DISTRIBUTED OPTIMIZATION AND MACHINE LEARNING OVER NETWORKS

ECE 6970 PROJECT REPORT

Polina Alexeenko [*]
pa357@cornell.edu

Jiahe Chen [*]
jc3472@cornell.edu

Zhaopeng Xu [*]
zx273@cornell.edu

December 22, 2020

ABSTRACT

In many engineering applications, multiple intelligent agents are physically deployed and coordinated to achieve a common goal. Such multi-agent optimization system can be implemented in a distributed manner, which is more robust and may require much less communication resources compared with its centralized counterpart. In the general framework of distributed optimization, all or subset of nodes in a time-varying network cooperate to optimize an overall objective function. In this report, we present an overview of the design and the analysis of distributed optimization, focusing on three foundational papers in this field [1] [2] [3]. The reported papers reveal that the criteria of convergence to the optimum and the convergence rate depend on numerous factors, including the network size, desired level of accuracy, network topology, communication protocols, updating rules, noises, constraints, etc. Additionally, we comment on some extensions of the fundamental results in the papers and propose several potential applications.

1 Introduction

The rapid evolution of communication and sensing technologies has resulted in the emergence of large-scale networked systems in many engineering applications such as power systems, sensor networks, swarm robotics, and manufacturing. Because these networks are time-varying, involve a large number of agents, and are comprised of agents with limited access to central communication, centralized coordination of the networks is limited in its effectiveness. Instead, these networks should be controlled through distributed algorithms which are robust to changes in network topology and able to scale efficiently with network size. In this report, we discuss an approach to the problem of distributed optimization over networks based on the use of subgradient methods. The remainder of the report is organized as follows. In Section2, we describe the basic problem setting and present classical results. Section3 discusses extensions of the basic setting to more sophisticated networks, e.g., networks where agents' decision making is subject to constraints and where communication is noisy. Section4 discusses limitation of existing methodology and directions for future work, and Section 5 concludes.

1.1 Notation

This report will use the following notation. We denote by x' the transpose of the column vector x and by $x'y$ the inner product of two vectors x and y. We will use $\|x\|$ to denote the Euclidean norm of x. We denote by $\text{dist}\,(x, X)$ the Euclidean distance between a vector x from a set X, i.e.,

$$\text{dist}\,(x, X) = \inf_{\bar{x} \in X} \|x - \bar{x}\| .$$

For a convex function f, we write the domain of f as $\text{dom}\,(f)$, where

$$\text{dom}\,(f) = \{x \in \mathbb{R}^n \mid f\,(x) < \infty\}$$

[*]School of Electrical and Computer Engineering, Cornell University

For a convex function f, we define the subgradient of f at a point $\bar{x} \in \text{dom}\,(f)$ to be a vector $s_f\,(\bar{x})$ satisfying the following:

$$f\,(\bar{x}) + s_f\,(\bar{x})'\,(x - \bar{x}) \leq f\,(x) \qquad \text{for all } x \in \text{dom}\,(f)\,.$$

Futhermore, we denote the set of all subgradients of f at $\bar{x}$ by $\partial f\,(\bar{x})$.

2 Preliminaries

In this section, we describe the distributed optimization algorithm presented in a seminal paper by Nedic and Ozdaglar [1] and their closely related survey [4]. Section2.1 describes the model of the network over which agents communicate and presents the algorithm by which agents attempt to minimize the global objective. In Section2.2, we discuss the convergence properties of the algorithm and characterize its performance with respect to the global optimum.

2.1 Agent Model

We consider a setting in which n agents communicate over a network. Each agent $i \in \{1, \ldots, n\}$ is associated with a convex function $f_i : \mathbb{R}^n \to \mathbb{R}$ which we assume is known only to agent i. The agents aim to cooperatively optimize the global objective function

$$\begin{aligned} &\text{minimize } \sum_{i=1}^{n} f_i\,(x) \\ &\text{subject to } x \in \mathbb{R}^n \end{aligned} \tag{1}$$

where x is a global decision vector.

