Privacy Preserving Randomized Gossip Algorithms

Filip Hanzely[⋆] Jakub Konečný[⋆] Nicolas Loizou[⋆] Peter Richtárik[◇⋆]
Dmitry Grishchenko[†]

[⋆] *University of Edinburgh, UK*
[◇] *KAUST, KSA*
[†] *Higher School of Economics, Russia*

June 21, 2017

Abstract

In this work we present three different randomized gossip algorithms for solving the average consensus problem while at the same time protecting the information about the initial private values stored at the nodes. We give iteration complexity bounds for all methods, and perform extensive numerical experiments.

1 Introduction

In this paper we consider the average consensus (AC) problem. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected connected network with node set $\mathcal{V} = \{1, 2, \ldots, n\}$ and edges $\mathcal{E}$ such that $|\mathcal{E}| = m$. Each node $i \in \mathcal{V}$ "knows" a private value $c_i \in \mathbb{R}$. The goal of AC is for every node of the network to compute the average of these values, $\bar{c} \stackrel{\text{def}}{=} \frac{1}{n}\sum_i c_i$, in a distributed fashion. That is, the exchange of information can only occur between connected nodes (neighbours).

The literature on methods for solving the average consensus problem is vast and has long history [52, 51, 4, 30]. The algorithms for solving this problem can be divided in two broad categories: the average consensus algorithms [54] and the gossip algorithms [5, 49]. The main difference is that the former work in a synchronous setting while the gossip algorithms model the case of asynchronous setting. In the average consensus algorithms, all nodes of the network update their values simultaneously by communicate with a set of their neighbours and in all iterations the same update occurs. In gossip algorithms, at each iteration, only one edge is selected randomly, and the corresponding nodes update their values to their average. In this work we focus on randomized gossip algorithms and propose techniques for protecting information of the initial values c_i, in the case when these may be sensitive.

In this work we develop and analyze three private variants of the randomized pairwise gossip algorithm for solving the average consensus problem. As an additional requirement we wish to prevent nodes to "learn" information about the private values of other nodes. While we shall not formalize the notion of privacy preservation in this work, it will be intuitively clear that our methods indeed make it harder for nodes to infer information about the private values of other nodes.

1.1 Background

The average consensus problem and randomized gossip algorithms for solving it appear in many applications, including distributed data fusion in sensor networks [55], load balancing [10] and clock synchronization [18]. This subject was first introduced in [52], and was studied extensively in the last decade; the seminal 2006 paper of Boyd et al. [5] on randomized gossip algorithms motivated a large amount of subsequent research and generated more than 1500 citations to date.

In this work, we focus on modifying the basic algorithm of [5], which we refer to as "Standard Gossip" algorithm. In the following, we review several avenues of research the gossip algorithms were evolved. While we do not address any privacy considerations in these settings, they can serve

as inspiration for further work. For a survey of relevant work prior to 2010, we refer the reader to reviews in [12, 40, 45].

The *Geographic Gossip algorithm* was proposed in [11], in which the authors combine the gossip approach with a geographic routing towards a randomly chosen location with main goal the improvement of the convergence rate of Standard Gossip algorithm. In each step, a node is activated, assuming that it is aware of its geographic location and some additional assumptions on the network topology, it chooses another node from the rest of the network (not necessarily one of its neighbours) and performs a pairwise averaging with this node. Later, using the same assumptions, this algorithm was extended into *Geographic Gossip Algorithm with Path Averaging* [3], in which connected sequences of nodes were chosen in each step and they averaged their values. More recently, in [19] and [20] authors propose a geographic and path averaging methods which converge to the average consensus without the assumption that nodes are aware of their geographic location.

Another important class of randomized gossip algorithms are the *Broadcast Gossip algorithms*, firts proposed in [2] and then extended in [17, 53, 29]. The idea of this algorithm is simple: In each step a node in the network is activated uniformly at random, following the asynchronous time model, and broadcasts its value to its neighbours. The neighbours receive this value and update their own values. It was experimentally shown that this method converge faster than the pairwise and geographic randomized gossip algorithms.

Alternative approach to the gossip framework are so called *non-randomized Gossip algorithms* [38, 27, 34, 56]. Typically, this class of algorithms executes the pairwise exchanges between nodes in a deterministic, such as pre-defined cyclic, order. T-periodic gossiping is a protocol which stipulates that each node must interact with each of its neighbours exactly once every T time units. Under suitable connectivity assumptions of the network $\mathcal{G}$, the T-periodic gossip sequence will converge at a rate determined by the magnitude of the second largest eigenvalue of the stochastic matrix determined by the sequence of pairwise exchanges which occurs over a period. It has been shown that if the underlying graph is a tree, this eigenvalue is the same for all possible T-periodic gossip protocols.

A different approach, uses memory in the update of the values each node holds, to get *Accelerated Gossip algorithms*. The nodes update their value using an update rule that involve not only the current values of the sampled nodes but also their previous values. This idea is closely related to the shift register methods studied in numerical linear algebra for improving the convergence rate of linear system solvers. The works [6, 33] have shown theoretically and numerically, that under specific assumptions this idea can improve the performance of the Standard Gossip algorithm.

Randomized Kaczmarz-type Gossip algorithms. Very recently has been proved that popular randomized Kaczmarz-type methods for solving large linear systems can also solve the AC problem. In particular, in [23] and [35] it was shown how that existing Randomized Kaczmarz and Randomized Block Kaczmarz methods can be interpreted as randomized gossip algorithms for solving the AC problem, by solving a particular system encoding the underlying network structure. This approach was the first time that a connection between the two research areas of linear system solvers and distributed algorithms have been established.

In this work we are interested in the asynchronous time model [5, 4]. More precisely, we assume that each node of our network has a clock which ticks at a rate of 1 Poisson process. This is equivalent of having available a global clock which ticks according to a rate n Poisson process and selects an edge of the network uniformly at random. In general the synchronous setting (all nodes update the values of their nodes simultaneously using information from a set of their neighbours) is convenient for theoretical considerations, but is not representative of some practical scenarios, such as the distributed nature of sensor networks. For more details on clock modelling we refer the reader to [5], as the contribution of this paper is orthogonal to these considerations.

Privacy and Average Consensus. Finally, the introduction of notions of privacy within the AC problem is relatively recent in the literature, and the existing works consider two different ideas.

The concept of differential privacy [13] is used to protect the output value $\bar{c}$ computed by all nodes in [28]. In this work, an exponentially decaying Laplacian noise is added to the consensus computation. This notion of privacy refers to protection of the final average, and formal guarantees are provided.

A different goal is the protection of the initial values c_i the nodes know at the start. In [36, 37], the

goal is to make sure that each node is unable to infer a lot about the initial values c_i of any other node. Both of these methods add noise correlated across individual iterations, to make sure they converge to the exact average. A formal notion of privacy breach is formulated in [37], in which they also show that their algorithm is optimal in that particular sense.

1.2 Main Contributions

In this work we present three different approaches for solving the Average Consensus problem while at the same time protecting the information about the initial values. All of the above mentioned works focus on the synchronous setting of the AC problem. This work is the first which combines the *gossip framework* with the privacy concept of protection of the initial values. It is important to stress that we provide tools for protection of the initial values, but we do not address any specific notion of privacy or a threat model, nor how these quantitatively translate to any explicit measure. These would be highly application dependent, and we only provide theoretical convergence rates for the techniques we propose.

The methods we propose are all dual in nature. The dual approach is explained in detail in Section 2. It was first proposed for solving linear systems in [23] and then extend to the concept of average consensus problems in [35]. The dual updates immediately correspond to updates to the primal variables, via an affine mapping. One of the contributions of our work is that we exactly recover existing convergence rates for the primal iterates as a special case.

We now outline the three different techniques we propose in this paper, which we refer to as "Binary Oracle", "ϵ-Gap Oracle" and "Controlled Noise Insertion". The first two are, to the best of our knowledge, the first proposal of weakening the oracle used in the gossip framework. The latter is inspired by and similar to the addition of noise proposed in [37] for the synchronous setting. We extend this technique by providing explicit finite time convergence guarantees.

Binary Oracle. The difference from standard Gossip algorithms we propose is to reduce the amount of information transmitted in each iteration to a single bit. More precisely, when an edge is selected, each corresponding node will only receive information whether the value on the other node is smaller or larger. Instead of setting the value on each node to their average, each node increases or decreases its value by a pre-specified step.

ϵ-Gap Oracle. In this case, we have an oracle that returns one of three options, and is parametrized by ϵ. If the difference in values of sampled nodes is larger than ϵ, an update similar to the one in Binary Oracle is taken. Otherwise, the values remain unchanged. An advantage compared to the Binary Oracle is that this approach will converge to a certain accuracy and stop there, determined by ϵ (Binary Oracle will oscillate around optimum for a fixed stepsize). However, in general it will disclose more information about the initial values.

Controlled Noise Insertion. This approach is inspired by the works of [36, 37], and protects the initial values by inserting noise in the process. Broadly speaking, in each iteration, each of the sampled nodes first add a noise to its current value, and an average is computed afterwards. Convergence is guaranteed because of correlation in the noise across iterations. Each node remembers the noise it added last time it was sampled, and in the following iteration, the previously added noise is first subtracted, and a fresh noise of smaller magnitude is added.

Empirically, the protection of initial values is provided by first injecting noise in the system, which propagates across network, but is gradually withdrawn to ensure convergence to the true average.

Convergence Rates of our Methods. In Table 1, we present summary of convergence guarantees for the above three techniques. By $\|\cdot\|$ we denote the standard Euclidean norm.

The two approaches to restricting the amount of information disclosed, Binary Oracle and ϵ-Gap Oracle, converge slower than the standard Gossip. In particular, these algorithms have sublinear convergence rate. At first sight, this should not be surprising, since we indeed use much less information. However, in Theorem 7, we show that if we had in a certain sense perfect global information, we could use it to construct a sequence of adaptive stepsizes, which would push the capability of the binary oracle to a linear convergence rate. However, this rate is still m-times slower than the standard rate of the binary gossip algorithm. We note, however, that having global information at hand is an

Main Results			
Randomized Gossip Methods	Convergence Rate	Success Measure	Thm
Standard Gossip [5]	$\left(1-\frac{\alpha(\mathcal{G})}{2m}\right)^k$	$\mathbb{E}\left[\frac{1}{2}\|\bar{c}\mathbf{1}-x^k\|^2\right]$	5
New: Private Gossip with Binary Oracle	$1/\sqrt{k}$	$\min_{t\leq k}\mathbb{E}\left[\frac{1}{m}\sum_e \lvert x_i^t - x_j^t\rvert\right]$	6
New: Private Gossip with ϵ-Gap Oracle	$1/(k\epsilon^2)$	$\mathbb{E}\left[\frac{1}{k}\sum_{t=0}^{k-1}\Delta^t(\epsilon)\right]$	9
New: Private Gossip with Controlled Noise Insertion	$\left(1-\min\left(\frac{\alpha(\mathcal{G})}{2m},\frac{\gamma}{m}\right)\right)^k$	$\mathbb{E}\left[D(y^*)-D(y^k)\right]$	11

Table 1: Complexity results of all proposed gossip algorithms.

impractical assumption. Nevertheless, this result highlights that there is a potentially large scope for improvement, which we leave for future work.

These two oracles could be in practice implemented using established secure multiparty computation protocols [8]. However, this would require the sampled nodes to exchange more than a single message in each iteration. This is inferior to the requirements of the standard gossip algorithm, but the concern of communication efficiency is orthogonal to the contribution of this work, and we do not address it further.

The approach of Controlled Noise Insertion yields a linear convergence rate which is driven by maximum of two factors. Without going into details, which of these is bigger depends on the speed by which the magnitude of the inserted noise decays. If the noise decays fast enough, we recover the convergence rate of standard the gossip algorithm. In the case of slow decay, the convergence is driven by this decay. By $\alpha(\mathcal{G})$ we denote the *algebraic connectivity* of graph $\mathcal{G}$ [16]. The parameter γ controls the decay speed of the inserted noise, see (20).

1.3 Measures of Success

Second major distinction to highlight is that convergence of each of these proposals naturally depend on a different measure of suboptimality. All of them go to 0 as we approach optimal solution. The details of these measures will be described later in the main body, but we give a brief summary of their connections below.

The standard Gossip and Controlled Noise Insertion depend on the same quantity, but we present the latter in terms of dual values as this is what our proofs are based on. Lemma 1 formally specifies this equivalence. The binary oracle depends on average difference among directly connected nodes. The measure for the ϵ-Gap Oracle depends on quantities $\Delta^t(\epsilon)=\frac{1}{m}\left|\{(i,j)\in\mathcal{E}:|x_i^t-x_j^t|\geq\epsilon\}\right|$, which is the number of edges, connecting nodes, values of which differ by more than ϵ.

To draw connection between these measures, we provide the following lemma, proved in the Appendix. The dual function D is formally defined in Section 2.

