsf{dis}(\bar{\mathbf{y}}) \end{bmatrix} \tag{A.9}$$

is a Lyapunov function for the algorithm, that is, it is both positive definite and decreasing along system trajectories. Note that $\lambda \in (0, 1)$ since $\rho \in (0, 1)$, so the matrix in (A.9) is positive definite. Then V is also positive definite, meaning that $V(\bar{\mathbf{x}}, \bar{\mathbf{y}}) \geq 0$ for all $\bar{\mathbf{x}}$ and $\bar{\mathbf{y}}$, and $V(\bar{\mathbf{x}}, \bar{\mathbf{y}}) = 0$ if and only if $\bar{\mathbf{x}} = 0$ and $\mathsf{dis}(\bar{\mathbf{y}}) = 0$ (recall that $\mathsf{avg}(\bar{\mathbf{y}}^k) = 0$). Next, we show that the Lyapunov function decreases by a factor of at least ρ^2 at each iteration. Define the weighted difference in the Lyapunov function between iterations as

$$\Delta V^k := V(\bar{\mathbf{x}}^{k+1}, \bar{\mathbf{y}}^{k+1}) - \rho^2\,V(\bar{\mathbf{x}}^k, \bar{\mathbf{y}}^k).$$

Subsituting the expressions for the iterates in (A.2) and using the properties of the average iterates in (A.7) and (A.8), we have

$$\begin{aligned} \Delta V^k = &- \big(\rho^2\,\|\bar{\mathbf{v}}^k\|^2 - \|\bar{\mathbf{u}}^k\|^2\big) \\ &- 2\rho^2\,\big(\sigma_0^2\,\|\mathsf{dis}(\bar{\mathbf{x}}^k)\|^2 - \|\mathsf{dis}(\bar{\mathbf{v}}^k)\|^2\big) \\ &- 2\sigma_0^2\,\big\|\mathsf{dis}\big(\bar{\mathbf{v}}^k + \lambda\,(\bar{\mathbf{x}}^k + \bar{\mathbf{y}}^k)\big)\big\|^2. \end{aligned}$$

The first term is nonpositive since $\mathcal{T}$ satisfies (A.4), the second since $\mathcal{W}^k$ satisfies (A.5), and the third since it is a squared norm. Therefore, $\Delta V^k \leq 0$ for all $k \geq 0$. Applying this inequality at each iteration and summing, we obtain the bound

$$V(\bar{\mathbf{x}}^k, \bar{\mathbf{y}}^k) \leq \rho^{2k}\,V(\bar{\mathbf{x}}^0, \bar{\mathbf{y}}^0) \quad \text{for all } k \geq 0.$$

Bound. Finally, we use the Lyapunov function to show that $\|x_i^k - x^\star\|$ converges to zero linearly with rate ρ for each agent $i \in \{1, \ldots, n\}$. The norm is upper-bounded by

$$\|x_i^k - x^\star\|^2 \leq \mathsf{cond}\left(\begin{bmatrix} 1 & \lambda \\ \lambda & \lambda \end{bmatrix}\right) V(\bar{\mathbf{x}}^k, \bar{\mathbf{y}}^k) \leq c^2\,\rho^{2k}$$

where the nonnegative constant $c \in \mathbb{R}$ is defined as

$$c := \sqrt{\mathsf{cond}\left(\begin{bmatrix} 1 & \lambda \\ \lambda & \lambda \end{bmatrix}\right) V(\bar{\mathbf{x}}^0, \bar{\mathbf{y}}^0)}$$

and $\mathsf{cond}(\cdot)$ denotes the condition number. Taking the square root, we obtain the bound

$$\|x_i^k - x^\star\| \leq c\,\rho^k$$

for each agent $i \in \{1, \ldots, n\}$ and iteration $k \geq 0$. □