Simple Synchronous and Asynchronous Algorithms for Distributed Minimax Optimization

Kenta Hanada *, Ryosuke Morita **, Takayuki Wada *, and Yasumasa Fujisaki *

Abstract: Synchronous and asynchronous algorithms are presented for distributed minimax optimization. The objective here is to realize the minimization of the maximum of component functions over the standard multi-agent network, where each node of the network knows its own function and it exchanges its decision variable with its neighbors. In fact, the proposed algorithms are standard consensus and gossip based subgradient methods, while the original minimax optimization is recast as minimization of the sum of component functions by using a $p$-norm approximation. A scalable step size depending on the approximation ratio $p$ is also presented in order to avoid slow convergence. Numerical examples illustrate that the algorithms with this step size work well even in the high approximation ratios.

Key Words: distributed algorithms, minimax optimization, synchronous algorithms, asynchronous algorithms.

1. Introduction

Multi-agent based distributed algorithms are well studied in order to deal with large scale or complex systems. The distributed algorithms consist of agents and networks. Each agent holds a decision variable, which is also called a state, to represent a condition of the system. The network is formed to describe a relationship among the agents whether agents can communicate with each other. It can be expressed as a graph, where nodes are agents and edges are communication links.

One of the simplest subjects of multi-agent based distributed algorithms is an averaging consensus [1]–[3]. The purpose of the averaging consensus is to achieve the average value of the initial state of the nodes. Practical applications are proposed with the averaging consensus to achieve an average value of sensors [4] and to synchronize clock time [5]. Distributed optimization is another subject of multi-agent based algorithms, where the nodes try to minimize a sum of their functions in a distributed manner. In general, these algorithms assume that nodes can exchange only their decision variables so that each node is able to keep its privacy. Many synchronous and asynchronous algorithms are developed and analyzed for standard distributed optimization [6]–[11].

This study focuses on distributed minimax optimization, which is another subject of the distributed algorithms. The distributed minimax optimization is defined as minimization whose objective function returns the maximum value of the set of functions at a certain value. We can minimize the worst case cost by the distributed minimax optimization. Thus, we can consider the robustness of the networked systems. In the problem, we assume that each node in a graph has its own function which consists of the global objective function. We also assume that each node exchanges only decision variables to its neighbors of the graph because the nodes have motivation not to reveal their value of the objective function due to the privacy issue. Notice here that, in order to get the optimal solution of the original minimax optimization, since all of the nodes have to evaluate the maximum value over the functions, the nodes must exchange their values of the functions as well as their decision variables in principle. In the existing algorithms to solve the original minimax optimization [12], [13], each node has to exchange further information in addition to the decision variable. Otherwise, we have to employ a non-standard distributed algorithm [14].

In this paper, we propose an approximation of the distributed minimax optimization to overcome the above drawbacks. Here we recast the original minimax optimization as minimization of the sum of component functions by using a $p$-norm approximation. Such a recast enables us to solve the optimization without exchanging the value of the objective function. We consider a synchronous networked algorithm and an asynchronous gossip algorithm for the approximated minimax optimization. Since $p$ represents an approximation ratio against the original minimax optimization, the algorithms can achieve good quality solution when a large number is given to $p$. On the other hand, a large number of $p$ often causes a solution oscillation and decreases the convergence speed of the algorithms. We therefore propose a scalable step size depending on the approximation ratio $p$ of the problem for the synchronous and asynchronous algorithms. Numerical examples show that the algorithms are able to achieve a good approximated solution to the actual optimal one.

The rest of this paper is organized as follows. In Section 2, we formulate the distributed minimax optimization problem and its approximation. In Section 3, we establish a synchronous networked algorithm for the approximated minimax optimization problem. We also establish an asynchronous gossip algorithm with the subgradient method for it in the next section. In Section 5, numerical examples illustrate that the proposed algorithm works well even in high approximation ratios.