Lemma 1. *(Relationship between convergence measures) Suppose that x is primal variable corresponding to the dual variable y as defined in (9). Dual suboptimality can be expressed as the following [23]:*

$$D(y^*)-D(y)=\tfrac{1}{2}\|\bar{c}\mathbf{1}-x\|^2. \tag{1}$$

Moreover, for any $x\in\mathbb{R}^n$ we have :

$$\frac{1}{2n}\sum_{i=1}^{n}\sum_{j=1}^{n}(x_j - x_i)^2 = \|\bar{c}\mathbf{1} - x\|^2 \tag{2}$$

$$\sum_{e=(i,j)\in\mathcal{E}} |x_i - x_j| \leq \sqrt{mn}\|\bar{c}\mathbf{1} - x\|, \tag{3}$$

$$\sum_{e=(i,j)\in\mathcal{E}} |x_i - x_j| \geq \sqrt{\alpha(\mathcal{G})}\|\bar{c}\mathbf{1} - x\|, \tag{4}$$

$$\sum_{e=(i,j)\in\mathcal{E}} |x_i - x_j| \geq \epsilon\left|\{(i,j)\in\mathcal{E} : |x_i - x_j| \geq \epsilon\}\right|. \tag{5}$$

1.4 Outline

The remainder of this paper is organized as follows: Section 2 introduces the basic setup that are used through the paper. A detailed explanation of the duality behind the randomized pairwise gossip algorithm is given. We also include a novel and insightful dual analysis of this method as it will make it easier to the reader to parse later development. In Section 3 we present our three private gossip algorithms as well as the associated iteration complexity results. Section 4 is devoted to the numerical evaluation of our methods. Finally, conclusions are drawn in Section 5. All proofs not included in the main text can be found in the Appendix.

2 Dual Analysis of Randomized Pairwise Gossip

As we outlined in the introduction, our approach to extending the (standard) randomized pairwise gossip algorithm to privacy preserving variants utilizes duality. The purpose of this section is to formalize this duality, following the development in [23]. In addition, we provide a novel and self-contain dual analysis of randomized pairwise gossip. While this is of an independent interest, we include the proofs as their understanding aids in the understanding of the more involved proofs of our private gossip algorithms developed in the remainder of the paper.

2.1 Primal and Dual Problems

Consider solving the (primal) problem of projecting a given vector $c = x^0 \in \mathbb{R}^n$ onto the solution space of a linear system:

$$\min_{x\in\mathbb{R}^n}\{P(x) \stackrel{\text{def}}{=} \tfrac{1}{2}\|x - x^0\|^2\} \quad \text{subject to} \quad \mathbf{A}x = b, \tag{6}$$

where $\mathbf{A} \in \mathbb{R}^{m\times n}$, $b \in \mathbb{R}^m$, $x^0 \in \mathbb{R}^n$. We assume the problem is feasible, i.e., that the system $\mathbf{A}x = b$ is consistent. With the above optimization problem we associate the dual problem

$$\max_{y\in\mathbb{R}^m} D(y) \stackrel{\text{def}}{=} (b - \mathbf{A}x^0)^\top y - \tfrac{1}{2}\|\mathbf{A}^\top y\|^2. \tag{7}$$

The dual is an unconstrained concave (but not necessarily strongly concave) quadratic maximization problem. It can be seen that as soon as the system $\mathbf{A}x = b$ is feasible, the dual problem is bounded. Moreover, all bounded concave quadratics in $\mathbb{R}^m$ can be written in the as $D(y)$ for some matrix $\mathbf{A}$ and vectors b and x^0 (up to an additive constant).

With any dual vector y we associate the primal vector via an affine transformation:

$$x(y) = x^0 + \mathbf{A}^\top y.$$

It can be shown that if y^* is dual optimal, then $x^* = x(y^*)$ is primal optimal. Hence, any dual algorithm producing a sequence of dual variables $y^t \to y^*$ gives rise to a corresponding primal algorithm producing the sequence $x^t \stackrel{\text{def}}{=} x(y^t) \to x^*$. We shall now consider one such dual algorithm.

2.2 Stochastic Dual Subspace Ascent

SDSA [23] is a stochastic method for solving the dual problem (7). If we assume that $b = 0$, the iterates of SDSA take the form

$$y^{t+1} = y^t - \mathbf{S}_t(\mathbf{S}_t^\top \mathbf{A}\mathbf{A}^\top \mathbf{S}_t)^\dagger \mathbf{S}_t^\top \mathbf{A}(x^0 + \mathbf{A}^\top y^t), \tag{8}$$

where $\mathbf{S}_t$ is a random matrix drawn from independently at each iteration t from an arbitrary but fixed distribution $\mathcal{D}$, and $\dagger$ denotes the Moore-Penrose pseudoinverse.

The corresponding primal iterates are defined via:

$$x^t \overset{\text{def}}{=} x(y^t) = x^0 + \mathbf{A}^\top y^t \tag{9}$$

The relevance of this all to average consensus follows through the observation, as we shall see next, that for a specific choice of matrix $\mathbf{A}$ (as defined in the next subsection) and distribution $\mathcal{D}$, method (9) is equivalent to the (standard) randomized pairwise gossip method. In that case, SDSA is a dual variant of randomized pairwise gossip. In particular, we define $\mathcal{D}$ as follows: S_t is a unit basis vector in $\mathbb{R}^m$, chosen uniformly at random from the collection of all such unit basis vectors, denoted $\{f_e \mid e \in \mathcal{E}\}$. In this case, SDSA is a randomized coordinate ascent method applied to the dual problem.

For general distributions $\mathcal{D}$, the primal methods obtained from SDSA via (9) (but without observing that it arises that way) was fist proposed and studied in [22] under a full rank assumption on $\mathbf{A}$. This assumption was lifted, and duality exposed and studied as we explain it here, in [23]. For deeper insights and connections to stochastic optimization, stochastic fixed point methods, stochastic linear systems and probabilistic intersection problems, we refer the reader to [48]. The method can be extended to compute the inverse [25] and pseudoinverse [24] of a matrix, in which case it has deep connections with quasi-Newton updates [25]. In particular, it can be used to design a stochastic block extension of the famous BFGS method [25] and applied to the empirical risk minimization problem arising in machine learning to design a fast stochastic quasi-Newton training method [21].

2.3 Randomized Gossip Setup: Choosing A

We wish $(\mathbf{A}, b)$ to be an *average consensus (AC)* system, defined next.

Definition 2. *([35]) Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be an undirected graph with $|\mathcal{V}| = n$ and $|\mathcal{E}| = m$. Let $\mathbf{A}$ be a real matrix with n columns. The linear system $\mathbf{A}x = b$ is an "average consensus (AC) system" for graph $\mathcal{G}$ if $\mathbf{A}x = b$ iff $x_i = x_j$ for all $(i, j) \in \mathcal{E}$.*

Note that if $\mathbf{A}x = b$ is an AC system, then the solution of the primal problem (6) is necessarily $x^* = \bar{c} \cdot \mathbf{1}$, where $\bar{c} = \frac{1}{n}\sum_{i=1}^n x_i^0$. This is exactly what we want: we want the solution of the primal problem to be $x_i^* = \bar{c}$ for all i: the average of the private values stored at the nodes.

It is easy to see that a linear system is an AC system precisely when $b = 0$ and the nullspace of $\mathbf{A}$ is $\{t\mathbf{1} : t \in \mathbb{R}\}$, where $\mathbf{1}$ is the vector of all ones in $\mathbb{R}^n$. Hence, $\mathbf{A}$ has rank $n - 1$.

In the rest of this paper we focus on a specific AC system; one in which the matrix $\mathbf{A}$ is the incidence matrix of the graph $\mathcal{G}$ (see Model 1 in [23]). In particular, we let $\mathbf{A} \in \mathbb{R}^{m \times n}$ be the matrix defined as follows. Row $e = (i, j) \in \mathcal{E}$ of $\mathbf{A}$ is given by $\mathbf{A}_{ei} = 1$, $\mathbf{A}_{ej} = -1$ and $\mathbf{A}_{el} = 0$ if $l \notin \{i, j\}$. Notice that the system $\mathbf{A}x = 0$ encodes the constraints $x_i = x_j$ for all $(i, j) \in \mathcal{E}$, as desired.

2.4 Randomized Pairwise Gossip

We provide both primal and dual form of the (standard) randomized pairwise gossip algorithm.

The primal form is standard and needs no lengthy commentary. At the beginning of the process, node i contains private information $c_i = x_i^0$. In each iteration we sample a pair of connected nodes $(i, j) \in \mathcal{E}$ uniformly at random, and update x_i and x_j to their average. We let the values at the remaining nodes intact.

Algorithm 1 (Primal form)

Input: vector of private values $c \in \mathbb{R}^n$

Initialize: Set $x^0 = c$.

for $t = 0, 1, \ldots, k-1$ **do**

1. Choose node $e = (i, j) \in \mathcal{E}$ uniformly at random
2. Update the primal variable:

$$x_l^{t+1} = \begin{cases} \frac{x_i^t + x_j^t}{2}, & l \in \{i, j\} \\ x_l^t, & l \notin \{i, j\}. \end{cases}$$

end

return x^k

The dual form of the standard randomized pairwise gossip method is a specific instance of SDSA, as described in (8), with $x^0 = c$ and $\mathbf{S}_t$ being a randomly chosen standard unit basis vector f_e in $\mathbb{R}^m$ (e is a randomly selected edge). It can be seen [23] that in that case, (8) takes the following form:

Algorithm 1 (Dual form)

Input: vector of private values $c \in \mathbb{R}^n$

Initialize: Set $y^0 = 0 \in \mathbb{R}^m$.

for $t = 0, 1, \ldots, k-1$ **do**

1. Choose node $e = (i, j) \in \mathcal{E}$ uniformly at random
2. Update the dual variable:

$$y^{t+1} = y^t + \lambda^t f_e \quad \text{where} \quad \lambda^t = \text{argmax}_{\lambda'} D(y^t + \lambda' f_e).$$

end

return y^k

The following lemma is useful for the analysis of all our methods.It describes the increase in the dual function value after an arbitrary change to a single dual variable e.

Lemma 3. *Define $z = y^t + \lambda f_e$, where $e = (i, j)$ and $\lambda \in \mathbb{R}$. Then*

$$D(z) - D(y^t) = -\lambda(x_i^t - x_j^t) - \lambda^2. \tag{10}$$

Proof. The claim follows by direct calculation:

$$\begin{aligned} D(y^t + \lambda f_e) - D(y^t) &= -(\mathbf{A}c)^\top (y^t + \lambda f_e) - \tfrac{1}{2}\|\mathbf{A}^\top (y^t + \lambda f_e)\|^2 + (\mathbf{A}c)^\top y^t + \tfrac{1}{2}\|\mathbf{A}^\top y^t\|^2 \\ &= -\lambda f_e^\top \mathbf{A} \underbrace{(c + \mathbf{A}^\top y^t)}_{x^t} - \tfrac{1}{2}\lambda^2 \underbrace{\|\mathbf{A}^\top f_e\|^2}_{=2} \quad = \quad -\lambda(x_i^t - x_j^t) - \lambda^2. \end{aligned}$$

□

The maximizer in λ of the expression in (10) leads to the exact line search formula $\lambda^t = (x_j^t - x_i^t)/2$ used in the dual form of the method.

2.5 Complexity Results

With graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ we now associate a certain quantity, which we shall denote $\beta = \beta(\mathcal{G})$. It is the smallest nonnegative number β such that the following inequality[1] holds for all $x \in \mathbb{R}^n$:

[1] We write $\sum_{(i,j)}$ to indicate sum over all *unordered* pairs of vertices. That is, we do not count (i, j) and (j, i) separately, only once. By $\sum_{(i,j)\in\mathcal{E}}$ we denote a sum over all edges of $\mathcal{G}$. On the other hand, by writing $\sum_i \sum_j$, we are summing over all (unordered) pairs of vertices twice.

$$\sum_{(i,j)}(x_j - x_i)^2 \leq \beta \sum_{(i,j)\in\mathcal{E}} (x_j - x_i)^2. \tag{11}$$

The Laplacian matrix of graph $\mathcal{G}$ is given by $\mathbf{L} = \mathbf{A}^\top\mathbf{A}$. Let $\lambda_1(\mathbf{L}) \geq \lambda_2(\mathbf{L}) \geq \cdots \geq \lambda_{n-1}(\mathbf{L}) \geq \lambda_n(\mathbf{L})$ be the eigenvalues of $\mathbf{L}$. The *algebraic connectivity* of $\mathcal{G}$ is the second smallest eigenvalue of $\mathbf{L}$:

$$\alpha(\mathcal{G}) = \lambda_{n-1}(\mathbf{L}). \tag{12}$$

We have $\lambda_n(\mathbf{L}) = 0$. Since we assume $\mathcal{G}$ to be connected, we have $\alpha(\mathcal{G}) > 0$. Thus, $\alpha(\mathcal{G})$ is the smallest nonzero eigenvalue of the Laplacian: $\alpha(\mathcal{G}) = \lambda^+_{\min}(\mathbf{L}) = \lambda^+_{\min}(\mathbf{A}^\top\mathbf{A})$. As the next result states, the quantities $\beta(\mathcal{G})$ and $\alpha(\mathcal{G})$ are inversely proportional.

Lemma 4. $\beta(\mathcal{G}) = \frac{n}{\alpha(\mathcal{G})}$.

The following theorem gives a complexity result for (standard) randomized gossip. Our analysis is dual in nature (see the Appendix).