In order to solve Problem 1, agents employ a subgradient based algorithm. At each time period, all agents maintain an estimate of the optimal decision. We denote agent i's estimate of the optimal decision at time t_k by x_i^k. The algorithm by which agents update their estimates of the optimal decision vector consists of two components: a “consensus step" and a subgradient step. In the consensus step, agents aggregate their opinions and those of their neighbors. In the subgradient step, agents move in the direction of the negative of their subgradients. Formally, at time t_{k+1} agent i updates their estimates according to

$$x_i^{k+1} = \sum_{i=1}^{n} A_{i,j}^k x_j^k - \alpha_i^k g_i^k. \tag{2}$$

Here, α_i^k denotes the step size used by agent i at time period t_k. Throughout the remainder of Section2, we will assume that the step sizes used by agents are equal and constant in time, and denote the step size by α. The scalars $A_{i,j}^k$ correspond to the weights each agent assigns to their opinions and to those of other agents at time t_k. Furthermore, we denote by A^k the matrix of weights $\left[A_{i,j}^k\right]_{i,j=1,\ldots,m}$. Throughout, we assume that $A_{i,i}^k$ is positive for all times t_k (i.e., each agent always considers their previous estimate in generating their new estimate). For $i \neq j$, $A_{i,j}^k > 0$ if agent j communicates with agent i at time t_k, and $A_{i,j}^k = 0$ otherwise.

Throughout this report, we assume that the network over which agents communicate obeys certain structural assumptions which we discuss below. It is worth noting that the model of information exchange between agents is based on the model proposed by Tsitsiklis [5] and subsequently studied by Blondel et al [6].

Assumption 1 (Assumptions 1 and 4 in [1]) *For all $k \geq 0$, the weight matrix A^k is doubly stochastic with positive diagonal. Additionally, there is a scalar $\eta > 0$ such that if $A_{i,j}^k > 0$, then $A_{i,j}^k \geq \eta$.*

Assumption 1 guarantees that in the long run, each agent's objective function receives the same weight, ensuring that agents optimize the sum of the functions rather than a weighted sum of the functions. Futhermore, we assume that the network is “frequently connected", i.e., that the information of any agent i eventually reaches all other agents and that the frequency with which agents communicate with their neighbors is bounded from below. The frequent connectivity requirement is stated formally in Assumption 2.

Assumption 2 (Assumption of Frequent Connectivity) *There exists an integer $B \geq 1$ such that the directed graph $\left(V, E^k \cap E^{k+1} \cdots \cap E^{k+B-1}\right)$ is strongly connected for all $k \geq 0$.*

2.2 Convergence behavior

Using the assumptions presented in the previous section, we can characterize the asymptotic behavior of the agent estimates. In order to make clearer the dependency of the estimates on the weight matrix A, we can re-write the update dynamics in terms of the products of the weight matrices. In particular, we define the *transition matrix* $\Phi(k,s)$ according to

$$\Phi(k,s) = A(k)\,A(k-1)\ldots A(s+1)\,A(s)\,. \tag{3}$$

We can now re-write the update rule in terms of the transition matrices as

$$x_i^{k+1} = \sum_{j=1}^{n} [\Phi(k,s)]_{ij}\, x_j^s - \alpha \sum_{r=s}^{k-1}\sum_{j=1}^{n} [\Phi(k,r+1)]_{ij}\, g_i^r - \alpha d_i^k. \tag{4}$$

Next, we characterize the asymptotic behavior of the estimates x_i^k by studying the behavior of the transition matrices $\Phi(k,s)$ as k approaches infinity.

Theorem 1 (Proposition 1 in [1]) *The limit matrices $\Phi(s) = \lim_{k\to\infty}\Phi(k,s)$ are doubly stochastic and correspond to a uniform steady state distribution for all s, i.e.,*

$$\Phi(s) = \frac{1}{n}ee' \quad \textit{for all } s. \tag{5}$$

Furthermore, the entries $[\Phi(k,s)]_i^j$ converge to $\frac{1}{n}$ as $k\to\infty$ with a geometric rate uniformly with respect to i and j, i.e., for all $i,j\in\{1,\ldots,n\}$,

$$\left|[\Phi(k,s)]_i^j - \frac{1}{m}\right| \le 2\frac{1+\eta^{-B_0}}{1-\eta^{B_0}}\left(1-\eta^{B_0}\right)^{\frac{k-s}{B_0}} \quad \textit{for all s and k with } k\ge s, \tag{6}$$

where $B_0 = (n-1)\,B$.

This result tells us that in the long run, agents converge to a consensus based on equal weighting of each agent's objective functions. Next, we will characterize the convergence properties of the subgradient algorithm 2. The analysis relies on the use of an assumption on the boundedness of subgradients associated with the agent cost functions f_i, which we state formally below.