The preliminary versions of this study were presented as po-
2. Distributed Minimax Optimization

In this paper, we tackle distributed minimax optimization over a multi-agent network. Let $V = \{1, \ldots, N\}$ be a set of nodes, where $N$ is the number of nodes, and $E \subseteq V \times V$ is the set of edges. We assume that the graph $G = (V, E)$ is undirected and connected. The distributed minimax optimization is defined as

$$J = \min_{x \in X} \max_{i \in S} f_i(x),$$

(1)

where $f_i : \mathbb{R}^m \to \mathbb{R}$ is an objective function at each node $i \in V$, $x \in \mathbb{R}^m$ is a global (common) decision variable, and $X \subseteq \mathbb{R}^m$ represents a constraint on $x$. We assume that the function $f_i$ is convex and subdifferentiable with respect to $x$ for all $i \in V$. Furthermore, $f_i$ takes a non-negative value for any $x$, i.e.,

$$f_i(x) \geq 0$$

(2)

for all $x \in X$ and $i \in V$. Node $i$ only holds its own function $f_i$ and it never knows the other objective functions $f_j$, where $j \in V \setminus \{i\}$. We also assume that the set $X$ is compact, convex, and known to all the nodes, and the optimal set $X^\ast$ is non-empty.

In order to get the optimal minimax value of (1), all of the nodes must evaluate $\max_{1 \leq i \leq N} f_i(x)$, as an approximation of (1), which is the main contribution of this paper. Here $p \in \mathbb{N}$ is a fixed number which we call the approximation ratio. We see that the optimization (3) is in fact an approximation of (1) as follows.

We therefore introduce an optimization problem

$$J_p = \min_{x \in X} \sum_{i=1}^N f_i(x),$$

(3)

as an approximation of (1), which is the main contribution of this paper. Here $p \in \mathbb{N}$ is a fixed number which we call the approximation ratio. We see that the optimization (3) is in fact an approximation of (1) as follows.

We first recall a well-known fact about $p$-norm. That is,

$$\|y\|_p \leq \|y\|_1 \leq \|y\|_\infty,$$

holds true for any vector $y = [y_i] \in \mathbb{R}^N$ and any $p \geq 1$, where the norms are defined by

$$\|y\|_p = \left( \sum_{i=1}^N |y_i|^p \right)^{1/p}, \quad \|y\|_\infty = \max_{1 \leq i \leq N} |y_i|.$$

In fact, we can immediately obtain the inequalities

$$\|y\|_\infty = \|y_M\| = (\|y_M\|^p)^{1/p} \leq \left( \sum_{i=1}^N |y_i|^p \right)^{1/p} = \|y\|_p,$$

$$\|y\|_p \leq \left( \sum_{i=1}^N |y_i|^p \right)^{1/p} \leq (N|y_M|^p)^{1/p} = N^{1/p} |y_M| = N^{1/p} \|y\|_\infty,$$

where $M = \arg \max_{1 \leq i \leq N} |y_i|$, that is, $|y_M| = \max_{1 \leq i \leq N} |y_i|$. Since $N^{1/p} \to 1$ as $p \to \infty$ for any $N \geq 1$, it turns out that $\|y\|_p \to \|y\|_\infty$ as $p \to \infty$. The above fact together with non-negativity of $f_i$ implies that

$$\min_{x \in X} \max_{1 \leq i \leq N} f_i(x) \leq \min_{x \in X} \left( \sum_{i=1}^N (f_i(x))^p \right)^{1/p} \leq N^{1/p} \min_{x \in X} \max_{1 \leq i \leq N} f_i(x),$$

min $\sum_{i=1}^N (f_i(x))^p \to \min \max f_i(x)$ as $p \to \infty$.

Since $\min \sum_{i=1}^N (f_i(x))^p$ and $\min \sum_{i=1}^N (f_i(x))^p$ have the same optimal solution, we finally see that the optimization (3) is in fact an approximation of (1). That is, the optimal value $J$ of the approximated problem (3) is related to the optimal value $J_p$ of the original problem (1) as

$$J \leq J_p \leq N^{1/p} J,$$

$$J \to J_p \to J \quad \text{as} \quad p \to \infty.$$

We remark that the optimization (3) is the form of a standard distributed minimization [6] whose objective function is the sum of component functions. Notice here that the function $\bar{f}_i$ is convex for any $p$ and for any convex function $f_i$ which takes non-negative value [17]. We therefore see that we can apply any existing distributed algorithms, e.g., the methods developed in [6],[7], to the approximated problem (3) under some assumptions. We now assume that subgradients $g_i(x)$ of $f_i$ at $x$ are uniformly bounded over $X$, that is, there exists a positive scalar $C$ such that $\sup_{x \in X} \|g_i(x)\| \leq C$ for all $i \in V$, where $g_i(x)$ is said to be a subgradient of $f_i$ at $x$ if

$$\bar{f}_i(y) \geq \bar{f}_i(x) + g_i(x)^\top (y - x)$$

holds for all $y$. The details of algorithms will be shown in the following sections.