Theorem 5. *Consider the randomized gossip algorithm (Algorithm 1) with uniform edge-selection probabilities: $p_e = 1/m$. Then:*

$$\mathbb{E}\left[D(y^*) - D(y^k)\right] \leq \left(1 - \frac{\alpha(\mathcal{G})}{2m}\right)^k [D(y^*) - D(y^0)].$$

Theorem 5 yields the complexity estimate $\mathcal{O}\left(\frac{2m}{\alpha(\mathcal{G})}\log(1/\epsilon)\right)$, which exactly matches the complexity result obtained from the primal analysis [23]. Hence, the primal and dual analyses give the same rate.

Randomized coordinate descent methods were first analyzed in [32, 39, 46, 47]. For a recent treatment, see [42, 43]. Duality in randomized coordinate descent methods was studied in [50, 44]. Acceleration was studied in [31, 15, 1]. These methods extend to nonsmooth problems of various flavours [14, 7].

With all of this preparation, we are now ready to formulate and analyze our private gossip algorithms; we do so in Section 3.

3 Private Gossip Algorithms

In this section we introduce three novel private gossip algorithms, complete with iteration complexity guarantees. In Section 3.1 we protect privacy via a binary communication protocol. In Section 3.2 we communicate more: besides binary information, we allow for the communication of a bound on the gap, introducing the ϵ-gap oracle. In Section 3.3 we introduce a privacy-protection mechanism based on a procedure we call *controlled noise insertion*.

3.1 Private Gossip via Binary Oracle

We now present the gossip algorithm with Binary Oracle in detail and provide theoretical convergence guarantee. The information exchanged between sampled nodes is constrained to a single bit, describing which of the nodes has the higher value. As mentioned earlier, we only present the conceptual idea, not how exactly would the oracle be implemented within a secure multiparty protocol between participating nodes [8].

We will first introduce dual version of the algorithm.

Algorithm 2 (Dual form)

Input: vector of private values $c \in \mathbb{R}^n$, sequence of positive stepsizes $\{\lambda^t\}_{t=0}^{\infty}$
Initialize: Set $y^0 = 0 \in \mathbb{R}^m$, $x^0 = c$
for $t = 0, 1, \ldots, k-1$ **do**

1. Choose node $e = (i, j) \in \mathcal{E}$ uniformly at random
2. Update the dual variable:
$$y^{t+1} = \begin{cases} y^t + \lambda^t f_e, & x_i^t < x_j^t, \\ y^t - \lambda^t f_e, & x_i^t \geq x_j^t. \end{cases}$$
3. Set
$$\begin{aligned} x_i^{t+1} &= \begin{cases} x_i^t + \lambda^t, & x_i^t < x_j^t, \\ x_i^t - \lambda^t, & x_i^t \geq x_j^t. \end{cases} \\ x_j^{t+1} &= \begin{cases} x_j^t - \lambda^t, & x_i^t < x_j^t, \\ x_j^t + \lambda^t & x_i^t \geq x_j^t. \end{cases} \\ x_l^{t+1} &= x_l^t \qquad l \notin \{i, j\} \end{aligned}$$

end
return y^k

The update of primal variables above is equivalent to set x^{t+1} as primal point corresponding to dual iterate: $x^{t+1} = c + \mathbf{A}^\top y^{t+1} = x^t + \mathbf{A}^\top (y^{t+1} - y^t)$. In other words, the primal iterates $\{x^t\}$ associated with the dual iterates $\{y^t\}$ can be written in the form:

$$x^{t+1} = \begin{cases} x^t + \lambda^t \mathbf{A}_{e:}^\top, & x_i^t < x_j^t, \\ x^t - \lambda^t \mathbf{A}_{e:}^\top, & x_i^t \geq x_j^t. \end{cases}$$

It is easy to verify that due to the structure of $\mathbf{A}$, this is equivalent to the updates above.

Since the evolution of dual variables $\{y^k\}$ serves only the purpose of the analysis, the method can be written in the primal-only form as follows:

Algorithm 2 (Primal form)

Input: vector of private values $c \in \mathbb{R}^n$, sequence of positive stepsizes $\{\lambda^t\}_{t=0}^{\infty}$
Initialize: Set $x^0 = c$
for $t = 0, 1, \ldots, k-1$ **do**

1. Choose node $e = (i, j) \in \mathcal{E}$ uniformly at random
2. Set
$$\begin{aligned} x_i^{t+1} &= \begin{cases} x_i^t + \lambda^t, & x_i^t < x_j^t, \\ x_i^t - \lambda^t, & x_i^t \geq x_j^t. \end{cases} \\ x_j^{t+1} &= \begin{cases} x_j^t - \lambda^t, & x_i^t < x_j^t, \\ x_j^t + \lambda^t & x_i^t \geq x_j^t. \end{cases} \\ x_l^{t+1} &= x_l^t \qquad l \notin \{i, j\} \end{aligned}$$

end
return x^k

Given a sequence of stepsizes $\{\lambda^t\}$, it will be convenient to define $\alpha^k \overset{\text{def}}{=} \sum_{t=0}^{k} \lambda^t$ and $\beta^k \overset{\text{def}}{=}$

$\sum_{t=0}^{k} \left(\lambda^t\right)^2$. In the following theorem, we study the convergence of the quantity

$$L^t \stackrel{\text{def}}{=} \frac{1}{m} \sum_{e=(i,j)\in\mathcal{E}} |x_i^t - x_j^t|. \tag{13}$$

Theorem 6. *For all $k \geq 1$ we have*

$$\min_{t=0,1,\ldots,k} \mathbb{E}\left[L^t\right] \leq \sum_{t=0}^{k} \frac{\lambda^t}{\alpha^k} \mathbb{E}\left[L^t\right] \leq U^k \stackrel{def}{=} \frac{D(y^*) - D(y^0)}{\alpha^k} + \frac{\beta^k}{\alpha^k}. \tag{14}$$

Moreover:

(i) If we set $\lambda^t = \lambda^0 > 0$ for all t, then $U^k = \frac{D(y^*)-D(y^0)}{\lambda^0(k+1)} + \lambda^0$.

(ii) Let R be any constant such that $R \geq D(y^*) - D(y^0)$. If we fix $k \geq 1$, then the choice of stepsizes $\{\lambda^0, \ldots, \lambda^k\}$ which minimizes U^k correspond to the constant stepsize rule $\lambda^t = \sqrt{\frac{R}{k+1}}$ for all $t = 0, 1, \ldots, k$, and $U^k = 2\sqrt{\frac{R}{k+1}}$.

(iii) If we set $^t = a/\sqrt{t+1}$ for all $t = 0, 1, \ldots, k$, then

$$U^k \leq \frac{D(y^*) - D(y^0) + a^2\left(\log(k+3/2) + \log(2)\right)}{2a\left(\sqrt{k+2}-1\right)} = O\left(\frac{\log(k)}{\sqrt{k}}\right)$$

The part *(ii)* of Theorem 6 is useful in the case if we know exactly the number of iterations before running the algorithm, providing in a sense optimal stepsizes and rate $O(1/\sqrt{k})$. However, this might not be the case in practice. Therefore part *(iii)* is also relevant, which yields the rate $O(\log(k)/\sqrt{k})$. These bounds are significantly weaker than the standard bound in Theorem 5. This should not be surprising though, as we use significantly less information than the Standard Gossip algorithm.

Nevertheless, there is a potential gap in terms of what rate can be practically achievable. The following theorem can be seen as a form of a bound on what convergence rate is possible to attain with the Binary Oracle. However, this is attained with access to very strong information needed to set the sequence of stepsizes λ^t, likely unrealistic in any application. This result points at a gap in the analysis which we leave open. We do not know whether the sublinear convergence rate in Theorem 6 is necessary or improvable without additional information about the system.

Theorem 7. *For Algorithm 2 with stepsizes chosen in iteration t adaptively to the current values of x^t as $\lambda^t = \frac{1}{2m}\sum_{e\in\mathcal{E}} |x_i^t - x_j^t|$, we have*

$$\mathbb{E}\left[\|\bar{c}\mathbf{1} - x^k\|^2\right] \leq \left(1 - \frac{\alpha(\mathcal{G})}{2m^2}\right)^k \|\bar{c}\mathbf{1} - x^0\|^2$$

Comparing Theorem 7 with the result for standard Gossip in Theorem 5, the convergence rate is worse by factor of m, which is the price we pay for the weaker oracle.

An alternative to choosing adaptive stepsizes is the use of adaptive probabilities [9]. We leave such a study to future work.

3.2 Private Gossip via ϵ-Gap Oracle

Here we present the gossip algorithm with ϵ-Gap Oracle in detail and provide theoretical convergence guarantees. The information exchanged between sampled nodes is restricted to be one of three cases, based on difference in values on sampled nodes. As mentioned earlier, we only present the conceptual idea, not how exactly would the oracle be implemented within a secure multiparty protocol between participating nodes [8].

We will first introduce dual version of the algorithm.

Algorithm 3 (Dual form)

Input: vector of private values $c \in \mathbb{R}^n$; error tolerance $\epsilon > 0$
Initialize: Set $y^0 = 0 \in \mathbb{R}^m$; $x^0 = c$.
for $t = 0, 1, \ldots, k-1$ **do**

1. Choose node $e = (i, j) \in \mathcal{E}$ uniformly at random
2. Update the dual variable:

$$y^{t+1} = \begin{cases} y^t + \frac{\epsilon}{2} f_e, & x_i^t - x_j^t < -\epsilon \\ y^t - \frac{\epsilon}{2} f_e, & x_j^t - x_i^t < -\epsilon, \\ y^t, & \text{otherwise.} \end{cases}$$

3. If $x_i^t \leq x_j^t - \epsilon$ then $x_i^{t+1} = x_i^t + \frac{\epsilon}{2}$ and $x_j^{t+1} = x_j^t - \frac{\epsilon}{2}$
4. If $x_j^t \leq x_i^t - \epsilon$ then $x_i^{t+1} = x_i^t - \frac{\epsilon}{2}$ and $x_j^{t+1} = x_j^t + \frac{\epsilon}{2}$

end
return x^k

Note that the primal iterates $\{x^t\}$ associated with the dual iterates $\{y^t\}$ can be written in the form:

$$x^{t+1} = \begin{cases} x^t + \frac{\epsilon}{2} \mathbf{A}_{e:}^\top, & x_i^t - x_j^t < -\epsilon \\ x^t - \frac{\epsilon}{2} \mathbf{A}_{e:}^\top, & x_j^t - x_i^t < -\epsilon, \\ x^t, & \text{otherwise.} \end{cases}$$

The above is equivalent to setting $x^{t+1} = x^t + \mathbf{A}^\top (y^{t+1} - y^t) = c + \mathbf{A}^\top y^{t+1}$.

Since the evolution of dual variables $\{y^t\}$ serves only the purpose of the analysis, the method can be written in the primal-only form as follows:

Algorithm 3 (Primal form)

Input: vector of private values $c \in \mathbb{R}^n$; error tolerance $\epsilon > 0$
Initialize: Set $x^0 = c$.
for $t = 0, 1, \ldots, k-1$ **do**

1. Set $x^{t+1} = x^t$
2. Choose node $e = (i, j) \in \mathcal{E}$ uniformly at random
3. If $x_i^t \leq x_j^t - \epsilon$ then $x_i^{t+1} = x_i^t + \frac{\epsilon}{2}$ and $x_j^{t+1} = x_j^t - \frac{\epsilon}{2}$
4. If $x_j^t \leq x_i^t - \epsilon$ then $x_i^{t+1} = x_i^t - \frac{\epsilon}{2}$ and $x_j^{t+1} = x_j^t + \frac{\epsilon}{2}$

end
return x^k

Before stating the convergence result, let us define a quantity the convergence will naturally depend on. For each edge $e = (i, j) \in \mathcal{E}$ and iteration $t \geq 0$ define the random variable

$$\Delta_e^t(\epsilon) \quad \stackrel{\text{def}}{=} \quad \begin{cases} 1, & |x_i^t - x_j^t| \geq \epsilon, \\ 0, & \text{otherwise.} \end{cases}$$

Moreover, let

$$\Delta^t(\epsilon) \quad \stackrel{\text{def}}{=} \quad \frac{1}{m} \sum_{e \in \mathcal{E}} \Delta_e^t(\epsilon). \tag{15}$$

The following Lemma bounds the expected increase in dual function value in each iteration.

Lemma 8. *For all $t \geq 0$ we have $\mathbb{E}\left[D(y^{t+1}) - D(y^t)\right] \geq \frac{\epsilon^2}{4}\mathbb{E}\left[\Delta^t(\epsilon)\right]$.*

Our complexity result will be expressed in terms of the quantity:

$$\delta^k(\epsilon) \quad \stackrel{\text{def}}{=} \quad \mathbb{E}\left[\frac{1}{k}\sum_{t=0}^{k-1}\Delta^t(\epsilon)\right] = \frac{1}{k}\sum_{t=0}^{k-1}\mathbb{E}\left[\Delta^t(\epsilon)\right]. \tag{16}$$

Theorem 9. *For all $k \geq 1$ we have*

$$\delta^k(\epsilon) \leq \frac{4\left(D(y^*) - D(y^0)\right)}{k\epsilon^2}.$$

Note that if $\Delta^k(\epsilon) = 0$, it does not mean the primal iterate x^k is optimal. This only implies that the values of all pairs of directly connected nodes are differ by less than ϵ.

