A Distributed Optimization Algorithm over Time-Varying Graphs with Efficient Gradient Evaluations ⋆

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 propose an algorithm for distributed optimization over time-varying communication networks. Our algorithm uses an optimized ratio between the number of rounds of communication and gradient evaluations to achieve fast convergence. The iterates converge to the global optimizer at the same rate as centralized gradient descent when measured in terms of the number of gradient evaluations while using the minimum number of communications to do so. Furthermore, the iterates converge at a near-optimal rate when measured in terms of the number of communication rounds. We compare our algorithm with several other known algorithms on a distributed target localization problem.

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 all the agents to calculate the global optimizer using only local communications and gradient evaluations.

Many algorithms have been proposed recently 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), Exact Diffusion by Yuan et al. (2019), and SVL by Sundararajan et al. (2019) among others. In each algorithm, agents do the following at each step:

- communicate state variables with local neighbors,
- evaluate the local gradient ∇f_i, and
- update local state variables.

Each algorithm alternates between these three steps and therefore uses the same number of communications and local gradient evaluations. In this paper, however, we allow this ratio to depend on the properties of the objective function and the communication network. To characterize the convergence properties of our algorithm, we use the following notions of time.

- We define a *step* as one round of communication and at most one gradient evaluation.
- We define an *iteration* as m rounds of communication and one gradient evaluation.

In other words, an iteration consists of m steps where each step is at least as simple as that of the algorithms previously mentioned. We assume that local state updates have neglible cost and can therefore be performed any number of times per step or iteration.

For example, consider an algorithm that updates as follows:

iteration	1			2			3		
step	1	2	3	4	5	6	7	8	9
communication	✓	✓	✓	✓	✓	✓	✓	✓	✓
gradient evaluation			✓			✓			✓

This algorithm performs three rounds of communication per gradient evaluation, so $m = 3$.

Main contributions. In this work, we propose a novel decentralized algorithm for solving (1). Instead of using the same number of communication rounds as gradient evaluations, our algorithm sets the ratio between these using global problem parameters. We show the following:

(1) The iterates of our algorithm converge to the optimizer with the same rate as centralized gradient descent in terms of number of iterations. Furthermore, our algorithm uses the minimum number m of communications per gradient evaluation to achieve this.

(2) The iterates of our algorithm converge to the optimizer with a near-optimal rate in terms of number of steps, despite not evaluating the gradient at each step.

A decentralized algorithm can trivially obtain the same rate as centralized gradient descent if we use an infinite number of communication rounds per iteration (i.e., $m \to \infty$) since then every agent can compute an exact average at each iteration (and therefore can evaluate the global gradient). We show, however, that our algorithm achieves the same rate with a *finite* number of communication rounds per iteration, and we characterize precisely how many communication rounds are required.

⋆ This material is based upon work supported by the National Science Foundation under Grant No. 1656951 and 1750162.

To prove convergence of our algorithm, we make the following assumptions.

- The local objective functions satisfy a contraction property that is weaker than assuming smoothness and strong convexity.
- The communication network may be time-varying and either directed or undirected as long as it is sufficiently connected and the associated weight matrix is doubly stochastic at each step.

Perhaps the algorithm most similar to ours is the multi-step dual accelerated (MSDA) algorithm by Scaman et al. (2017). This algorithm also adjusts the ratio between the number of communication rounds and gradient evaluations to achieve fast convergence. The MSDA algorithm is provably optimal in terms of both the number of communications and gradient evaluations when the objective function is smooth and strongly convex and the communication network is fixed. Compared to our algorithm, the MSDA algorithm achieves an accelerated rate of convergence by making stronger assumptions on both the objective function and the communication network while we prove a non-accelerated rate using weaker assumptions.

The remainder of the paper is organized as follows. We first set up 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 $\mathbb{1} \in \mathbb{R}^n$ and the identity matrix by $I_n \in \mathbb{R}^{n\times n}$. We use $\|\cdot\|$ to denote the 2-norm of a vector as well as the induced 2-norm of a matrix.

2. PROBLEM SETUP

We now discuss the assumptions on the objective function and the communication network that we make in order to solve the distributed optimization problem (1).