Assumption 3 (Assumption 7 in [1]) *For some scalar $L>0$, we have for all $x\in\mathbb{R}^n$ and all i*

$$\|g\| \le L \quad \textit{for all } g\in\partial f_i(x)\,.$$

In order to study the behavior of the subgradient algorithm, we will define an auxiliary sequence $\left\{y^k\right\}$ as follows:

$$y^{k+1} = y^k - \frac{\alpha}{n}\sum_{i=1}^{n} g_i^k \tag{7}$$

where g_i^k is the subgradient of $f_i\left(x_i^k\right)$ used in the algorithm 2 and

$$y^0 = \frac{1}{n}\sum_{i=1}^{n} x_i^0. \tag{8}$$

We start by characterizing the difference between the original sequence of interest and the auxillary sequence.

Theorem 2 (Proposition 3a in [1]) *For every $i\in\{1,\ldots,n\}$, a uniform upped bound on $\left\|y^k - x_i^k\right\|$ is given by*

$$\left\|y^k - x_i^k\right\| \le 2\alpha L C_1 \quad \textit{for all } k\ge 0 \tag{9}$$

where $C_1 = 1 + \frac{n}{1-\left(1-\eta^{B_0}\right)\frac{1+\eta^{-B_0}}{1-\eta^{B_0}}}$.

With this result in hand, we proceed to a result characterizing the value of the objective function at the averages of the auxiliary sequence. That is, we define

$$\hat{y}^k = \frac{1}{k}\sum_{h=1}^{k} y^h \tag{10}$$

and provide a bound on the difference between the objective function value at $\hat{y}^k$ and the optimal function value.

Theorem 3 (Proposition 3b in [1]) *An upper bound on the objective cost $f\left(\hat{y}^k\right)$ is given by*

$$f\left(\hat{y}^k\right) \leq f^* + \frac{n dist\left(y\left(0\right), X^*\right)}{2\alpha k} + \frac{\alpha L^2 C}{2} \tag{11}$$

where X^* denotes the set of optimal solutions and $C = 1 + 8nC_1$.

Combining these two results, we are able to characterize the performance of the algorithm. In particular, we bound the value of the objective function at the time average of the iterates. Specifically, we define the vector $\hat{x}_i^k$ as follows

$$\hat{x}_i^k = \frac{1}{k}\sum_{h=1}^{k} x_i^h.$$

and bound the value of the objective function at $\hat{x}_i^k$.

Theorem 4 (Proposition 3b in [1]) *An upper bound on the objective value $f\left(\hat{x}_i^k\right)$ is given by*

$$f\left(\hat{x}_i^k\right) \leq f^* + \frac{m dist^2\left(y\left(0\right), X^*\right)}{\alpha k} + \alpha L\left(LC + 2mLC_1\right) \quad \textit{for all } k \geq 1.$$

It is important to note that the bound obtained in Theorem 4 is the sum of two terms. The first term of the bound decays at a rate $1/k$, while the second term is constant. The fact that the second term is constant comes from the fact that the algorithm uses a constant update rule. Subgradient algorithms can use various alternative step-size rules to ensure that the error decays to zero asymptotically (e.g., [7–9]). In the sequel, we will discuss a set of conditions on the choice of step jason schwartzman size resulting in asymptotic convergence.

3 Extension and Discussion

In this section, we present some extensions of the fundamental results reported in Section 1 and discuss their advantages and limitations. We consider a time-varying network described by a sequence of stochastic matrices A^0, A^1, A^2,... $[A_\alpha]$ denotes the *threshold matrix* of A, which is obtained from A by setting every element smaller than α to zero. Agents are represented by the vertex set $\{1, ..., n\}$ and links among agents are represented by the edge set $E_A = \{(i,j)|A_{i,j} > 0\}$. At iteration k, each agent i sends messages to its out-neighbors $N_i^{out,k} = \{j|A_{j,i}^k > 0\}$ and receives messages from its in-neighbors $N_i^{in,k} = \{j|A_{i,j}^k > 0\}$. Each agent has out-degree $d_i^{out,k} = |N_i^{out,k}|$ and in-degree $d_i^{in,k} = |N_i^{in,k}|$.