3.3 Private Gossip via Controlled Noise Insertion

In this section, we present the Gossip algorithm with Controlled Noise Insertion. As mentioned in the introduction, the approach is similar the technique proposed in [36, 37]. Those works, however, address only algorithms in the synchronous setting, while our work is the first to use this idea in the asynchronous setting. Unlike the above, we provide finite time convergence guarantees and allow each node to add the noise differently, which yields a stronger result.

In our approach, each node adds noise to the computation independently of all other nodes. However, the noise added is correlated between iterations for each node. We assume that every node owns two parameters — initial magnitude of the generated noise σ_i^2 and rate of decay of the noise ϕ_i. The node inserts noise $w_i^{t_i}$ to the system every time that an edge corresponding to the node was chosen, where variable t_i carries an information how many times the noise was added to the system in the past by node i. Thus, if we denote by t the current number of iterations, we have $\sum_{i=1}^n t_i = 2t$.

In order to ensure convergence to the optimal solution, we need to choose a specific structure of the noise in order to guarantee the mean of the values x_i converges to the initial mean. In particular, in each iteration a node i is selected, we subtract the noise that was added last time, and add a fresh noise with smaller magnitude:

$$w_i^{t_i} = \phi_i^{t_i} v_i^{t_i} - \phi_i^{t_i-1} v_i^{t_i-1},$$

where $0 \leq \phi_i < 1, v_i^{-1} = 0$ and $v_i^{t_i} \sim N(0, \sigma_i^2)$ for all iteration counters $k_i \geq 0$ is independent to all other randomness in the algorithm. This ensures that all noise added initially is gradually withdrawn from the whole network.

After the addition of noise, a standard Gossip update is made, which sets the values of sampled nodes to their average. Hence, we have

$$\begin{aligned}
\lim_{t\to\infty}\mathbb{E}\left[\left(\overline{c} - \frac{1}{n}\sum_{i=1}^n x_i^t\right)^2\right] &= \lim_{t\to\infty}\mathbb{E}\left[\left(\frac{1}{n}\sum_{i=1}^n \phi_i^{t_i-1}v_i^{t_i-1}\right)^2\right] \quad \leq \quad \lim_{t\to\infty}\mathbb{E}\left[\frac{1}{n}\sum_{i=1}^n\left(\phi_i^{t_i-1}v_i^{t_i-1}\right)^2\right] \\
&= \frac{1}{n}\lim_{t\to\infty}\sum_{i=1}^n\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}\right)^2\right] \quad = \quad \frac{1}{n}\lim_{t\to\infty}\sum_{i=1}^n\mathbb{E}\left[\phi_i^{2t_i-2}\right]\mathbb{E}\left[\left(v_i^{t_i-1}\right)^2\right] \\
&= \frac{1}{n}\lim_{t\to\infty}\sum_{i=1}^n\mathbb{E}\left[\phi_i^{2t_i-2}\right]\sigma_i^2 \quad = \quad \frac{1}{n}\sum_{i=1}^n\sigma_i^2\lim_{t\to\infty}\mathbb{E}\left[\phi_i^{2t_i-2}\right] \\
&= 0,
\end{aligned}$$

as desired.

It is not the purpose of this paper to define any quantifiable notion of protection of the initial values formally. However, we note that it is likely the case that the protection of private value c_i will be stronger for bigger σ_i and for ϕ_i closer to 1.

For simplicity, we provide only the primal algorithm below.

Algorithm 4 (Primal form)

Input: vector of private values $c \in \mathbb{R}^n$; initial variances $\sigma_i^2 \in \mathbb{R}_+$ and variance decrease rate ϕ_i such that $0 \leq \phi_i < 1$ for all nodes i.
Initialize: Set $x^0 = c$; $t_1 = t_2 = \cdots = t_n = 0$, $v_1^{-1} = v_2^{-1} = \cdots = v_n^{-1} = 0$.
for $t = 0, 1, \ldots k-1$ **do**

1. Choose node $e = (i, j) \in \mathcal{E}$ uniformly at random
2. Generate $v_i^{t_i} \sim N(0, \sigma_i^2)$ and $v_j^{t_j} \sim N(0, \sigma_j^2)$
3. Set
$$w_i^{t_i} = \phi_i^{t_i} v_i^{t_i} - \phi_i^{t_i - 1} v_i^{t_i - 1}$$
$$w_j^{t_j} = \phi_j^{t_j} v_j^{t_j} - \phi_j^{t_j - 1} v_j^{t_j - 1}$$
4. Update the primal variable:
$$x_i^{t+1} = x_j^{t+1} = \frac{x_i^t + w_i^{t_i} + x_j^t + w_j^{t_j}}{2}, \ \forall\, l \neq i, j : \ x_l^{t+1} = x_l^t$$
5. Set $t_i = t_i + 1$, $t_j = t_j + 1$

end
return x^k

We now provide results of dual analysis of Algorithm 4. The following lemma provides us the expected decrease in dual suboptimality for each iteration.

Lemma 10. *Let d_i denote the number of neighbours of node i. Then,*

$$\begin{aligned} \mathbb{E}\left[D(y^*) - D(y^{t+1})\right] \leq \left(1 - \frac{\alpha(\mathcal{G})}{2m}\right) \mathbb{E}\left[D(y^*) - D(y^t)\right] + \frac{1}{4m} \sum_{i=1}^n d_i \sigma_i^2 \mathbb{E}\left[\phi_i^{2t_i}\right] \\ - \frac{1}{2m} \sum_{e \in \mathcal{E}} \mathbb{E}\left[\left(\phi_i^{t_i - 1} v_i^{t_i - 1} x_j^t + \phi_j^{t_j - 1} v_j^{t_j - 1} x_i^t\right)\right]. \end{aligned} \tag{17}$$

We use the lemma to prove our main result, in which we show linear convergence for the algorithm. For notational simplicity, we decided to have $\rho^t = (\rho)^t$, i.e. superscript of ρ denotes its power, not an iteration counter.

Theorem 11. *Let us define the following quantities:*

$$\begin{aligned} \rho &\overset{def}{=} 1 - \frac{\alpha(\mathcal{G})}{2m}, \\ \psi^t &\overset{def}{=} \frac{1}{\sum_{i=1}^n \left(d_i \sigma_i^2\right)} \sum_{i=1}^n d_i \sigma_i^2 \left(1 - \frac{d_i}{m}\left(1 - \phi_i^2\right)\right)^t. \end{aligned}$$

Then for all $k \geq 1$ we have the following bound

$$\mathbb{E}\left[D(y^*) - D(y^k)\right] \leq \rho^k \left(D(y^*) - D(y^0)\right) + \frac{\sum \left(d_i \sigma_i^2\right)}{4m} \sum_{t=1}^k \rho^{k-t} \psi^t.$$

Note that ψ^t is a weighted sum of t-th powers of real numbers smaller than one. For large enough t, this quantity will depend on the largest of these numbers. This brings us to define M as the set of indices i for which the quantity $1-\frac{d_i}{m}\left(1-\phi_i^2\right)$ is maximized:

$$M = \arg\max_i \left\{1-\frac{d_i}{m}\left(1-\phi_i^2\right)\right\}.$$

Then for any $i_{\max} \in M$ we have

$$\psi^t \approx \frac{1}{\sum_{i=1}^n \left(d_i\sigma_i^2\right)} \sum_{i\in M} d_i\sigma_i^2 \left(1-\frac{d_i}{m}\left(1-\phi_i^2\right)\right)^t = \frac{\sum_{i\in M} d_i\sigma_i^2}{\sum_{i=1}^n \left(d_i\sigma_i^2\right)} \left(1-\frac{d_{i_{\max}}}{m}\left(1-\phi_{i_{\max}}^2\right)\right)^t,$$

which means that increasing ϕ_j for $j \notin M$ will not substantially influence convergence rate.

Note that as soon as we have

$$\rho > 1-\frac{d_i}{m}\left(1-\phi_i^2\right) \tag{18}$$

for all i, the rate from theorem 11 will be driven by ρ^k (as $k\to\infty$) and we will have

$$\mathbb{E}\left[D(y^*)-D(y^k)\right] = \tilde{O}\left(\rho^k\right) \tag{19}$$

One can think of the above as a threshold: if there is i such that ϕ_i is large enough so that the inequality (18) does not hold, the convergence rate is driven by $\phi_{i_{\max}}$. Otherwise, the rate is not influenced by the insertion of noise. Thus, in theory, we do not pay anything in terms of performance as long as we do not hit the threshold. One might be interested in choosing ϕ_i so that the threshold is attained for all i, and thus $M=\{1,\dots,n\}$. This motivates the following result:

Corollary 12. *Let us choose*

$$\phi_i \overset{def}{=} \sqrt{1-\frac{\gamma}{d_i}} \tag{20}$$

for all i, where $\gamma \le d_{\min}$. Then

$$\mathbb{E}\left[D(y^*)-D(y^k)\right] \;\le\; \left(1-\min\left(\frac{\alpha(\mathcal{G})}{2m},\frac{\gamma}{m}\right)\right)^k \left(D(y^*)-D(y^0)+\frac{\sum_{i=1}^n\left(d_i\sigma_i^2\right)}{4m}k\right).$$

As a consequence, $\phi_i = \sqrt{1-\frac{\alpha(\mathcal{G})}{2d_i}}$ is the largest decrease rate of noise for node i such that the guaranteed convergence rate of the algorithm is not violated.

While the above result clearly states the important threshold, it is not always practical as $\alpha(\mathcal{G})$ might not be known. However, note that if we choose $\frac{nd_{\min}}{2(n-1)} \le \gamma \le d_{\min}$, we have $\min\left(\frac{\alpha(\mathcal{G})}{2m},\frac{\gamma}{m}\right) = \frac{\alpha(\mathcal{G})}{2m}$ since $\frac{\alpha(\mathcal{G})}{2} \le \frac{n}{n-1}\frac{d_{\min}}{2} \le \gamma$, where $e(\mathcal{G})$ denotes graph *edge connectivity*: the minimal number of edges to be removed so that the graph becomes disconnected. Inequality $\alpha(\mathcal{G}) \le \frac{n}{n-1}d_{\min}$ is a well known result in spectral graph theory [16]. As a consequence, if for all i we have

$$\phi_i \le \sqrt{1-\frac{(n-1)d_{\min}}{2nd_i}},$$

then the convergence rate is not driven by the noise.

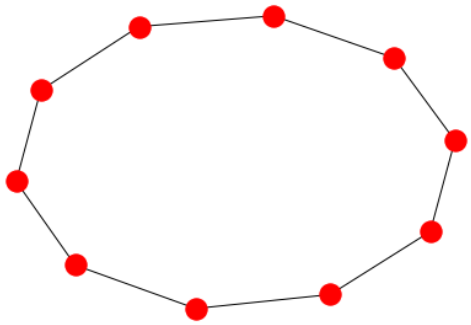
4 Numerical Evaluation

We devote this section to experimentally evaluate the performance of the Algorithms 2, 3 and 4 we proposed in the previous sections, applied to the Average Consensus problem. In the following experiments, we used the following popular graph topologies.

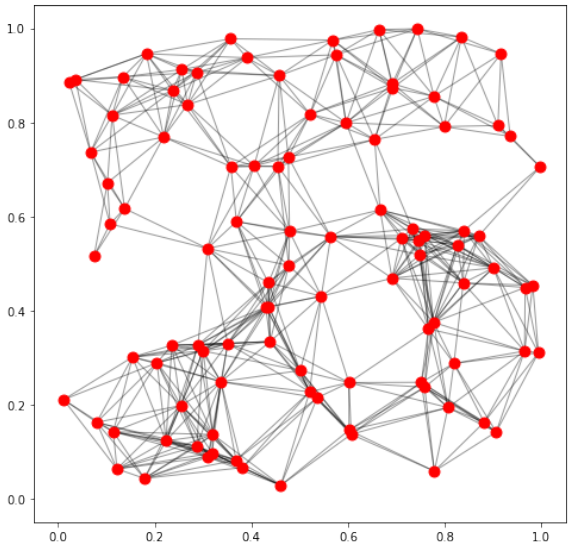
- *Cycle graph* with n nodes: $\mathcal{C}(n)$. In our experiments we choose $n = 10$. This small simple graph with regular topology is chosen for illustration purposes.

- *Random geometric graph* with n nodes and radius r: $\mathcal{G}(n, r)$. Random geometric graphs [41] are very important in practice because of their particular formulation which is ideal for modeling wireless sensor networks [26, 5]. In our experiments we focus on a 2-dimensional randomized geometric graph $\mathcal{G}(n, r)$ which is formed by placing n nodes uniformly at random in a unit square with edges between nodes which are having euclidean distance less than the given radius r. We set this to be to be $r = r(n) = \sqrt{\log(n)/n}$ — it is well know that the connectivity is preserved in this case [26]. We set $n = 100$.

An illustration of the two graphs appears is in Figure 1.



(a) Cycle Graph: $\mathcal{C}(10)$



(b) Random Geometric Graph: $\mathcal{G}(n, r)$

Figure 1: Illustration of the two graph topologies we focus on in this section.