2.1 Objective function

Assumption 1. The distributed optimization problem (1) has an optimizer $x^\star \in \mathbb{R}^d$. Furthermore, there exists a *stepsize* $\alpha > 0$ and *contraction factor* $\rho \in (0,1)$ such that

$$\|x - x^\star - \alpha\left(\nabla f_i(x) - \nabla f_i(x^\star)\right)\| \le \rho\,\|x - x^\star\| \qquad (2)$$

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

Each $\nabla f_i(x^\star)$ is in general nonzero, although we have

$$\sum_{i=1}^n \nabla f_i(x^\star) = 0. \qquad (3)$$

Assumption 1 also implies that

$$\|x - x^\star - \alpha\,\nabla f(x)\| \le \rho\,\|x - x^\star\| \quad \text{for all } x \in \mathbb{R}^d,$$

so the global objective function satisfies the same property as the local functions. Assumption 1 holds if the local functions satisfy a one-point smooth and strong convexity property as described in the following proposition.

Proposition 1. Let $0 < \mu \le L$, and suppose each local function f_i is one-point μ-smooth and L-strongly convex with respect to the global optimizer, in other words,

$$\mu\|x - x^\star\|^2 \le \left(\nabla f_i(x) - \nabla f_i(x^\star)\right)^{\mathsf{T}}(x - x^\star) \le L\|x - x^\star\|^2$$

for all $x \in \mathbb{R}^d$ and all $i \in \{1,\dots,n\}$. Then (2) holds with stepsize $\alpha = \frac{2}{L+\mu}$ and contraction factor $\rho = \frac{L-\mu}{L+\mu}$.

Since the optimizer $x^\star$ is not known a priori, it may be difficult to verify (2). In this case, a useful sufficient condition is that each f_i is μ-smooth and L-strongly convex, meaning that

$$\mu\|x - y\|^2 \le \left(\nabla f_i(x) - \nabla f_i(y)\right)^{\mathsf{T}}(x - y) \le L\|x - y\|^2$$

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

2.2 Communication network

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

Definition 2. (Gossip matrix). We say that the 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

$$\sigma := \|W - \tfrac{1}{n}\mathbb{1}\mathbb{1}^{\mathsf{T}}\|. \qquad (4)$$

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

The spectral gap characterizes the connectivity of the communication network. In particular, a small spectral gap corresponds to a well-connected network and vice versa. One way to obtain a gossip matrix is to set $W = I - \mathcal{L}$ where $\mathcal{L}$ is the (possibly weighted) graph Laplacian. We make the following assumption about 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 step of the algorithm.

This assumption allows the communication network to be both directed and time-varying. At each step, 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, and see Nedić and Olshevsky (2015) for distributed optimization over non-doubly-stochastic networks using the push-sum protocol.

2.3 Centralized gradient descent

The (centralized) gradient descent iterations are given by

$$x^{k+1} = x^k - \alpha\,\nabla f(x^k) \qquad (5)$$

where $\alpha > 0$ is the stepsize. Under Assumption 1, this method converges to the optimizer linearly with rate ρ. In other words, $\|x^k - x^\star\| = \mathcal{O}(\rho^k)$. While this method could be approximated in a decentralized manner using a large number of steps per iteration (so that every agent can compute the average gradient at each iteration), we show that our algorithm achieves the same convergence rate using the minimal number m of necessary rounds of communication per gradient evaluation.

3. MAIN RESULTS

To solve the distributed optimization problem, we now introduce our algorithm, which depends on the stepsize α, contraction factor ρ, and spectral gap σ.