We first introduce an assumption about the network connectivity that will be repeated utilized in later shown results in Assumption 4

Definition 1 (B-Strongly-Connected Matrix Sequence) *We say that a matrix sequence A^0, A^1, A^2,... is B-strongly-connected if the graph with the same vertex set and edge set as $\bigcup_{k=lB}^{(l+1)B-1} E_{A^k}$ is strongly connected for each $l = 0, 1, 2, ...$*

Assumption 4 (Assumption 1 from [3]) *The sequence of directed graphs G^0, G^1, G^2,... is B-strongly-connected and each node has a self-loop in every graph G^k.*

Assumption 4 assures that for a time-varying network, all nodes are sufficiently well-connected over periodic window of time. Given all nodes are well working, communication network that meets such assumption should not be very hard for implementation for reasonable length B.

3.1 Distributed Optimization for Time-Varying Undirected Graph

We then introduce a theorem about the convergence criteria and convergence time for the distributed subgradient method over a time-varying undirected graph in Theorem 5. Recall that the distributed subgradient method is expressed by Eq. 12.

$$x_i^{k+1} = \sum_{j \in N_i^k} A_{i,j}^k x_j^k - \alpha^k g_i^k, \tag{12}$$

where A^k is the adjacency matrix associated with the iteration rule and g_i^k is the subgradient of $f_i(\cdot)$ at x_i^k. For an undirected graph A, we have: $(i,j) \in E_A \Leftrightarrow (j,i) \in E_A$. This implies that a pair of linked nodes can exchange information back and forth.

Theorem 5 (Theorem 8 from [3]) *Let $\mathcal{X}^*$ denote the set of minimizers of the function f. We assume that: (i) each f_i is convex; (ii) $\mathcal{X}^*$ is nonempty; (iii) g_i is bounded by constant L at any point for any function f_i; (iv) matrices A^k are doubly stochastic and there exists some $\alpha > 0$ such that graph sequence $G_{[A^0]_\alpha}$, $G_{[A^1]_\alpha}$, $G_{[A^2]_\alpha}$,... satisfy Assumption 4; and (v) x_i^0 are the same for all nodes. Then we have:*

- *If α^k are square summable but not summable or namely:*
$$\sum_{k=0}^{\infty} \alpha^k = +\infty \quad \textit{and} \quad \sum_{k=0}^{\infty} [\alpha^k]^2 < \infty,$$
then for any $x^* \in \mathcal{X}^*$, we have that for all $i = 1, ..., n$, following results hold:
$$\lim_{t\to\infty} f\Big(\frac{\sum_{l=0}^{t} \alpha^l x_i^l}{\sum_{l=0}^{t} \alpha^l}\Big) = f(x^*),$$
$$\lim_{t\to\infty} x_i^t = x^*.$$

- *If we run for T steps with $\alpha^k = 1/\sqrt{T}$, then we have for all $i = 1, ..., n$,*
$$f\Big(\frac{\sum_{l=0}^{T-1} y^l}{T}\Big) - f(x^*) \le \frac{(y^0 - x^*)^2 + L^2}{2\sqrt{T}} + \frac{L^2}{\sqrt{T}(1-\lambda)},$$
$$\textit{with} \quad y^l = (1/n)\sum_{i=1}^{n} x_i^l \quad \textit{and} \quad \lambda = \sup_{l\ge 0} \sigma_2(A^l)$$
where $\sigma_2(A^l)$ denotes the second-largest singular value of the matrix A^l.

We first look at those five assumption in Theorem 5. Assumption (i-ii) are obvious. Assumption (iii) assures that the underlying function is Lipschitz continuous which is essential for the proof of convergence. Assumption (iv) assures that the updating rule will lead to consensus convergence of the system. Threshold matrices are required here to rule out the possibility that weights of A^k die out. One way to make A^k doubly stochastic is to apply lazy or regular Metropolis iteration. Here lazy Metropolis iteration (Eq 13) is preferred since it has more attractive convergence properties. Assumption (v) is not necessary for part (1) of Theorem 5, but leads to a smaller bound in part (2) of the theorem. When this assumption is violated, there is an extra term on the bound which decays exponentially on the order of λ^k. Thus, initialization of all nodes at same value is not critical for convergence but helpful for making each node approach the optimum faster.

$$x_i^{k+1} = x_i^k + \sum_{j\in N_i^k} \frac{1}{2\max\{d_i^k, d_j^k\}}(x_j^k - x_i^k). \tag{13}$$

Generally, Theorem 5 states that for distributed subgradient method, a choice of diminishing step size will lead to a convergence to the optimum, and a choice of constant step size will lead to a convergence to the optimum with an error bound depending on the number of iterations and spectral gap $1 - \lambda$ of the network. Notice that if constant step size is chosen, computation of $(\sum_{l=0}^{T-1} y^l)/T$ requires each node to keep track of $(\sum_{l=0}^{T-1} x_i^l)/T$.