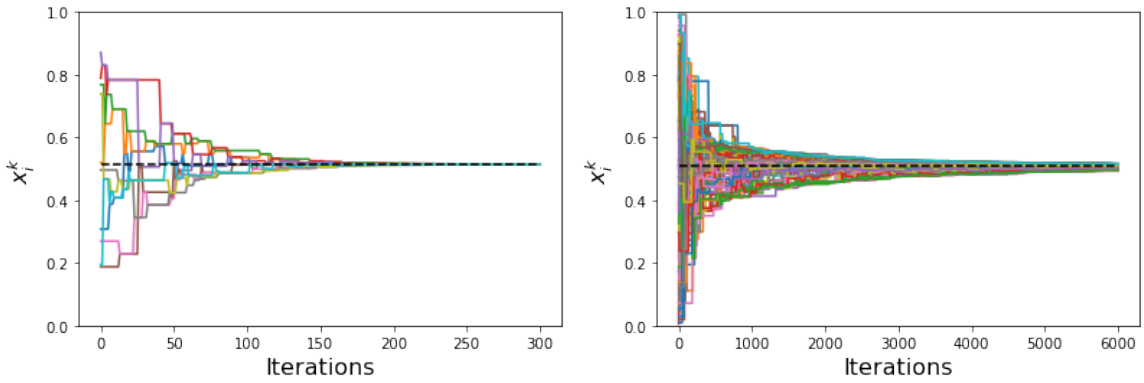
Setup: In all experiments we generate a vector with of initial values c_i from a uniform distribution over $[0, 1]$. We run several experiments and present two kinds of figures that helps us to understand how the algorithms evolve and verify the theoretical results of the previous sections. These figures are:

1. The evolution of the initial values of the nodes. In these figures we plot how the trajectory of the values x_i^k of each node i evolves throughout iterations. The black doted horizontal line represents the exact average consensus value which all nodes should approach, and thus all other lines should approach this level.

2. The evolution of the relative error measure $q^t \overset{\text{def}}{=} \frac{\|x^t - x^*\|^2}{\|x^0 - x^*\|^2}$.

We run each method for several parameters and for a pre-specified number of iterations not necessarily the same for each experiment. In each figure we have the relative error, both in normal scale or logarithmic scale, on the vertical axis and number of iterations on the horizontal axis.

To illustrate the first concept, we provide a simple example with the evolution of the initial values x_i^k for the case of the Standard Gossip algorithm [5] in Figure 2. The horizontal black dotted line represents the average consensus value. It is the exact average of the initial values c_i of the nodes in the network.



(a) Cycle Graph (b) Random Geometric Graph

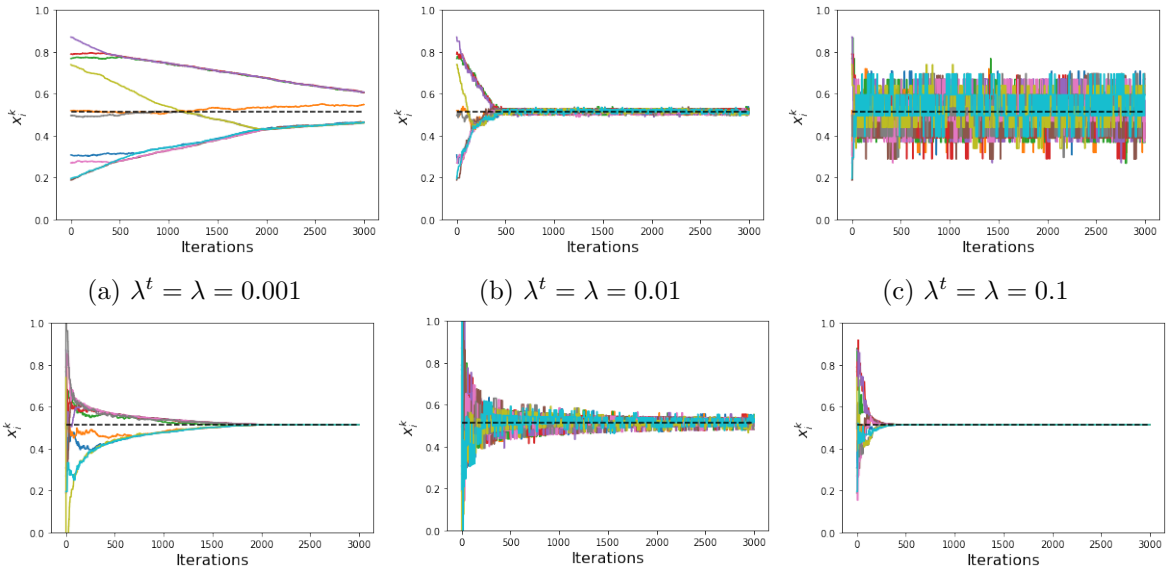
Figure 2: Trajectories of the values x_i^t for the Standard Gossip algorithm for Cycle Graph and a random geometric graph. Each line corresponds to x_i for some i.

In the rest of this section we evaluate the performance of the novel algorithms we propose, and contrast with the above Standard Gossip algorithm, which we refer to as "Baseline" in the following figures labels.

4.1 Private Gossip via Binary Oracle

In this section we evaluate the performance of Algorithm 2 presented in Section 3.1. In the algorithm, the input parameters are the positive stepsizes $\{\lambda^t\}_{t=0}^{\infty}$. The goal of the experiments is to compare the performance of the proposed algorithm using different choices of λ^t.

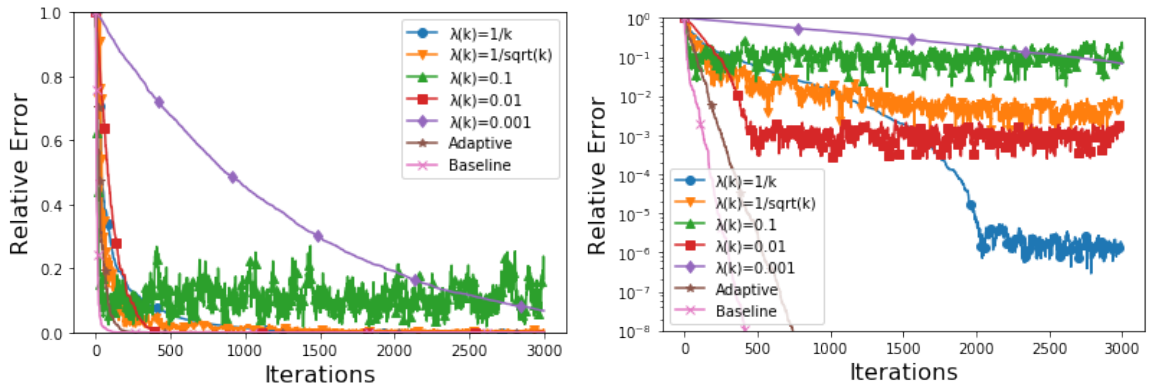
In particular, we use decreasing sequences of stepsizes $\lambda^t = 1/t$ and $\lambda^t = 1/\sqrt{t}$, and three different fixed values for the stepsizes $\lambda^t = \lambda \in \{0.001, 0.01, 0.1\}$. We also include the adaptive choice $\lambda^t = \frac{1}{4m}\sum_{e\in\mathcal{E}} |x_i^t - x_j^t|$ which we have proven to converge with linear rate in Theorem 7. We compare these choices in Figures 4 and 6, along with the Standard Gossip algorithm for clear comparison.



(a) $\lambda^t = \lambda = 0.001$ (b) $\lambda^t = \lambda = 0.01$ (c) $\lambda^t = \lambda = 0.1$

(d) $\lambda^t = \frac{1}{k}$ (e) $\lambda^t = \frac{1}{\sqrt{k}}$ (f) $\lambda^t =$ Adaptive

Figure 3: Trajectories of the values of x_i^t for Binary Oracle run on the cycle graph.



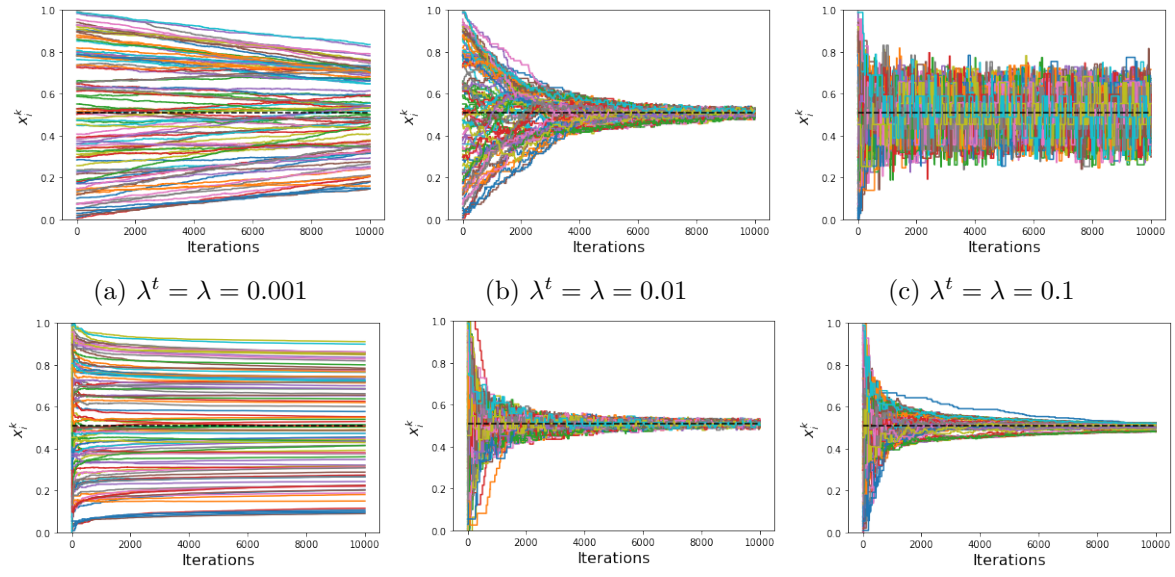
(a) Linear Scale

(b) Logarithmic Scale

Figure 4: Convergence of the Binary Oracle run on the cycle graph.

In general, we clearly see what is expected with the constant stepsizes — that they converge to a certain neighbourhood and oscillate around optimum. With smaller stepsize, this neighbourhood is more accurate, but it takes longer to reach. With decreasing stepsizes, Theorem 6 suggests that λ^t of order $1/\sqrt{t}$ should be optimal. Figure 6 demonstrates this, as the choice of $\lambda^t = 1/t$ decreases the stepsizes too quickly. However, this is not the case in Figure 4 in which we observe the opposite effect. This is due to the cycle graph being small and simple, and hence the diminishing stepsize becomes problem only after relatively large number of iterations. With the adaptive choice of stepsizes, we recover linear convergence rate as predicted by Theorem 7.

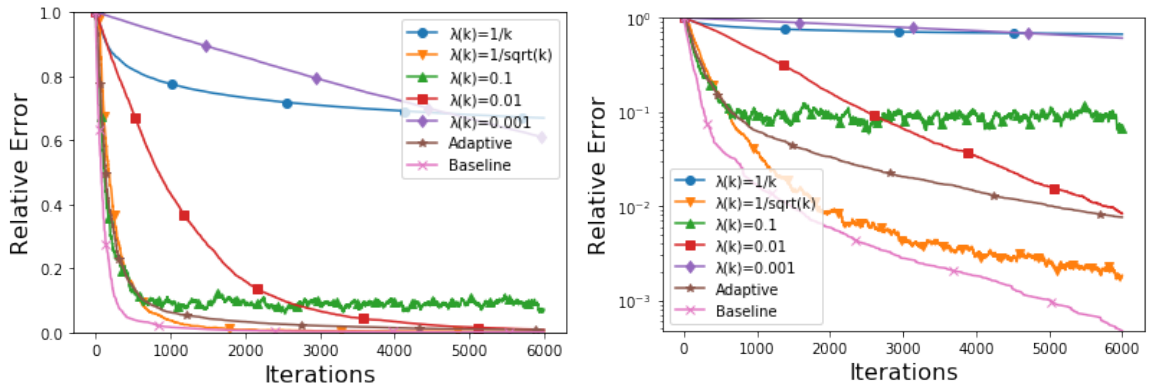
The results in Figure 6 show one surprising comparison. The adaptive choice of stepsizes does not seem to perform better than $\lambda^t = 1/\sqrt{t}$. However, we verified that when running for more iterations, the linear rate of adaptive stepsize is present and converges significantly faster to higher accuracies. We chose to present the results for 6000 iterations since we found it overall more clean.



(a) $\lambda^t = \lambda = 0.001$ (b) $\lambda^t = \lambda = 0.01$ (c) $\lambda^t = \lambda = 0.1$

(d) $\lambda^t = \frac{1}{t}$ (e) $\lambda^t = \frac{1}{\sqrt{t}}$ (f) $\lambda^t =$ Adaptive

Figure 5: Trajectories of the values of x_i^t for Binary Oracle run on the random geometric graph.