Algorithm

```
Parameters: stepsize $\alpha > 0$, contraction factor $\rho \in (0,1)$, and spectral gap $\sigma \in (0,1)$.
Inputs: local functions $f_i : \mathbb{R}^d \to \mathbb{R}$ on agent $i \in \{1,\dots,n\}$, gossip matrices $\{w_{ij}^{k\ell}\}$ at iteration $k$ and communication round $\ell$.
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
      $m := \underset{r\geq\rho,\, s\geq\sigma}{\text{minimize}} \left\lceil \log_s\left(\frac{\sqrt{1+r}-\sqrt{1-r}}{2}\right)\right\rceil.$

for iteration $k = 0,1,2,\dots$ do
  for agent $i \in \{1,\dots,n\}$ do
    $v_{i,0}^k = x_i^k$
    for step $\ell = 1,\dots,m$ do
      $v_{i,\ell}^k = \sum_{j=1}^n w_{ij}^{k\ell}\, v_{j,\ell-1}^k$          (local communication)
    end for
    $u_i^k = v_{i,m}^k - \alpha\nabla f_i(v_{i,m}^k)$          (local gradient evaluation)
    $y_i^{k+1} = y_i^k + x_i^k - v_{i,m}^k$          (local state update)
    $x_i^{k+1} = u_i^k - \sqrt{1-\rho^2}\, y_i^{k+1}$          (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 gradient ∇f_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 optimizer $x^\star$ of the global objective function f. Note that agents are required to know the global parameters ρ and σ so that they can calculate the number of communication rounds m.

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

$$m := \underset{r\geq\rho,\, s\geq\sigma}{\text{minimize}} \left\lceil \log_s\left(\tfrac{\sqrt{1+r}-\sqrt{1-r}}{2}\right)\right\rceil. \tag{6}$$

This is the minimum integer number of communication rounds 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 $\frac{\sqrt{1+\rho}-\sqrt{1-\rho}}{2}$. Since only one gradient evaluation is performed per iteration, this adjusts the ratio between the number of communications and gradient evaluations as shown in Figure 1. In particular, the algorithm uses a single communication per gradient evaluation when the network is sufficiently connected (σ small) and the objective function is ill-conditioned (ρ large). As the network becomes more disconnected and/or the objective function becomes more well-conditioned, the algorithm uses more communications per gradient evaluation in order to keep the ratio at the optimal operating point.

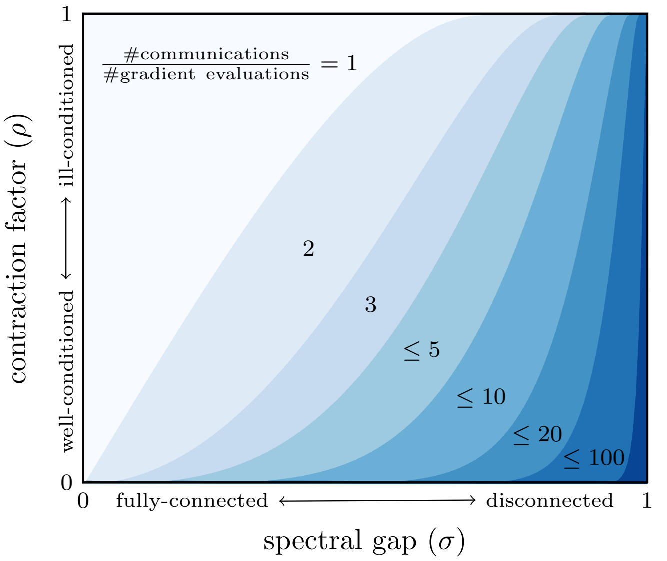

Fig. 1. Ratio between the number of communications and gradient evaluations as a function of the spectral gap σ and the contraction factor ρ. The color indicates the ratio from light (small ratio) to dark (large ratio).

We now present our main result, which states that the iterates of each agent converge to the global optimizer linearly with a rate equal to the contraction factor ρ. We prove the result in Appendix A.

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

$$\|x_i^k - x^\star\| = \mathcal{O}(\rho^k) \quad \text{for all } i \in \{1,\dots,n\}. \tag{7}$$