3. A Synchronous Networked Algorithm with the Subgradient Method

In this section, we consider a synchronous networked algorithm with the subgradient method for the approximated distributed minimax optimization (3).

Let us define the set of neighboring nodes of node $i$ as $\mathcal{N}_i = \{ j \in V | (i, j) \in E \}$. We assume that node $i$ can communicate with only neighboring nodes. Since the graph $G$ is undirected, node $j$ can also communicate with node $i$ if node $j$ is in $\mathcal{N}_i (i \in \mathcal{N}_j)$. To find the optimal solution of the problem (3), we use a standard updating rule

$$v_i[k] = \sum_{j=1}^N a_{ij} x_j[k],$$

(4)

$$x_i[k + 1] = P_X (v_i[k] - r[k]g_i(x_i[k])),$$

(5)

where $k \in \mathbb{N}$ is the $k$-th tick of the synchronous clock, $x_i[k] \in \mathbb{R}^m$ is an estimated optimal solution of (3) which is held by node $i$ at the $k$-th tick, $P_X$ is the Euclidean projection on the set $X$, $r[k] \in \mathbb{R}$ is a step size, $g_i(x_i[k])$ is the subgradient of $\bar{f}_i$ at $x_i[k]$, and $a_{ij} \in \mathbb{R}$ is the weight of node $i$. If node $i$ gets the information from a neighboring node $j$, $a_{ij}$ is positive. Otherwise, $a_{ij} = 0$. We assume that the weight matrix $A = [a_{ij}]$ satisfies $a_{ij} \geq 0$, $\sum_{j=1}^N a_{ij} = 1$, and $\sum_{i=1}^N a_{ij} = 1$, i.e., $A$ is a doubly stochastic matrix.

For this updating rule, we employ the diminishing step size

$$r[k] = \frac{1}{k}.$$  

(6)

It is known that each $x_i[k]$ converges to the optimal solution of (3) by the updating rule (4) and (5) [6],[18].
One important issue on the above updating rule is convergence speed. In our context, a large $p$ is preferable for a good approximation. However, according to (3), the larger $p$ is, the much larger (or smaller) a value of the subgradient is at a certain $x$. This may invoke a slow convergence. To deal with this issue, we propose a scalable step size depending on the approximation ratio $p$ as

$$r[k] = 10^{-5} \cdot \frac{1}{k}. \quad (7)$$

This scalable step size is effective to reduce oscillations of the second term of (5) in the right hand side, which will be demonstrated in Section 5.

4. An Asynchronous Gossip Algorithm with the Subgradient Method

In this section, we consider an asynchronous gossip algorithm with the subgradient method for the approximated distributed minimax optimization (3).

The gossip algorithm is an asynchronous algorithm to broadcast information. We apply it to the approximated minimax optimization (3) as follows.

Although it is an asynchronous algorithm, it is easier to understand the algorithm with a single virtual clock [2],[7]. We assume that each node $i$ has a local clock which ticks at the times of a rate 1 Poisson process. Equivalently, we can consider that the virtual clock that ticks according to a Poisson process with rate $N$. In this section, $k$ represents the $k$-th tick of the virtual clock, and let $i$ be the node whose local clock actually ticked at the time. Then node $i$ chooses its neighboring node $j \in \mathcal{N}_i$ randomly, and node $i$ and $j$ exchange their value of $x$ at the time and they update their value with each other. Note that node $j$ is chosen with the same probability.

