Logarithmic Communication for Distributed Optimization in Multi-Agent Systems

Palma London
California Institute of Technology
Pasadena, California, USA
plondon@caltech.edu

Shai Vardi
Purdue University
West Lafayette, Indiana, USA
svardi@purdue.edu

Adam Wierman
California Institute of Technology
Pasadena, California, USA
adamw@caltech.edu

ABSTRACT

Classically, the design of multi-agent systems is approached using techniques from distributed optimization such as dual descent and consensus algorithms. Such algorithms depend on convergence to global consensus before any individual agent can determine its local action. This leads to challenges with respect to communication overhead and robustness, and improving algorithms with respect to these measures has been a focus of the community for decades.

This paper presents a new approach for multi-agent system design based on ideas from the emerging field of local computation algorithms. The framework we develop, LOcal Convex Optimization (LOCO), is the first local computation algorithm for convex optimization problems and can be applied in a wide-variety of settings. We demonstrate the generality of the framework via applications to Network Utility Maximization (NUM) and the distributed training of Support Vector Machines (SVMs), providing numerical results illustrating the improvement compared to classical distributed optimization approaches in each case.

CCS CONCEPTS

• **Theory of computation → Convex optimization**; • **Computing methodologies → Multi-agent systems**; *Distributed algorithms.*

KEYWORDS

distributed algorithms; distributed optimization; multi-agent systems

ACM Reference Format:

Palma London, Shai Vardi, and Adam Wierman. 2020. Logarithmic Communication for Distributed Optimization in Multi-Agent Systems. In *ACM SIGMETRICS / International Conference on Measurement and Modeling of Computer Systems (SIGMETRICS '20 Abstracts), June 8–12, 2020, Boston, MA, USA.* ACM, New York, NY, USA, 2 pages. https://doi.org/10.1145/3393691.3394197

1 OVERVIEW

This extended abstract summarizes [4], which introduces a novel approach for distributed optimization in multi-agent systems based on ideas from an emerging area in theoretical computer science – *local computation algorithms* [8]. The method proposed allows distributed agents to compute a local action or estimate with exponentially reduced communication and significantly improved robustness.

Distributed optimization is an area of crucial importance to the design and control of multi-agent systems. It provides a framework for the design of multi-agent systems where the system goal is formalized via a global objective and the distributed agents work together to solve this global optimization problem. Then, the agents determine their action by looking at the piece of the global solution associated with them. Importantly, in this framework an agent's goal is to determine its own action, i.e., its piece of the global solution. *It does not need to know the full global solution.*

Settings where distributed optimization has been used in the design of multi-agent systems are numerous and varied. For example, recently such approaches have become prominent in the emerging field of federated machine learning [3, 5], where data is distributed across a set of agents and the goal of the agents is to train a model using the full data set without sharing data between them, in part due to privacy concerns.

Distributed optimization is a field with a long history. Beginning in the 1960s approaches emerged for solving large scale linear programs via decomposition into pieces that could be solved in a distributed manner. Often these methods employ consensus schemes as a mechanism for distributing the computation among the processing units, forming the basis for many first order and second order distributed optimization algorithms, e.g., [1, 6].

Despite a wide variety of approaches to distributed optimization in multi-agent systems, the approaches that are studied and used today are similar at a high level – and this similarity leads to fundamental limitations on their scalability and robustness. In particular, all the approaches listed above, at their core, pass current estimates of the global solution between agents in a sequential process, gradually improving those estimates at each step with the goal of convergence to a (near) optimal solution, i.e., consensus. Universally in such approaches, the distributed agents are required to store, update, and broadcast a vector of dimension that matches that of the full system-wide solution to the problem at each step, which for multi-agent systems in modern applications can be enormous. Further, no individual agent can determine its own action or estimate without global convergence of all agents in the network. This is a result of the fact that distributed optimization algorithms are designed to allow each distributed agent to compute the full global solution. But, this is overkill for multi-agent systems, where typically an agent needs only to compute its local piece of the solution in order to determine its action.

SIGMETRICS '20 Abstracts, June 8–12, 2020, Boston, MA, USA

ACM ISBN 978-1-4503-7985-4/20/06.
https://doi.org/10.1145/3393691.3394197

As a result, there are a number of serious and fundamental challenges when it comes to applying distributed optimization algorithms in the design of multi-agent distributed systems.

First, since the network size could be enormous, consisting of tens or hundreds of thousands of distributed agents if we consider emerging internet of things (IoT) applications, the communication and storage demands for each iteration may be extreme. In fact, for most such approaches, e.g., consensus-style approaches, the communication within a single round requires $O(n)$ messages, typically containing a current estimate of the global solution. There has been considerable research that seeks to reduce the communication overhead of these approaches, e.g., [7, 9]. These approaches seek to partition the global solution into multiple blocks, each of which can be communicated less frequently, thus lowering the communication overhead. However, to this point, order-of-magnitude improvements have not been possible for general classes of optimization problems.