Theorem 1 states that the iterates of our algorithm converge to the optimal solution of (1) in a decentralized manner at the *same* rate as centralized gradient descent (5) in terms of the number of iterations. 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 (6).

The convergence rate in Theorem 1 is in terms of the number of iterations. To compare the performance of our algorithm in terms of the number of steps, we plot the convergence rate per step in Figure 2. For comparison, we also plot the rate of the algorithm SVL by Sundararajan et al. (2019). This algorithm is designed to optimize the convergence rate per step and requires agents to compute their local gradient at each step of the algorithm. In contrast, our algorithm is slightly slower than the optimal algorithm but uses far fewer computations since local gradients are only evaluated once every m steps.

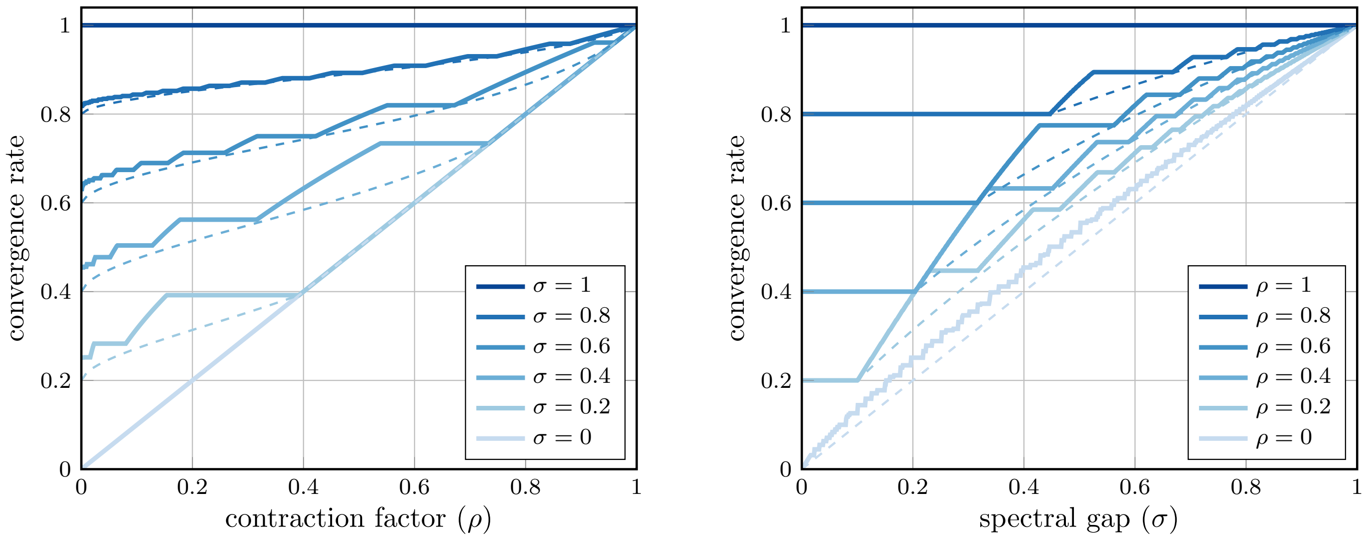

Fig. 2. Convergence rate in terms of the number of steps as a function of the contraction factor ρ and spectral gap σ. Solid lines indicate our algorithm while dashed lines indicate the optimal algorithm SVL by Sundararajan et al. (2019). Since our algorithm converges with rate ρ with respect to the number of iterations and performs m steps per iteration, the convergence rate with respect to the number of steps is $\rho^{1/m}$ where m is defined in (6).

4. APPLICATION: TARGET LOCALIZATION

To illustrate our results, we use our algorithm to solve the distributed target localization problem illustrated in Figure 3, which is inspired by the example in Section 18.3 of the book by Boyd and Vandenberghe (2018). We assume each agent (blue dot) can measure its distance (but not angle) to the target (red dot) and can communicate with local neighbors.

Suppose agents are located in a two-dimensional plane where the location of agent $i \in \{1, \ldots, n\}$ 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

$$\operatorname*{minimize}_{p,q \in \mathbb{R}} \; \frac{1}{n} \sum_{i=1}^{n} f_i(p, q). \tag{8}$$

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 $\sigma \approx 0.7853$.

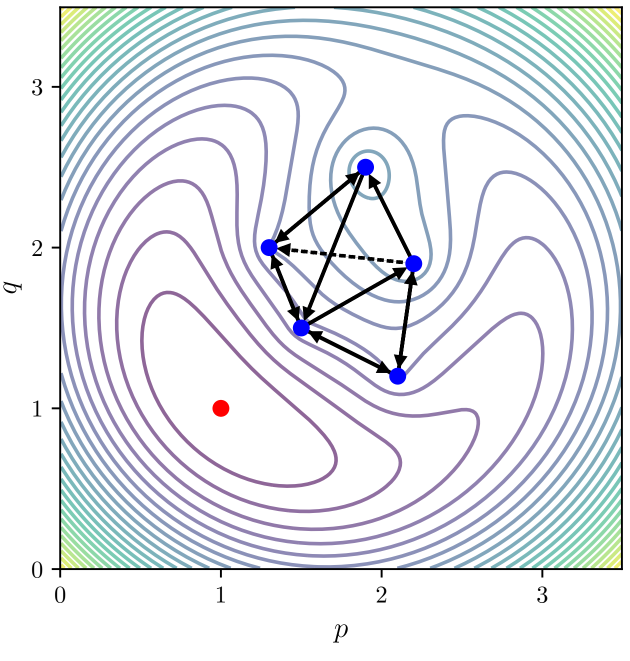

Fig. 3. Setup of the target localization problem. The position $(p_i, q_i) \in \mathbb{R}^2$ of agent $i \in \{1, \ldots, 5\}$ is denoted by a blue circle with the position of the target in red at $(p^\star, q^\star) = (1, 1)$. The black arrows indicate the flow of information with an arrow from agent 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 (8). Note that the problem is nonconvex since the level sets are nonconvex.

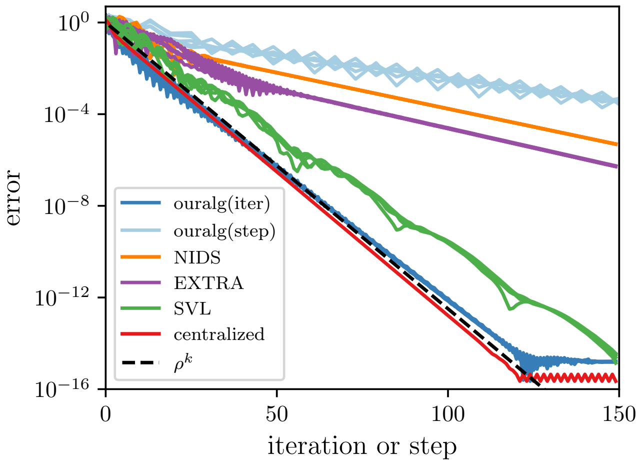

Fig. 4. Plot of the error for the target localization problem. The blue lines indicate the error $\|x_i^k - x^\star\|$ for each of the five agents computed using our proposed decentralized algorithm as a function of either the iteration (dark blue) or step (light blue); iterations and steps are equivalent for each of the other algorithms. The red line indicates the error using centralized gradient descent (5). Our algorithm performs one gradient evaluation 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 the 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