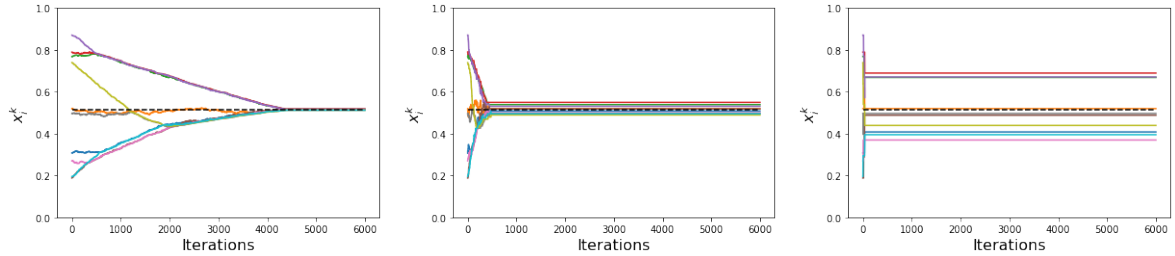
(a) Linear Scale (b) Logarithmic Scale

Figure 6: Convergence of the Binary Oracle run on the random geometric graph.

4.2 Private Gossip via ϵ-Gap Oracle

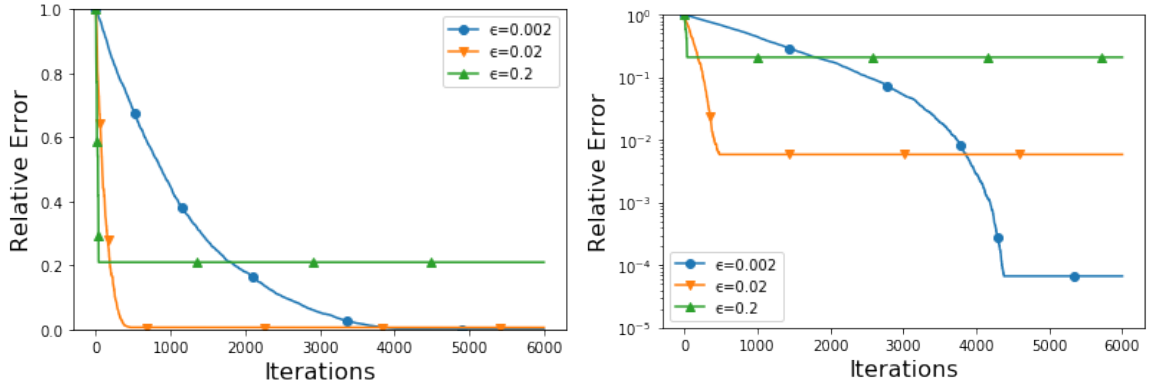
In this section we evaluate the performance of the Algorithm 3 presented in Section 3.2. In the algorithm, the input parameter is the positive error tolerance variable ϵ. For experimental evaluation. we choose three different values for the input, $\epsilon \in \{0.2, 0.02, 0.002\}$, and again use the same cycle and random geometric graphs. The trajectories of the values x_i^t are presented in Figures 7 and 9, respectively. The performance of the algorithm in terms of the relative error is presented in Figures 8 and 10.

The performance is exactly matching the expectation — with larger ϵ, the method converges very fast to a wide neighbourhood of the optimum. For a small value, it converges much closer to the optimum, but it requires more iterations.



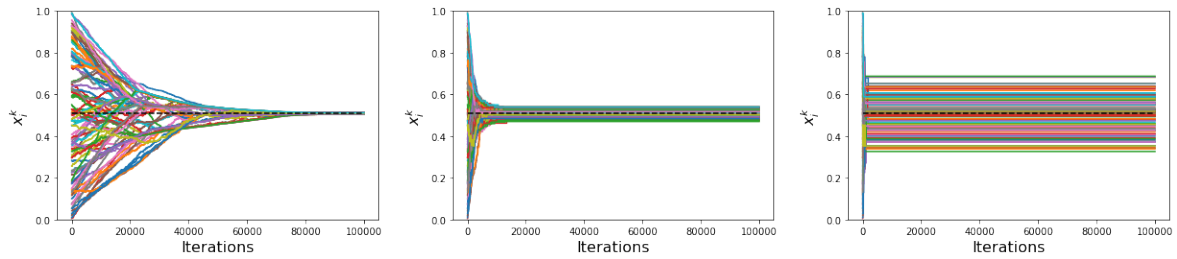
(a) $\epsilon = 0.002$ (b) $\epsilon = 0.02$ (c) $\epsilon = 0.2$

Figure 7: Trajectories of the values of x_i^t for ϵ-Gap Oracle run on the cycle graph.



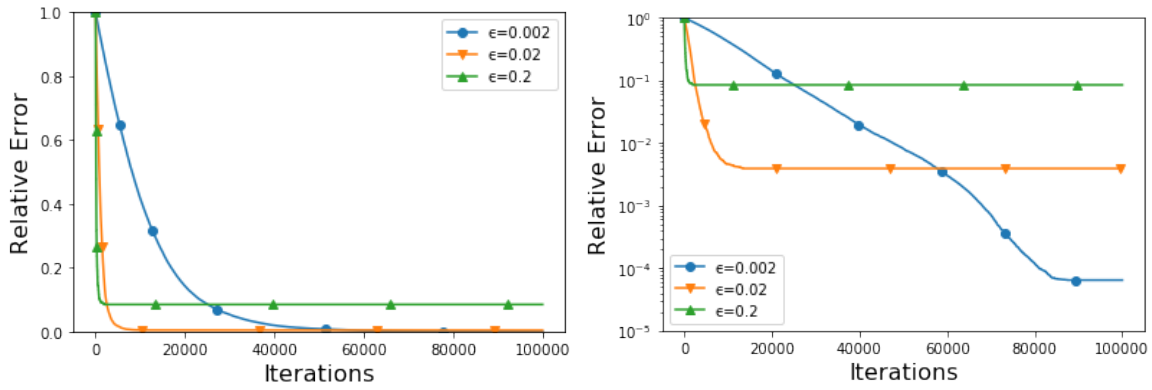
(a) Linear Scale (b) Logarithmic Scale

Figure 8: Convergence of the ϵ-Gap Oracle run on the cycle graph.



(a) $\epsilon = 0.002$ (b) $\epsilon = 0.02$ (c) $\epsilon = 0.2$

Figure 9: Trajectories of the values of x_i^t for ϵ-Gap Oracle run on the random geometric graph.



(a) Linear Scale (b) Logarithmic Scale

Figure 10: Convergence of the ϵ-Gap Oracle run on the random geometric graph.

4.3 Private Gossip via Controlled Noise Insertion

In this section we evaluate the performance of Algorithm 4 presented in Section 3.3. This algorithm has two different parameters for each node i. These are the initial variance $\sigma_i^2 \geq 0$ and the rate of decay, ϕ_i, of the noise.

To evaluate the impact of these parameters, we perform several experiments. As earlier, we use the same graph structures for evaluation: cycle graph and random geometric graph. The algorithm converges with a linear rate depending on maximum of two factors — see Theorem 11 and Corollary 12. We will verify that this is indeed the case, and for values of ϕ_i above a certain threshold, the convergence is driven by the rate at which the noise decays. This is true for both identical values of ϕ_i for all i, and for varying values as per (20). We further demonstrate the latter is superior in the sense that it enables insertion of more noise, without sacrificing the convergence speed. Finally, we study the effect of various magnitudes of the noise inserted initially.

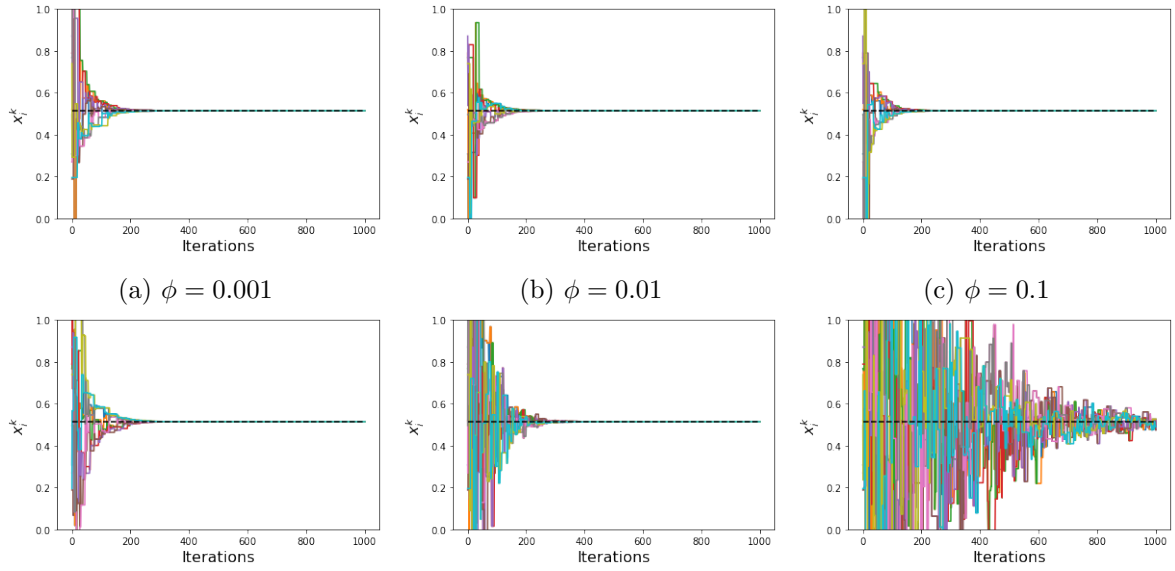
4.3.1 Fixed variance, identical decay rates

In this part, we run Algorithm 4 with $\sigma_i = 1$ for all i, and set $\phi_i = \phi$ for all i and some ϕ. We study the effect of varying the value of ϕ on the convergence of the algorithm.

In both Figures 12b and 14b, we see that for small values of ϕ, we eventually recover the same rate of linear convergence as the Standard Gossip algorithm. If the value of ϕ is sufficiently close to 1 however, the rate is driven by the noise and not by the convergence of the Standard Gossip algorithm. This value is $\phi = 0.98$ for cycle graph, and $\phi = 0.995$ for the random geometric graph in the plots we present.

Looking at the individual runs for small values of ϕ in Figure 14b, we see some variance in terms of when the asymptotic rate is realized. We would like to point out that this *does not* provide additional insight into whether specific small values of ϕ are in general better for the following reason. The Standard Gossip algorithm is itself a randomized algorithm, with an inherent uncertainty in the

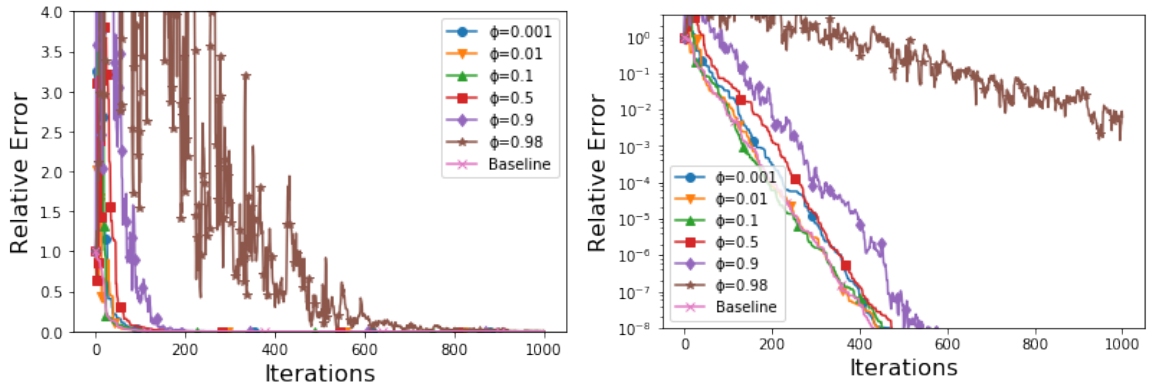
convergence of any particular run. If we ran the algorithms multiple times, we observe variance in the evolution of the suboptimality of similar magnitude, just as what we see in the figure. Hence, the variance is expected, and not significantly influenced by the noise.



(a) $\phi = 0.001$ (b) $\phi = 0.01$ (c) $\phi = 0.1$

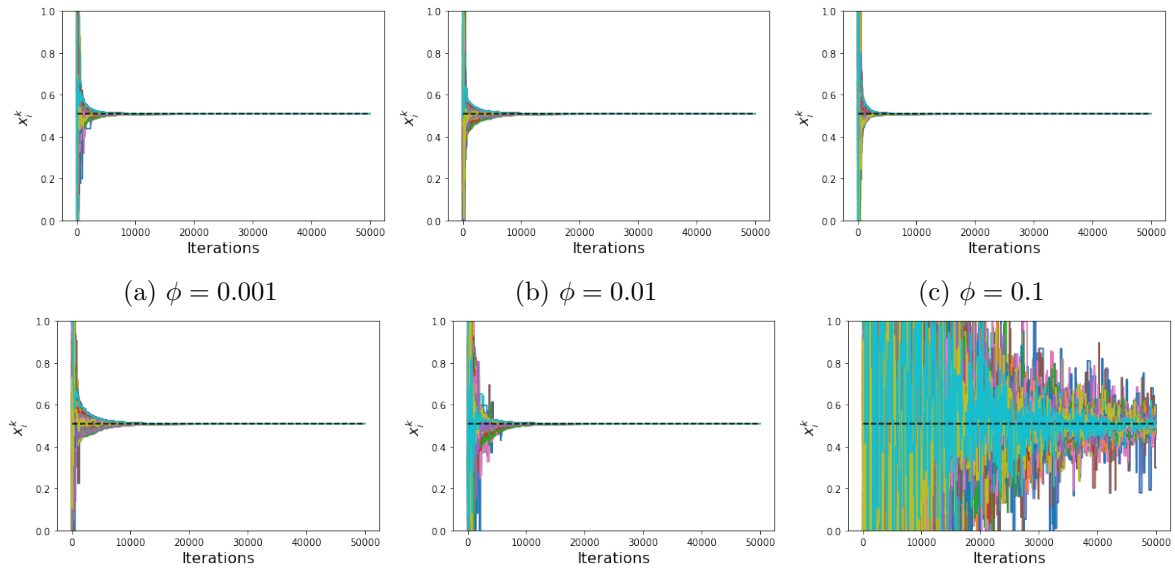
(d) $\phi = 0.5$ (e) $\phi = 0.9$ (f) $\phi = 0.98$

Figure 11: Trajectories of the values of x_i^t for Controlled Noise Insertion run on the cycle graph for different values of ϕ.