Second, the iterative convergence of traditional distributed optimization algorithms means that the convergence of *all nodes* can be delayed if a single node or communication link is congested. For example, if there is communication lag in one part of the network, a consensus algorithm cannot reach consensus, and thus no agent in the network can determine its local action. Such "stragglers" are frequent in modern distributed systems and lead to significant delays in many distributed optimization designs. The importance of this issue has been recognized for decades, and there has been considerable work toward developing asynchronous approaches for dual descent and consensus algorithms, e.g., [10]. However, even asynchronous algorithms require all nodes to communicate repeatedly in order for consensus to be achieved. Thus, if a set of agents is suffering from poor communication conditions, agents across the network must still wait for that part of the network to converge in order to determine their actions.

Third, classical approaches result in designs where any changes in network structure due to communication links failing or agents entering/leaving the network means that the algorithm is brought to a halt and needs to restart the convergence process. Again, this is a long-standing issue and the design of fault tolerant distributed optimization has received considerable attention. Robustness to failures and changes in the system are typically addressed through the design of fault-tolerant, Byzantine distributed optimization approaches, e.g., [2], however such approaches require significant adjustments to the classical algorithms and come at significant expense in terms of convergence rates and optimality guarantees.

Fourth, because classical distributed algorithms require *global* convergence/consensus before any individual agent can determine its local action, a single agent computing its individual action or estimate imposes communication and computation demands on every agent in the network. This introduces significant, unnecessary overhead and delay since it means that an individual agent is impacted by stragglers, agents entering/exiting, etc., across the *whole system* even though it only seeks to compute its *local* action. Ideally, an agent would be able to compute its part of the solution without the need to compute the full global solution.

Contributions. In [4] we seek to develop a new approach for distributed optimization in multi-agent systems that can reduce the communication overhead of traditional approaches, while also guaranteeing robustness to communication delay and failures in the system. To accomplish this, we seek a design that allows an individual agent to compute its local optimal action without the need for global communication.

Our approach toward achieving this goal is to develop a novel connection between distributed optimization and an emerging subfield of theoretical computer called *local computation algorithms* (LCAs) [8] – applying local computation algorithms to optimization problems for the first time.

The defining property of local computation algorithms is that they seek to compute a local "piece" of the solution to some algorithmic problem using only information that is "close" to that "piece" of the problem. For example, an LCA for matching allows each node in the graph compute its own match locally by communicating only with a small neighborhood of other "local" nodes, never computing a full matching for the graph. Yet, if all nodes run the LCA, then the solution each node computes is part of the same global matching.

In this work we develop the first local computation algorithm for convex optimization, LOCO (LOcal Convex Optimization). This optimization framework represents a fundamentally new approach for distributed optimization in multi-agent systems that allows an individual agent to compute its action with *exponentially less communication* than traditional approaches, while maintaining robustness to both stragglers and the entrance/exit of agents into the system. Further, LOCO allows an individual agent to compute its action or estimate without the need for global convergence, and thus without the need for global communication and computation.

2 ACKNOWLEDGMENTS

This work was supported in part by NSF grants AitF-1637598, CNS-1518941, CPS-154471, the Linde Institute, and an Amazon Fellowship in Artificial Intelligence.

REFERENCES

[1] Vincent D. Blondel, Julien M. Hendrickx, Alex Olshevsky, and John N. Tsitsiklis. 2005. Convergence in multiagent coordination, consensus, and flocking. In *Proc. of IEEE Conference on Decision and Control*. 2996–3000.

[2] Yudong Chen, Lili Su, and Jiaming Xu. 2017. Distributed Statistical Machine Learning in Adversarial Settings: Byzantine Gradient Descent. *Proc. ACM Meas. Anal. Comput. Syst. SIGMETRICS* 1, 2 (2017).

[3] Jakub Konecný, H. Brendan McMahan, and Daniel Ramage. 2015. Federated Optimization: Distributed Optimization Beyond the Datacenter. *ArXiv* abs/1511.03575 (2015).

[4] Palma London, Shai Vardi, and Adam Wierman. 2019. Logarithmic Communication for Distributed Optimization in Multi-Agent Systems. *ACM SIGMETRICS* 3, 3 (2019), 1–29.

[5] Brendan McMahan and Daniel Ramage. Accessed: 2017-04-10. Federated learning: Collaborative machine learning without centralized training data. https://research.googleblog.com/2017/04/federated-learning-collaborative.html.

[6] Angelia Nedić and Asuman Ozdaglar. 2010. Convergence rate for consensus with delays. *Journal of Global Optimization* 47, 3 (2010), 437–456.

[7] Angelia Nedic, Asuman Ozdaglar, and Pablo A. Parrilo. 2010. Constrained Consensus and Optimization in Multi-Agent Networks. *IEEE Trans. Automat. Control* 55, 4 (2010).

[8] Ronitt Rubinfeld, Gil Tamir, Shai Vardi, and Ning Xie. 2011. Fast Local Computation Algorithms. In *Proc. 2nd Sym. on Innov. in Computer Science (ICS)*. 223–238.

[9] Wei Shi, Qing Ling, Gang Wu, and Wotao Yin. 2015. A Proximal Gradient Algorithm for Decentralized Composite Optimization. *IEEE Transactions on Signal Processing* 63, 22 (2015).

[10] John N. Tsitsiklis, Dimitri P. Bertsekas, and Michael Athans. 1986. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE Trans. on Autom. Control* 31 (1986), 803–812.