We then show the convergence time of the distributed subgradient method. We measure the convergence rate by *ϵ-convergenece time* which is defined in Definition 2.

Definition 2 (ϵ-Convergenece Time [3]) *For a given desired accuracy level ϵ, the ϵ-convergence time is defined as the first time when*
$$f\Big(\frac{\sum_{l=0}^{T-1} y^l}{T}\Big) - f(x^*) \le \epsilon.$$

Intuitively, the convergence time depends on desired ϵ, network topology, number of nodes n, and the chosen updating rule. Suppose all the assumptions in Theorem 5 are met. For the lazy Metropolis based subgradient method, the convergence time can be upper bounded by Eq 14.
$$O\Big(\frac{\max((y^0 - x^*)^4, L^4 P_n^2)}{\epsilon^2}\Big), \tag{14}$$
where P_n depends on the spectral gap $1 - \lambda$ and varies from $O(1)$ to $O(n^2)$. It is possible to decrease the scaling from $O(L^4 P_n^2)$ to $O(L^2 P_n)$ if the step size is set to $\beta/\sqrt{T}$ where β depends on L, n and the graph family and needs to be separately optimized. For some applications, such requirement of prior knowledge of the network may not be accessible. Currently, further decreasing the convergence time without prior knowledge of the network remains an open question.

3.2 Distributed Optimization for Time-Varying Directed Graph

In this section, we focus on optimization over directed and time-varying graph. And the subgradient-push method will be used to optimize the system under the previous assumptions. The subgradient-push method only requires each node to know its out-degree at all times. The results in [2] show that each agent in graph converges at a rate of $O(\ln t/\sqrt{t})$, where the constant depends on the information diffusion speed of the corresponding directed graph sequence and the influence imbalance among the nodes. All the prior work in distributed optimization requires time-varying communications with some balance requirements, such as having a sequence of doubly stochastic matrices. A method proposed in [2] removes the need for doubly stochastic matrices.

The notations of $N_i^{in}(t)$ and $N_i^{out}(t)$ are a little bit different here which is given below:

$$N_i^{in}(t) = \{j|(j,i) \in E(t)\} \cup \{i\},$$

$$N_i^{out}(t) = \{j|(i,j) \in E(t)\} \cup \{i\},$$

and $d_i(t)$ is the out-degree of node i:

$$d_i(t) = |N_i^{out}(t)|.$$

In subgradient-push method, every node i maintains vector variables $x_i(t), w_i(t) \in R^d$, and a scalar variable $y_i(t)$. These variables will update in each iteration according to the following rules: for all $t \geq 0$ and all $i = 1, ..., n$:

$$w_i(t+1) = \sum_{j \in N_i^{in}(t)} \frac{x_j(t)}{d_j(t)},$$

$$y_i(t+1) = \sum_{j \in N_i^{in}(t)} \frac{y_j(t)}{d_j(t)},$$

$$z_i(t+1) = \frac{w_i(t+1)}{y_i(t+1)}),$$

$$x_i(t+1) = w_i(t+1) - \alpha(t+1) g_i(t+1), \tag{15}$$

where $g_i(t+1)$ is a subgradient of the function $f_i(z)$ at $z = z_i(t+1)$. It is initiated with an arbitrary vector $x_i(0) \in R^d$ at node i, and with $y_i(0) = 1$ for all i. The step size $\alpha(t+1) > 0$ satisfies the following decay condition:

$$\sum_{t=1}^{\infty} \alpha(t) = \infty \quad \text{and} \quad \sum_{t=1}^{\infty} \alpha^2(t) < \infty \quad \text{and} \quad \alpha(t) \leq \alpha(s) \text{ for all } t > s \geq 1.$$

Some broadcast-based feature can be found in the above updating rules: each node j broadcasts the value of $\frac{x_j}{d_j}$, $\frac{y_j}{d_j}$ at time t to all of its out neighbor nodes i, which simply sum all the messages they receive to obtain new w_i at time $t+1$. And obviously, $x_i(t+1)$ is computed without any further communication during step t. This protocol would be a version of the push-sum protocol if there are no subgradient term $g_i(t+1)$ in the representation of $x_i(t+1)$. With this subgradient term, $z_i(t+1)$ will converge to some common point rather than converging to each other like push-sum protocols does.