We consider an updating rule in the gossip algorithm

$$\bar{x}[k] = \bar{x}[k] = \frac{x_i[k] + x_j[k]}{2}, \quad (8)$$

$$x_i[k + 1] = P_X(\bar{x}[k] - r_i[k]g_i(x_i[k])), \quad (9)$$

$$x_j[k + 1] = P_X(\bar{x}[k] - r_j[k]g_j(x_j[k])), \quad (10)$$

$$x_h[k + 1] = x_h[k], \quad h \in V \setminus \{i, j\}, \quad (11)$$

where $r_i[k] \in \mathbb{R}$ is a local step size that node $i$ holds in $k$-th tick. The algorithm starts at $k = 0$. The nodes $i$ and $j$ update their values according to (8), (9), and (10). Nodes except $i$ and $j$ maintain their values (11). Then, the algorithm goes to next tick.

When we select the diminishing step size

$$r_i[k] = \frac{1}{c_i[k]}, \quad (12)$$

it is known that the estimated optimal solutions converge to the optimal by the algorithm [7], where $c_i[k]$ is the number of ticks of the local clock for node $i$ at $k$-th tick in the virtual clock. Formally, we denote it as

$$c_i[k] = \begin{cases} c_i[k - 1] + 1 & \text{if the local clock ticks}, \\ c_i[k - 1] & \text{otherwise}, \end{cases}$$

where $c_i[0] = 0$ for all $i \in V$.

As we mentioned before, the convergence speed may be slow due to the large number of the subgradient as is the case with the synchronous algorithm. Thus we propose a scalable step size depending on the approximation ratio $p$ as

$$r_i[k] = 10^{-5} \cdot \frac{1}{c_i[k]}. \quad (13)$$

Its effectiveness will be demonstrated in the next section.

5. Numerical Examples

Let us consider a multi-agent system having four nodes ($V = \{1, 2, 3, 4\}$) whose objective functions are

$$f_1(x) = \max \left\{ 0, \frac{1}{3} \right\}, \quad f_2(x) = \max \left\{ 0, \frac{2}{3} x \right\},$$

$$f_3(x) = \max \left\{ 0, -x + 5 \right\}, \quad f_4(x) = \frac{1}{4} (x - 1)^2.$$

Note that the optimal solution for these instance in (1) is 3.00. The set $E$ of edges of the network is defined as

$$E = \{(1, 2), (2, 3), (3, 4), (1, 4)\},$$

that is, the system has a ring topology. Figure 1 shows the topology of the network. We use $m = 1$, that is, the decision variable $x$ is a scalar. The weight matrix $A$ for the synchronous algorithm is selected as

$$A = \begin{bmatrix} 0.998 & 0.001 & 0.000 & 0.001 \\ 0.001 & 0.996 & 0.003 & 0.000 \\ 0.000 & 0.003 & 0.995 & 0.002 \\ 0.001 & 0.000 & 0.002 & 0.997 \end{bmatrix}.$$  

The initial condition was chosen as

$$x_1[1] = 0.5, \quad x_2[1] = 1.5, \quad x_3[1] = 2.5, \quad x_4[1] = 3.5.$$  

Under these settings, we executed 100,000 iterations according to the proposed algorithm.

When $r_i[k] \equiv 0$, the algorithm becomes a simple consensus algorithm. Figure 2 shows that the algorithm achieved the average consensus. Note that the vertical axes of all of the figures except Fig. 1 represent the value of the estimated optimal solution $x_i[k]$ for all $i \in V$, and the horizontal axes which are shown by the logarithmic scale represent the number $k$ of iterations.

When we choose $p = 1$, the objective function of (3) is reduced to the sum of $f_i$. Figure 3 shows that the result of $p = 1$
Fig. 3 Behavior of the synchronous multi-agent system when $p = 1$ with the step size (6).

Fig. 4 Behavior of the synchronous multi-agent system when $p = 1$ with the step size (7).

Fig. 5 Behavior of the asynchronous multi-agent system when $p = 1$ with the step size (12).

Fig. 6 Behavior of the asynchronous multi-agent system when $p = 1$ with the step size (13).

by the synchronous algorithm with the simple diminishing step size (6), Fig. 4 shows that the result of $p = 1$ by the synchronous algorithm with the scalable step size (7), Fig. 5 shows that the result of $p = 1$ by the asynchronous algorithm with the simple diminishing step size (12), and Fig. 6 shows that the result of $p = 1$ with the scalable step size (13). Note that the optimal solution for this instance in (3) ($p = 1$) is 1.00.