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

where the trace of the local Hessian is

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

The trace is equal to one at the target, so the optimal stepsize is $\alpha = 2$. Since NIDS and EXTRA are unstable with this stepsize, we instead use $\alpha = 1$ and $\alpha = 0.5$, respectively, for these algorithms. The parameters of SVL are completely determined by σ and $\kappa = \frac{1+\rho}{1-\rho}$.

We choose the contraction factor as the convergence rate of centralized gradient descent 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 either the iteration or step. As expected from Theorem 1, the error of our algorithm converges to zero at the same rate as centralized gradient descent (5) in terms of iterations. Our algorithm uses $m = 6$ communications per iteration while NIDS, EXTRA, and SVL use only one; while our algorithm is more efficient in terms of gradient evaluations, it also uses more communications than the other algorithms to obtain a solution with a given precision.

5. CONCLUSION

We developed an algorithm for distributed optimization that uses the minimal amount of communication necessary such that the iterates converge to the optimizer at the same rate as centralized gradient descent. Furthermore, the convergence rate of our algorithm is near-optimal (in the worst-case) in terms of the number of communication rounds even though the gradient is not computed at each step. Such an algorithm is particularly useful when gradient evaluations are expensive relative to the cost of communication.

REFERENCES

Boyd, S. and Vandenberghe, L. (2018). *Introduction to Applied Linear Algebra – Vectors, Matrices, and Least Squares*. Cambridge University Press, New York, NY, USA.

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. and Olshevsky, A. (2015). Distributed optimization over time-varying directed graphs. *IEEE Transactions on Automatic Control*, 60(3), 601–615.

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., Van Scoy, B., and Lessard, L. (2019). Analysis and design of first-order distributed optimization algorithms over time-varying graphs. *arXiv:1907.05448*.

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 optimizer of the global objective function.

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

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

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

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

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. Defining the parameter $\lambda := \sqrt{1-\rho^2}$, we can then write our algorithm in vectorized form as

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