Theorem 6 (Theorem 1 from [2]) *Suppose that:*

- *The graph sequence $G(t)$ is uniformly strongly connected.*
- *Each function $f_i(z)$ is convex over R^d and the set $Z^* = Argmin_{z \in R^d} F(z)$ is nonempty.*
- *The subgradients of each $f_i(z)$ are uniformly bounded, i.e., there exists $L_i < \infty$ such that for all $z \in R^d$,*

$$\|g_i\| \leq L_i \textit{ for all subgradients } g_i \textit{ of } f_i(z).$$

Then, the distributed subgradient-push method of Eq.(3) with the stepsize satisfying the conditions in Eq.(4) has the following property

$$\lim_{t \to \infty} z_i(t) = z^* \textit{ for all i and for some } z^* \in Z^*$$

Theorem 6 shows the correctness of the subgradient-push method for an arbitrary stepsize $\alpha(t)$ satisfying the specified condition. This holds under the assumptions we have laid out above, as well as an additional technical assumption on the subgradient boundedness

Theorem 7 (Theorem 2 from [2]) *Suppose that all the assumptions of Theorem 1 hold, and let $\alpha(t) = \frac{1}{\sqrt{t}}$ for $t \geq 1$. Moreover, suppose that every node i maintains the variable $\tilde{z}_i(t) \in R^d$ initialized at time $t = 0$ with any $\tilde{z}_i(0) \in R^d$ and updated by*

$$\tilde{z}_i(t+1) = \frac{\alpha(t+1)z_i(t+1) + S(t)\tilde{z}_i(t)}{S(t+1)} \text{ for } t \geq 0$$

where $S(0) = 0$ and $S(t) = \sum_{s=0}^{t-1} \alpha(s+1)$ for $t \geq 1$. Then, we have for all $t \geq 1, i = 1, ..., n$, and any $z^* \in Z^*$,

$$F(\tilde{z}_i(t+1)) - F(z^*) \leq \frac{n}{2}\frac{\|\bar{x}(0) - z^*\|_1}{\sqrt{t+1}} + \frac{L^2(1 + \ln t + 1)}{2n\sqrt{t+1}} +$$

$$\frac{24L\sum_{j=1}^{n}\|x_j(0)\|_1}{\delta(1-\lambda)\sqrt{t+1}} + \frac{24dL^2(1+\ln t)}{\delta(1-\lambda)\sqrt{t+1}}$$

where $\bar{x}(0) = \frac{1}{n}\sum_{i=1}^{n} x_i(0)$.

Theorem 7 indicates the rate at which the objective function converges to its optimal value. In order to get a perfect rate result, they make two tweaks.

- Take a stepsize which decays as $\alpha(t) = \frac{1}{\sqrt{t}}$ (stepsizes which decay at faster rates uaually inferior convergence rates),
- Each node i will maintain a convex combination of the values $z_i(1)$, $z_i(2)$, ... for which the convergence rate will be obtained.

They demonstrate that the subgradient-push converges at a rage of $O(\frac{\ln t}{\sqrt{t}})$. As we can see, the convergence rate of the distributed methods depends on some measure of the connectivity of the directed graph sequence $G(1), G(2)....$ Here, λ and δ reflect the feature of graph sequence. Moreover, the closeness of λ to 1 measures the speed at which the (connectivity) graph sequence $G(t)$ diffuses the information among the nodes over time. And δ is a measure of the imbalance of influences among the nodes. Time-varying directed regular networks are uniform in the influence and will have $\delta = 1$; however, the graphs which have a row very close to zero, since some nodes are less prone to be influenced by others, the convergence time will increase dramatically. Consequently, δ may be regarded as a measure of the uniformity of long-term influence among the nodes. Theorem 7 also shows that for a class of time-varying regular directed graphs, $\frac{1}{\delta(1-\lambda)}$ scales polynomially in n.