Figures 7, 11, and 15 show results of the synchronous algorithm with (6) in the case of $p = 2, 8$, and 128 respectively. Figures 8, 12, and 16 show results of the synchronous algorithm with (7) in the case of $p = 2, 8$ and 128 respectively. Figures 9, 13, and 17 show results of the asynchronous algorithm with (12) in the case $p = 2, 8$ and 128 respectively. Figures 10, 14, and 18 show results of the asynchronous algorithm with (13) in the case $p = 2, 8$ and 128 respectively. Note that the optimal solution for these instance in (3) ($p = 2, 8$, and 128) is 2.77, 3.07, and 3.01.

We should first point out that we can regard asynchronous algorithms as the synchronous ones in principle. Synchronous algorithms update all decision variables for all nodes. On the
Fig. 11 Behavior of the synchronous multi-agent system when $p = 8$ with the step size (6).

Fig. 12 Behavior of the synchronous multi-agent system when $p = 8$ with the step size (7).

Fig. 13 Behavior of the asynchronous multi-agent system when $p = 8$ with the step size (12).

Fig. 14 Behavior of the asynchronous multi-agent system when $p = 8$ with the step size (13).

other hand, asynchronous algorithms update decision variables for only 2 nodes at the same time. We need to remind these facts when we discuss about the convergence speed.

For small $p$ ($p = 1, 2$), it seems that there is no difference for the convergence speed between the simple diminishing step size and the proposed step size while asynchronous algorithms are better than synchronous ones in terms of the convergence speed. For large $p$ ($p = 8, 128$), we see that the convergence speed of algorithms with the simple diminishing step size is quite slow (Figs. 11, 13, 15, and 17). In contrast, the algorithms with the scalable step size converge to the optimal solution quickly even for large $p$ (Figs. 12, 14, 16, and 18). The convergence speed of the synchronous algorithm with the scalable step size (13) are better than the asynchronous one of the simple diminishing step size (12), but worse than the one with the scalable step size (13). Thus, we see that the scalable diminishing step size (7) and (13) are effective to enhance the convergence speed of the algorithms.
6. Conclusion

In this paper, we proposed an approximation problem of the distributed minimax optimization. We applied a synchronous networked algorithm and an asynchronous gossip algorithm with a subgradient method to approximated minimax optimization. We then proposed a scalable step size for the subgradient method by using its approximation ratio. Numerical examples illustrated that the proposed step size works well even in high approximation ratios.

Acknowledgments

The authors would like to thank Toru Asai, Izumi Masubuchi, and Tatsuhito Tsuchiya for useful discussions and beneficial comments. This research was supported by JST CREST Grant Number JPJIMCR15K2, Japan.