(a) Linear Scale (b) Logarithmic Scale

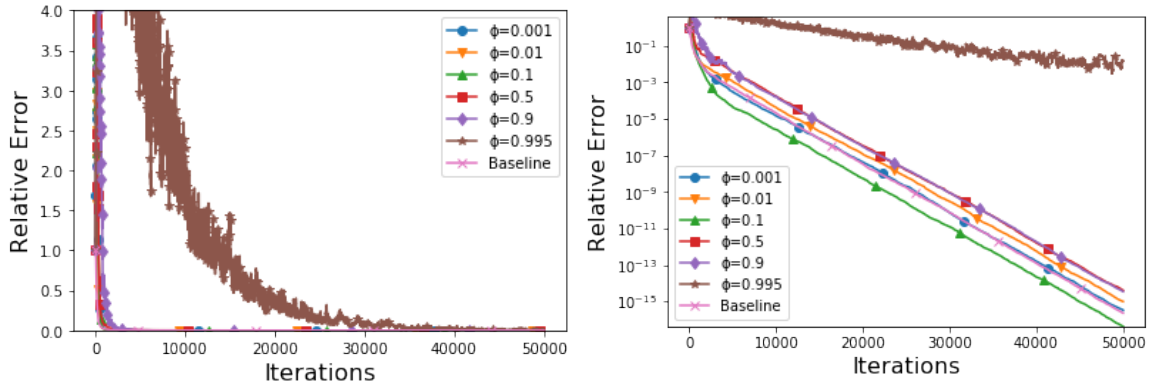
Figure 12: Convergence of the Controlled Noise Insertion run on the cycle graph for different values of ϕ.



(a) $\phi = 0.001$ (b) $\phi = 0.01$ (c) $\phi = 0.1$

(d) $\phi = 0.5$ (e) $\phi = 0.9$ (f) $\phi = 0.995$

Figure 13: Trajectories of the values of x_i^t for Controlled Noise Insertion run on the random geometric graph for different values of ϕ.



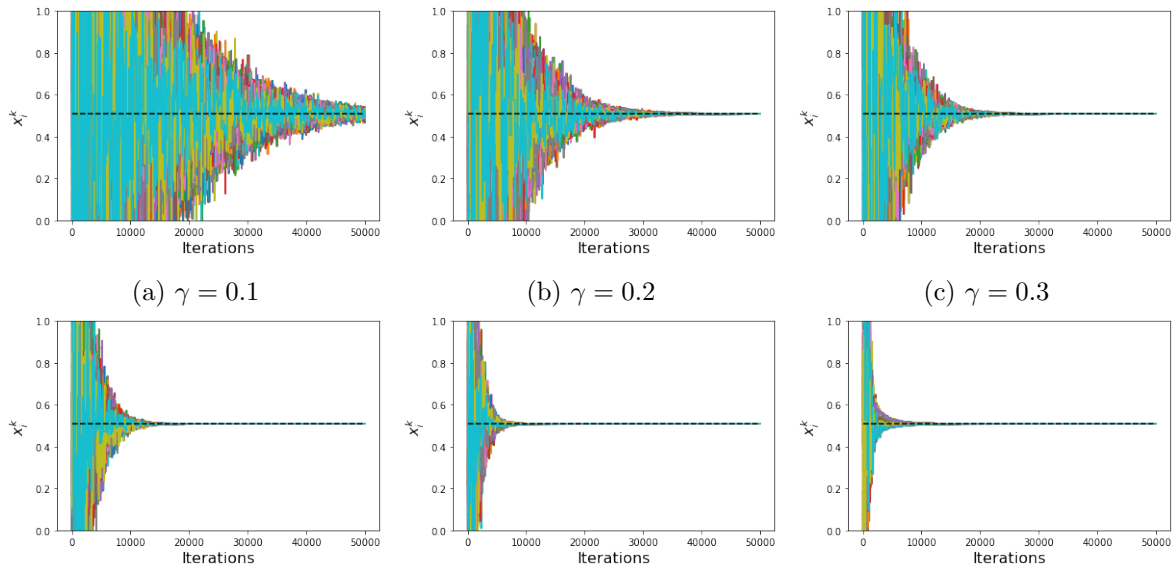
(a) Linear Scale (b) Logarithmic Scale

Figure 14: Convergence of the Controlled Noise Insertion run on the random geometric graph for different values of ϕ.

4.3.2 Variance 1 and different decay rates

In this section, we perform similar experiment as above, but let the values ϕ_i be vary for different nodes i. This is controlled by the choice of γ as in (20). Note that by decreasing γ, we increase ϕ_i, and thus smaller γ means the noise decays at a slower rate. Here, due to the regular structure of the cycle graph, we present only results for the random geometric graph.

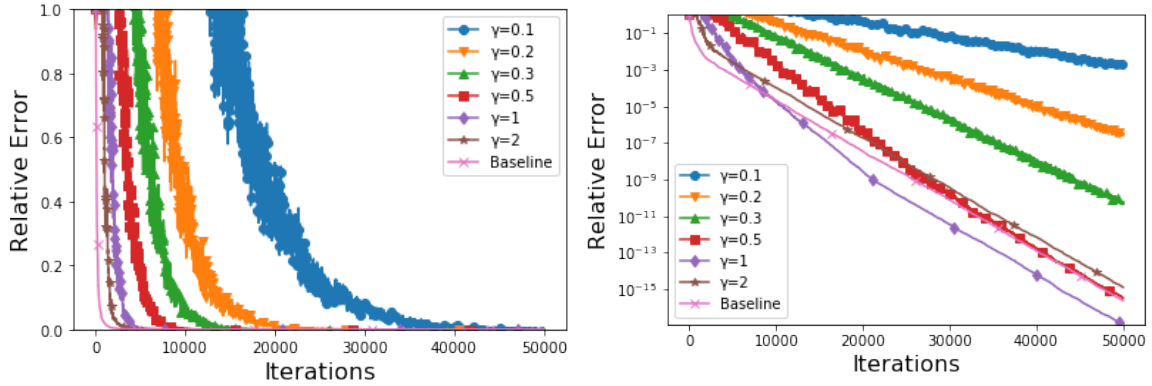
It is not straightforward to compare this setting with the setting of identical ϕ_i, and we return to it in the next section. Here we only remark that we again see the existence of a threshold predicted by theory, beyond which the convergence is dominated by the inserted noise. Otherwise, we recover the rate of the Standard Gossip algorithm.



(a) $\gamma = 0.1$ (b) $\gamma = 0.2$ (c) $\gamma = 0.3$

(d) $\gamma = 0.5$ (e) $\gamma = 1$ (f) $\gamma = 2$

Figure 15: Trajectories of the values of x_i^t for Controlled Noise Insertion run on the random geometric graph for different values of ϕ_i, controlled by γ.



(a) Normal Scale (b) Logarithmic Scale

Figure 16: Convergence of the Controlled Noise Insertion run on the random geometric graph for different values of ϕ_i, controlled by γ.

4.3.3 Impact of varying ϕ_i

In this experiment, we demonstrate the practical utility of letting the rate of decay ϕ_i to be different on each node i. In order to do so, we run experiment on the random geometric graph and compare the settings investigated in the previous two sections — the noise decay rate driven by ϕ, or by γ.

In first place, we choose the values of ϕ_i such that that the two factors in Corollary 12 are equal. For the particular graph we used, this corresponds to $\gamma \approx 0.17$ with $\phi_i = \sqrt{1 - \frac{\alpha(\mathcal{G})}{2d_i}}$. Second, we make the factors equal, but with constraint of having ϕ_i to be equal for all i. This corresponds to $\phi_i \approx 0.983$ for all i.

The performance for a large number of iterations is displayed in left side of Figure 17. We see that theabove two choices indeed yield very similar practical performance, which also eventually matches the rate predicted by theory. For complete comparison, we also include performance of the Standard Gossip algorithm.

The important message is conveyed in the histogram in the right side of Figure 17. The histogram shows the distribution of the values of ϕ_i for different nodes i. The minimum of these values is what we needed in the case of identical ϕ_i for all i. However, most of the values are significantly higher.

This means, that if we allow the noise decay rates to depend on the number of neighbours, we are able to increase the amount of noise inserted, without sacrificing practical performance. This is beneficial, as more noise will likely be beneficial for any formal notion of protection of the initial values.

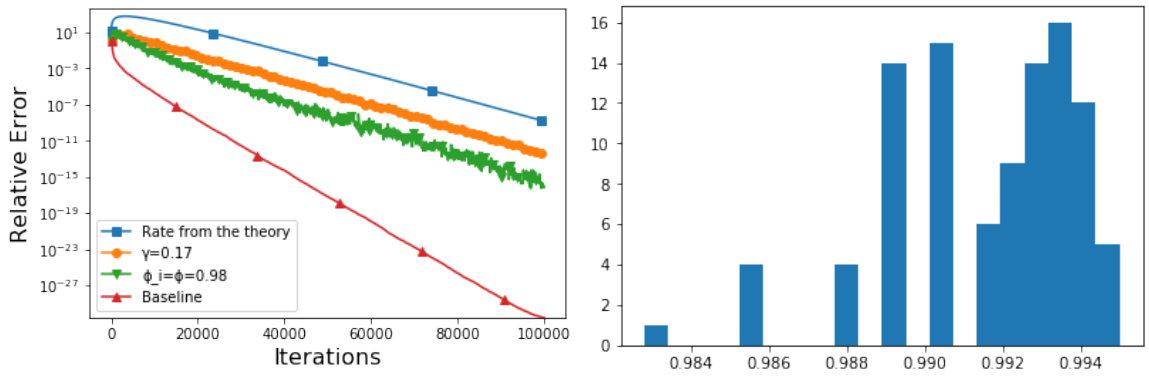


Figure 17: Left: Performance of the noise oracle with noise decrease rate chosen according to Corollary 12. Right: Histogram of of distribution of ϕ_i

5 Conclusion

In this work we addressed the Average Consensus problem via novel asynchronous randomized gossip algorithms. We propose algorithmic tools for protection of the private values each node in the network holds initially. However, we do not quantify any formal notion of privacy protection achievable using these tools; this is left for future research.

In particular, we propose two ways to achieve this goal. First, which we believe is the first of its kind, weakens the oracle used in the gossip framework, to provide only categorical (or even binary) information to each participating node about the value of the other node. In the second approach, we systematically inject and withdraw noise throughout the iterations, so as to ensure convergence to the average consensus value. In all cases, we provide explicit convergence rates and evaluate practical convergence on common simulated network topologies.

References

[1] Zeyuan Allen-Zhu, Zheng Qu, Peter Richtárik, and Yang Yuan. Even faster accelerated coordinate descent using non-uniform sampling. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1110–1119, 2016.

[2] Tuncer Can Aysal, Mehmet Ercan Yildiz, Anand D Sarwate, and Anna Scaglione. Broadcast gossip algorithms for consensus. *IEEE Transactions on Signal processing*, 57(7):2748–2761, 2009.

[3] Florence Bénézit, Alexandros G Dimakis, Patrick Thiran, and Martin Vetterli. Order-optimal consensus through randomized path averaging. *IEEE Transactions on Information Theory*, 56(10):5150–5167, 2010.

[4] Dimitri P Bertsekas and John N Tsitsiklis. *Parallel and distributed computation: numerical methods*, volume 23. Prentice hall Englewood Cliffs, NJ, 1989.

[5] Stephen Boyd, Arpita Ghosh, Balaji Prabhakar, and Devavrat Shah. Randomized gossip algorithms. *IEEE Transactions on Information Theory*, 14(SI):2508–2530, 2006.

[6] Ming Cao, Daniel A Spielman, and Edmund M Yeh. Accelerated gossip algorithms for distributed computation. In *Proceedings of the 44th Annual Allerton Conference on Communication, Control, and Computation*, pages 952–959, 2006.

[7] Antonin Chambolle, Matthias J. Ehrhardt, Peter Richtárik, and Carola-Bibiane Schoenlieb. Stochastic primal-dual hybrid gradient algorithm with arbitrary sampling and imaging applications. *arXiv:1706.04957*, 2017.

[8] Ronald Cramer, Ivan Bjerre Damgård, and Jesper Buus Nielsen. *Secure Multiparty Computation and Secret Sharing*. Cambridge University Press, 2015.