3.3 Distributed Optimization with Agent Constraints

In many applications, agent can only take values x_i from a constraint set. In a more general setting, each agent can have a distinct closed convex constraint set $X_i \in \mathbb{R}^d$ which is only known to the agent itself. Under such condition, if the next value x_i^{k+1} generated by Eq. 12 is out of X_i, we can project it back without losing the convergence property by applying Euclidean projection shown in Eq. 16:

$$\Pi_{X_i}[\overline{x}] = \operatorname*{argmin}_{x \in X_i} \|\overline{x} - x\|, \tag{16}$$

where $\overline{x}$ is the projected vector. The constrained distributed subgradient method is shwon in Eq. 17:

$$x_i^{k+1} = \Pi_{X_i}\left[\sum_{j \in N_i^k} A_{i,j}^k x_j^k - \alpha^k g_i^k\right]. \tag{17}$$

Since the projection mapping function $\Pi_{X_i}[\cdot]$ is non-expansive, the convergence property remains the same as the one shown in Theorem 5.

3.4 Distributed Optimization with Noises

In reality, the subgradient g_i^k of the function f_i evaluated at x_i^k could be stochastic and the value x_j^k received from neighbors could be noisy due to the noises embedded in the communication channel. Here we consider the general case shown in Eq. 18 where both subgradients and links are noisy:

$$x_i^{k+1} = \sum_{j \in N_i^k} A_{i,j}^k (x_j^k + \xi_{i,j}^k) - \alpha^k \tilde{g}_i^k(\omega), \tag{18}$$

where $\tilde{g}_i^k(\omega)$ is a stochastic vector depending on the random variable ω and $\xi_{i,j}^k$ is a random link noise between nodes i and j. If we have: (1) All assumptions in Theorem 5 hold; (2) the stochastic subgradients satisfy: $\mathrm{E}[\tilde{g}_i^k(\omega)|x_i^k] = g_i^k$ where g_i^k is some gradient, and $\mathrm{E}[\|\tilde{g}_i^k(\omega) - g_i^k\|_2^2|x_i^k]$ is bounded for all k and i; (3) the link noises satisfy: $\mathrm{E}[\xi_{i,j}^k] = 0$ and $\mathrm{Var}(\xi_{i,j}^k)$ is bounded, then almost surely, $\lim_{k\to\infty} x_i^k = x^*$. for all i and for some $x^* \in X^*$.

In many modern electronic systems, noises from natural sources such as thermal noise and shot noise can be modeled as Gaussian noise which has zero mean and bounded variance. Therefore the assumption of noises from the reported papers is practical.

3.5 Asynchronous Distributed Optimization

In asynchronous distributed optimization, agents do not update at the same time and each agent may use different step size. Such case can be considered as distributed optimization over a random graph sequence G^k. Since the graph is random, conditions in Assumption 4 are imposed on the expected graph $\mathrm{E}[G^k]$ and corresponding adjacency matrix $\overline{A} = \mathrm{E}[A^k]$. We consider two algorithms: the random gossip algorithm where at each iteration k a random link is activated, and the random broadcast algorithm where at each iteration k a random node is activated for broadcasting information to its neighbors. Since A^k is stochastic, the subgradients g_i^k is stochastic as well. Therefore we can use the same concept from Section 3.4 to express the optimization method in Eq. 19:

$$x_i^{k+1} = \sum_{j \in N_i^k} A_{i,j}^k x_j^k - \alpha^k \tilde{g}_i^k(\omega). \tag{19}$$

For random gossip algorithm, the underlying graph is undirected and static, and at iteration k, a random link connecting agent i_k and j_k is activated. The corresponding matrix A^k can be expressed in Eq. 20:

$$A^k = I - \gamma(e_{i_k} - e_{j_k})(e_{i_k} - e_{j_k})^T, \tag{20}$$

where $\gamma \in (0,1)$, and e_i denotes the unit-norm vector with the i^{th} entry equal to 1. One can prove that A^k is doubly stochastic and thereby $\overline{A}$ is also doubly stochastic.