References

[1] D. Kempe, A. Dobra, and J. Gehrke: Gossip-based aggregation of information, Proceedings of the 44th Annual IEEE Symposium on Foundations of Computer Science, pp. 482–491, 2003.
[2] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah: Randomized gossip algorithms, IEEE Transactions on Information Theory, Vol. 52, No. 6, pp. 2508–2530, 2006.
[3] S. Kar and J.M.F. Moura: Distributed consensus algorithms in sensor networks: Quantized data and random link failures, IEEE Transactions on Signal Processing, Vol. 58, No. 3, pp. 1383–1400, 2010.
[4] S. Kar and J.M.F. Moura: Distributed consensus algorithms in sensor networks with imperfect communication: Link failures and channel noise, IEEE Transactions on Signal Processing, Vol. 57, No. 1, pp. 355–369, 2009.
[5] L. Schenato and G. Gamba: A distributed consensus protocol for clock synchronization in wireless sensor network, Proceedings of the 46th IEEE Conference on Decision and Control, pp. 2289–2294, 2007.
[6] A. Nedić and A. Ozdaglar: Distributed subgradient methods for multi-agent optimization, IEEE Transactions on Automatic Control, Vol. 54, No. 1, pp. 48–61, 2009.
[7] S.S. Ram, A. Nedić, and V.V. Veeravalli: Asynchronous gossip algorithms for stochastic optimization, Proceedings of the 48th IEEE Conference on Decision and Control, pp. 3581–3586, 2009.
[8] S.S. Ram, A. Nedić, and V.V. Veeravalli: Asynchronous gossip algorithm for stochastic optimization: Constant stepsize analysis, Proceedings of the 14th Belgian-French-German Conference on Optimization, pp. 51–60, 2010.
[9] Z. Jiang, S. Sarkar, and K. Mukherjee: On distributed optimization using generalized gossip, Proceedings of the 54th IEEE Conference on Decision and Control, pp. 2667–2672, 2015.
[10] S. Lee and A. Nedić: Asynchronous gossip-based random projection algorithms over networks, IEEE Transactions on Automatic Control, Vol. 61, No. 4, pp. 953–968, 2016.
[11] J. Tsitsiklis, D. Bertsekas, and M. Athans: Distributed asynchronous deterministic and stochastic gradient optimization algorithms, IEEE Transactions on Automatic Control, Vol. 31, No. 9, pp. 803–812, 1986.
[12] K. Srivastava, A. Nedić, and D. Stipanović: Distributed minimax optimization in networks, Proceedings of the 17th International Conference on Digital Signal Processing, pp. 1–8, 2011.
[13] I. Masubuchi, T. Wada, R. Morita, T. Asai, Y. Ohta, and Y. Fujisaki: Distributed multi-agent optimization based on a constrained subgradient method, SICE JCMSI, Vol. 8, No. 3, pp. 234–240, 2015.
[14] I. Masubuchi, T. Wada, T. Asai, L.T.H. Nguyen, Y. Ohta, and Y. Fujisaki: Distributed multi-agent optimization based on an exact penalty method with equality and inequality constraints, SICE JCMSI, Vol. 9, No. 4, pp. 179–186, 2016.
[15] R. Morita and Y. Fujisaki: Distributed minimax optimization via multi-agent systems, Proceedings of the SICE Annual Conference 2013, pp. 1117–1119, 2013.
[16] K. Hanada, T. Wada, and Y. Fujisaki: Distributed minimax optimization by an asynchronous gossip algorithm with a subgradient method, Proceedings of the SICE Annual Conference 2016, pp. 1613–1616, 2016.
[17] R.T. Rockafellar: Convex Analysis, Princeton University Press, 1970.

Kenta Hanada (Member)

He received his B.S. and M.S. degrees in Maritime Science and Technology, Ph.D. degree in engineering from Kobe University, Kobe, Japan, in 2011, 2013, and 2016, respectively. In 2016, he joined Osaka University, where he is currently a Specially Appointed Assistant Professor (Full time) in Graduate School of Information Science and Technology, Osaka University, Suita, Japan. His research interests include distributed optimization.

Ryosuke Morita (Member)

He received the B.S. degree in engineering, the M.S. and Ph.D. degree in informatics from Kyoto University, Uji, Japan in 2008, 2010, and 2013 respectively. From 2013 to 2014, he was a Specially Appointed Researcher in Graduate School of Information Science and Technology, Osaka University, Suita, Japan. From 2014 to 2015, he was an Assistant Professor of the College of Science and Engineering, Aoyama Gakuin University, Sagamihara, Japan. He is currently an Assistant Professor of the Faculty of Engineering, Gifu University, Gifu, Japan.

Takayuki Wada (Member)

He received the B.E., M.E., and Ph.D. degrees in engineering from Kobe University, Kobe, Japan, in 2004, 2006, and 2009, respectively. During 2008 and 2010, he was a Research Fellow of Japan Society for the Promotion of Science. From 2010 to 2012, he was an Assistant Professor in Suzuka National College of Technology, Suzuka, Japan. He is currently an Assistant Professor in Graduate School of Information Science and Technology, Osaka University, Suita, Japan. His research interests involve in probabilistic approach to analysis and design of control systems.

Yasumasa Fujisaki (Member)

He received his B.S., M.S., and Ph.D. degrees in engineering from Kobe University, Kobe, Japan, in 1986, 1988, and 1994, respectively. From 1988 to 1991, he was a Research Member of the Electronics Research Laboratory, Kobe Steel, Ltd., Kobe, Japan. From 1991 to 2010, he was with Kobe University and served as a Research Associate and an Associate Professor. Since 2010, he has been a Professor of the Department of Information and Physical Sciences, Osaka University, Suita, Japan. He served as an Associate Editor of Automatica from 2001 to 2012 and an Editor of SICE Journal of Control, Measurement, and System Integration from 2011 to 2017. His research interests include analysis and design of large scale systems.