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 gradient operator $\bar{\nabla} f : \mathbb{R}^{nd} \to \mathbb{R}^{nd}$ are defined as[1]

$$\mathcal{W}^k := \prod_{\ell=1}^{m}\big(W^{k,\ell} \otimes I_d\big) \quad \text{and} \quad \bar{\nabla} f(\mathbf{v}) := \begin{bmatrix} \nabla f_1(v_1) \\ \vdots \\ \nabla f_n(v_n) \end{bmatrix}.$$

Fixed-point. Define the points $\mathbf{u}^\star, \mathbf{v}^\star, \mathbf{x}^\star, \mathbf{y}^\star \in \mathbb{R}^{nd}$ as $\mathbf{v}^\star = \mathbf{x}^\star = \mathbb{1} \otimes x^\star$, $\mathbf{u}^\star = \mathbf{v}^\star - \alpha\bar{\nabla} f(\mathbf{v}^\star)$, $\mathbf{y}^\star = \frac{1}{\lambda}(\mathbf{u}^\star - \mathbf{x}^\star)$. 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 doubly-stochastic at each step. Also, $\mathsf{avg}(\mathbf{y}^\star) = 0$ since $x^\star$ satisfies (3).

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 gradient operator $\bar{\nabla} f$ satisfies

$$\mathsf{avg}\big(\bar{\nabla} f(\mathbf{x}^\star)\big) = 0$$

and

$$\|\mathbf{x} - \mathbf{x}^\star - \alpha\,(\bar{\nabla} f(\mathbf{x}) - \bar{\nabla} f(\mathbf{x}^\star))\| \leq \rho\,\|\mathbf{x} - \mathbf{x}^\star\| \tag{A.3}$$

for all $\mathbf{x} \in \mathbb{R}^{nd}$. In other words, $I - \alpha\,\bar{\nabla} f$ 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

$$\|\mathsf{dis}\big(\mathcal{W}^k(\mathbf{x})\big)\| \leq \sigma^m\,\|\mathsf{dis}(\mathbf{x})\| \leq \sigma_0\,\|\mathsf{dis}(\mathbf{x})\| \tag{A.4}$$

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

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

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

$$\mathsf{avg}(\bar{\mathbf{x}}^k) = \mathsf{avg}(\bar{\mathbf{v}}^k) \quad \text{for all } k \geq 0. \tag{A.5}$$

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,

$$\mathsf{avg}(\bar{\mathbf{y}}^k) = 0 \quad \text{for all } k \geq 0. \tag{A.6}$$

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

$$V(\bar{\mathbf{x}}, \bar{\mathbf{y}}) := \|\mathsf{avg}(\bar{\mathbf{x}})\|^2 + \begin{bmatrix} \mathsf{dis}(\bar{\mathbf{x}}) \\ \mathsf{dis}(\bar{\mathbf{y}}) \end{bmatrix}^\mathsf{T} \left(\begin{bmatrix} 1 & \lambda \\ \lambda & \lambda \end{bmatrix} \otimes I_{nd}\right) \begin{bmatrix} \mathsf{dis}(\bar{\mathbf{x}}) \\ \mathsf{dis}(\bar{\mathbf{y}}) \end{bmatrix} \tag{A.7}$$

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.7) 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.5) and (A.6), we have

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

The first term is nonpositive since $\bar{\nabla} f$ satisfies (A.3), the second since $\mathcal{W}^k$ satisfies (A.4), 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 $k \geq 0$. □

[1] We use the over-bar in $\bar{\nabla} f$ to distinguish it from the gradient of the global objective function f in (1). The operators are related by $(\frac{1}{n}\mathbb{1}^\mathsf{T} \otimes I_d)\bar{\nabla} f(\mathbb{1} \otimes x) = \nabla f(x)$.