For random broadcast algorithm, at iteration k, a random node i_k is activated (each node can be activated with probability $1/n$). The corresponding matrix A^k can be expressed in Eq. 21:

$$\begin{aligned} A_{j,i_k}^k = \gamma, \quad A_{j,j}^k = 1-\gamma \quad \text{for all } j \in N_{i_k}, \\ A_{i,i}^k = 1 \quad \text{for all } i \neq i_k, \quad A_{i,j}^k = 0 \quad \text{otherwise}, \end{aligned} \tag{21}$$

where γ takes the same value in Eq. 20. Notice that for random broadcast algorithm, A^k is not doubly stochastic. However, study shows that the random broadcast based subgradient method can still converge to a minimizer of $f(\cdot)$ with probability of 1. [10]

4 Future Work

In Section 3.3, we present a distributed subgradient method with per-agent constraints. In real applications, the constraint of each agent X_i may be time-varying. For example, in robotic motion planing, each robot can only take positions/configurations from the configuration space which might be dependent on its surroundings. As each agent explores the environment, its configuration space can change over time. A possible future direction is to show how projections on the time-varying constraints will affect the convergence properties.

In Section 3.4, we present the effect of noisy subgradient and links. The analysis assumes the noises present in the system have zero mean and bounded variance and the link noises are uncorrelated. A practical future work is to investigate the effect of correlated link noises. For example, in many telecommunication systems, cross-talk could happen, which means that the link noise $\xi_{i,j}^k$ also depends on signals from other agents. In such case, assumptions of noises from the reported papers no longer hold. The correlated noises may cause additional "interactions" among agents through noises, which can affect the optimization process.

A fundamental limitation to the methods described in this report is that they consider only convex objective functions. In practice, many problems of interest may require the optimization of non-convex objectives. While the methods discussed will produce convergence to stationary points for any smooth functions, an interesting direction for future work would be to explore conditions for convergence to global optima in the presence of non-convexities.

5 Conclusion

The three papers ([1–3]) comprising the focus of this report present a general framework for distributed optimization over a multi-agent network, where each agent has a local objective function and exchanges information over the network to collaboratively optimize an overall objective function. In the presented case, the overall objective function is the sum of all local objective functions. The core algorithm discussed is the distributed subgradient method which combines the consensus algorithm and subgraident method. Under certain assumptions about the network connectivity and objective functions, by applying this algorithm, all agents can converge to some minimizer of the overall objective function, given appropriate setting of the iteration step sizes. The reported papers also presented some variations of distributed subgradient method and discussed how the network topology can affect their convergence rate. These variations are designed for systems with more realistic settings, including time-varying undirected and directed networks, optimization with agent constraints, optimization with noise, and asynchronous distributed optimization. These algorithms normally demand additional assumptions about the system, but are important advancements towards more practical algorithms. Finally, we comment on future directions, including the effect of time-varying per-agent constraints, effect of correlated link noises and optimization of non-convex objective functions.

References

[1] A. Nedic and A. Ozdaglar, “Distributed subgradient methods for multi-agent optimization,” *IEEE Transactions on Automatic Control*, vol. 54, no. 1, pp. 48–61, 2009.

[2] A. Nedić and A. Olshevsky, “Distributed optimization over time-varying directed graphs,” *IEEE Transactions on Automatic Control*, vol. 60, no. 3, pp. 601–615, 2014.

[3] A. Nedić, A. Olshevsky, and M. G. Rabbat, “Network topology and communication-computation tradeoffs in decentralized optimization,” *Proceedings of the IEEE*, vol. 106, no. 5, pp. 953–976, 2018.

[4] A. Nedic and A. Ozdaglar, “10 cooperative distributed multi-agent,” *Convex optimization in signal processing and communications*, vol. 340, 2010.

[5] J. N. Tsitsiklis, “Problems in decentralized decision making and computation.” Massachusetts Inst of Tech Cambridge Lab for Information and Decision Systems, Tech. Rep., 1984.

[6] V. D. Blondel, J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, “Convergence in multiagent coordination, consensus, and flocking,” in *Proceedings of the 44th IEEE Conference on Decision and Control*. IEEE, 2005, pp. 2996–3000.

[7] D. Bertsekas and A. Nedic, “Convex analysis and optimization (conservative),” 2003.

[8] A. Nedic and D. P. Bertsekas, “Incremental subgradient methods for nondifferentiable optimization,” *SIAM Journal on Optimization*, vol. 12, no. 1, pp. 109–138, 2001.

[9] A. Nedić and D. Bertsekas, “Convergence rate of incremental subgradient algorithms,” in *Stochastic optimization: algorithms and applications*. Springer, 2001, pp. 223–264.

[10] A. Nedić, “Random algorithms for convex minimization problems,” *Mathematical programming*, vol. 129, no. 2, pp. 225–253, 2011.