[9] Dominik Csiba, Zheng Qu, and Peter Richtárik. Stochastic Dual Coordinate Ascent with Adaptive Probabilities. *ICML 2015*, 2015.

[10] George Cybenko. Dynamic load balancing for distributed memory multiprocessors. *Journal of Parallel and Distributed Computing*, 7(2):279–301, 1989.

[11] Alexandros DG Dimakis, Anand D Sarwate, and Martin J Wainwright. Geographic gossip: Efficient averaging for sensor networks. *IEEE Transactions on Signal Processing*, 56(3):1205–1216, 2008.

[12] Alexandros G Dimakis, Soummya Kar, José MF Moura, Michael G Rabbat, and Anna Scaglione. Gossip algorithms for distributed signal processing. *Proceedings of the IEEE*, 98(11):1847–1864, 2010.

[13] Cynthia Dwork, Aaron Roth, et al. The algorithmic foundations of differential privacy. *Foundations and Trends® in Theoretical Computer Science*, 9(3–4):211–407, 2014.

[14] Olivier Fercoq and Peter Richtárik. Smooth minimization of nonsmooth functions by parallel coordinate descent. *arXiv:1309.5885*, 2013.

[15] Olivier Fercoq and Peter Richtárik. Accelerated, parallel and proximal coordinate descent. *SIAM Journal on Optimization*, (25):1997–2023, 2015.

[16] Miroslav Fiedler. Algebraic connectivity of graphs. *Czechoslovak mathematical journal*, 23(2):298–305, 1973.

[17] Mauro Franceschelli, Alessandro Giua, and Carla Seatzu. Distributed averaging in sensor networks based on broadcast gossip algorithms. *IEEE Sensors Journal*, 11(3):808–817, 2011.

[18] Nikolaos M Freris and Anastasios Zouzias. Fast distributed smoothing of relative measurements. In *IEEE 51st Annual Conference on Decision and Control (CDC)*, pages 1411–1416. IEEE, 2012.

[19] Valerio Freschi, Emanuele Lattanzi, and Alessandro Bogliolo. Accelerating distributed averaging in sensor networks: Randomized gossip over virtual coordinates. In *Sensors Applications Symposium (SAS), 2016 IEEE*, pages 1–6. IEEE, 2016.

[20] Valerio Freschi, Emanuele Lattanzi, and Alessandro Bogliolo. Fast distributed consensus through path averaging on random walks. *Wireless Pers Commun*, doi:10.1007/s11277-017-4451-5:1–15, 2017.

[21] Robert Mansel Gower, Donald Goldfarb, and Peter Richtárik. Stochastic block BFGS: squeezing more curvature out of data. In *Proceedings of The 33rd International Conference on Machine Learning*, pages 1869–1878, 2016.

[22] Robert Mansel Gower and Peter Richtárik. Randomized iterative methods for linear systems. *SIAM Journal on Matrix Analysis and Applications*, 36(4):1660–1690, 2015.

[23] Robert Mansel Gower and Peter Richtárik. Stochastic dual ascent for solving linear systems. *arXiv:1512.06890*, 2015.

[24] Robert Mansel Gower and Peter Richtárik. Linearly convergent randomized iterative methods for computing the pseudoinverse. *arXiv preprint arXiv:1612.06255*, 2016.

[25] Robert Mansel Gower and Peter Richtárik. Randomized quasi-Newton updates are linearly convergent matrix inversion algorithms. *arXiv preprint arXiv:1602.01768*, 2016.

[26] Piyush Gupta and Panganmala R Kumar. The capacity of wireless networks. *IEEE Transactions on Information Theory*, 46(2):388–404, 2000.

[27] Fenghua He, A Stephen Morse, Ji Liu, and Shaoshuai Mou. Periodic gossiping. *IFAC Proceedings Volumes*, 44(1):8718–8723, 2011.

[28] Zhenqi Huang, Sayan Mitra, and Geir Dullerud. Differentially private iterative synchronous consensus. In *Proceedings of the 2012 ACM workshop on Privacy in the electronic society*, pages 81–90. ACM, 2012.

[29] Ye Yu Jun and Michael Rabbat. Performance comparison of randomized gossip, broadcast gossip and collection tree protocol for distributed averaging. In *IEEE 5th International Workshop on Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP)*, pages 93–96. IEEE, 2013.

[30] David Kempe, Alin Dobra, and Johannes Gehrke. Gossip-based computation of aggregate information. In *Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on*, pages 482–491. IEEE, 2003.

[31] Yin Tat Lee and Aaron Sidford. Efficient accelerated coordinate descent methods and faster algorithms for solving linear systems. In *Foundations of Computer Science (FOCS), 2013 IEEE 54th Annual Symposium on*, pages 147–156. IEEE, 2013.

[32] Dennis Leventhal and Adrian S. Lewis. Randomized methods for linear constraints: Convergence rates and conditioning. *Mathematics of Operations Research*, 35(3):641–654, 2010.

[33] Ji Liu, Brian D.O. Anderson, Ming Cao, and A Stephen Morse. Analysis of accelerated gossip algorithms. *Automatica*, 49(4):873–883, 2013.

[34] Ji Liu, Shaoshuai Mou, A Stephen Morse, Brian DO Anderson, and Changbin Yu. Deterministic gossiping. *Proceedings of the IEEE*, 99(9):1505–1524, 2011.

[35] Nicolas Loizou and Peter Richtárik. A new perspective on randomized gossip algorithms. In *4th IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2016.

[36] Nicolaos E Manitara and Christoforos N Hadjicostis. Privacy-preserving asymptotic average consensus. In *Control Conference (ECC), 2013 European*, pages 760–765. IEEE, 2013.

[37] Yilin Mo and Richard M Murray. Privacy preserving average consensus. *IEEE Transactions on Automatic Control*, 62(2):753–765, 2017.

[38] Shaoshuai Mou, Changbin Yu, Brian DO Anderson, and A Stephen Morse. Deterministic gossiping with a periodic protocol. In *Decision and Control (CDC), 2010 49th IEEE Conference on*, pages 5787–5791. IEEE, 2010.

[39] Yurii Nesterov. Efficiency of coordinate descent methods on huge-scale optimization problems. *SIAM Journal on Optimization*, 22(2):341–362, 2012.

[40] Reza Olfati-Saber, J Alex Fax, and Richard M Murray. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95(1):215–233, 2007.

[41] Mathew Penrose. *Random geometric graphs*. Number 5. Oxford University Press, 2003.

[42] Zheng Qu and Richtárik. Coordinate descent with arbitrary sampling I: algorithms and complexity. *Optimization Methods and Software*, 31(5):829–857, 2016.

[43] Zheng Qu and Richtárik. Coordinate descent with arbitrary sampling II: expected separable overapproximation. *Optimization Methods and Software*, 31(5):858–884, 2016.

[44] Zheng Qu, Peter Richtárik, and Tong Zhang. Quartz: Randomized dual coordinate ascent with arbitrary sampling. In *Advances in Neural Information Processing Systems 28*, 2015.

[45] Wei Ren, Randal W Beard, and Ella M Atkins. Information consensus in multivehicle cooperative control. *IEEE Control Systems*, 27(2):71–82, 2007.

[46] Peter Richtárik and Martin Takáč. Iteration complexity of randomized block-coordinate descent methods for minimizing a composite function. *Mathematical Programming*, 144(2):1–38, 2014.

[47] Peter Richtárik and Martin Takáč. Parallel coordinate descent methods for big data optimization. *Mathematical Programming*, 156(1):433–484, 2016.

[48] Peter Richtárik and Martin Takáč. Stochastic reformulations of linear systems: algorithms and convergence theory. *arXiv preprint arXiv:1706.01108*, 2017.

[49] Devavrat Shah. Gossip algorithms. *Foundations and Trends® in Networking*, 3(1):1–125, 2009.

[50] Shai Shalev-Shwartz and Tong Zhang. Stochastic Dual Coordinate Ascent Methods for Regularized Loss Minimization. *Journal of Machine Learning Research*, 14:567–599, 2012.

[51] John Tsitsiklis, Dimitri Bertsekas, and Michael Athans. Distributed asynchronous deterministic and stochastic gradient optimization algorithms. *IEEE transactions on automatic control*, 31(9):803–812, 1986.

[52] John Nikolas Tsitsiklis. Problems in decentralized decision making and computation. Technical report, DTIC Document, 1984.

[53] Shaochuan Wu and Michael G Rabbat. Broadcast gossip algorithms for consensus on strongly connected digraphs. *IEEE Transactions on Signal Processing*, 61(16):3959–3971, 2013.

[54] Lin Xiao and Stephen Boyd. Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1):65–78, 2004.

[55] Lin Xiao, Stephen Boyd, and Sanjay Lall. A scheme for robust distributed sensor fusion based on average consensus. In *Proceedings of the 4th international symposium on Information processing in sensor networks*, page 9. IEEE Press, 2005.

[56] Changbin Brad Yu, Brian DO Anderson, Shaoshuai Mou, Ji Liu, Fenghua He, and A Stephen Morse. Distributed averaging using periodic gossiping. *IEEE Transactions on Automatic Control*, 2017.

A Proofs for Section 1

A.1 Proof of Lemma 1

Lemma 13.

$$\sum_{i=1}^{n}\left(\sum_{j=1}^{n}(x_j-x_i)\right)^2=\frac{n}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(x_j-x_i)^2 \tag{21}$$

Proof. Using simple algebra we have

$$\begin{aligned}\sum_{i=1}^{n}\left(\sum_{j=1}^{n}(x_j-x_i)\right)^2 &= \sum_{i=1}^{n}\left(\sum_{j=1}^{n}x_j-nx_i\right)^2\\ &= \sum_{i=1}^{n}\left(\left(\sum_{j=1}^{n}x_j\right)^2+n^2x_i^2-2nx_i\left(\sum_{j=1}^{n}x_j\right)\right)\\ &= n\left(\sum_{j=1}^{n}x_j\right)^2+n^2\sum_{i=1}^{n}x_i^2-2n\left(\sum_{j=1}^{n}x_j\right)^2\\ &= n^2\sum_{i=1}^{n}x_i^2-n\left(\sum_{i=1}^{n}x_i\right)^2.\end{aligned}$$

Manipulating right hand side of (21) we obtain

$$\begin{aligned}\frac{n}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}(x_j-x_i)^2 &= \frac{n}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\left(x_j^2+x_i^2-2x_ix_j\right)\\ &= n^2\sum_{i=1}^{n}x_i^2-n\sum_{i=1}^{n}\sum_{j=1}^{n}x_ix_j \quad = \quad n^2\sum_{i=1}^{n}x_i^2-n\left(\sum_{i=1}^{n}x_i\right)^2.\end{aligned}$$

Clearly, LHS and RHS of (21) are equal. □

In order to show (2) it is enough to notice that

$$\begin{aligned}\|\bar{c}\mathbf{1}-x\|^2 &= \sum_{i=1}^{n}(\bar{c}-x_i)^2 \quad = \quad \sum_{i=1}^{n}\left(\frac{1}{n}\sum_{j=1}^{n}x_j-x_i\right)^2\\ &= \sum_{i=1}^{n}\left(\sum_{j=1}^{n}\frac{1}{n}(x_j-x_i)\right)^2 \overset{(21)}{=} \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}\frac{1}{n}(x_j-x_i)^2\\ &= \frac{1}{2n}\sum_{i=1}^{n}\sum_{j=1}^{n}(x_j-x_i)^2.\end{aligned}$$

Note that we have

$$\frac{1}{nm}\left(\sum_{e=(i,j)\in\mathcal{E}}|x_i-x_j|\right)^2\le\frac{1}{n}\sum_{e\in\mathcal{E}}(x_i-x_j)^2\le\frac{1}{n}\sum_{(i,j)}(x_i-x_j)^2\overset{(2)}{=}\|\bar{c}\mathbf{1}-x\|^2,$$

which proves (3). On the other hand, we have

$$\frac{1}{\alpha(\mathcal{G})}\left(\sum_{e=(i,j)\in\mathcal{E}}|x_i-x_j|\right)^2\ge\frac{1}{\alpha(\mathcal{G})}\sum_{e\in\mathcal{E}}(x_i-x_j)^2\overset{(23)}{\ge}\|\bar{c}\mathbf{1}-x\|^2,$$

which concludes (4). Inequality (5) holds trivially.

B Proofs for Section 2.4

We now perform the analysis of Algorithm 1.

B.1 Proof of Lemma 4

Lemma 14. *The eigenvalues of* $\tilde{\mathbf{L}} = n\mathbf{I} - \mathbf{1}\mathbf{1}^\top$ *are* $\{0, n, n, \ldots, n\}$

Proof. Clearly, $\tilde{\mathbf{L}}\mathbf{1} = 0$. Consider some vector x such that $\langle x, \mathbf{1}\rangle = 0$. Then, $\tilde{\mathbf{L}}x = n\mathbf{I}x - \mathbf{1}\mathbf{1}^\top x = nx + \mathbf{1}\mathbf{1}^\top x = nx$ thus x is an eigenvector corresponding to eigenvalue n. Thus, we can pick $n-1$ linearly independent eigenvectors of $\tilde{\mathbf{L}}$ corresponding to eigenvalue n, which concludes the proof. □

The Laplacian matrix of $\mathcal{G}$ is the matrix $\mathbf{L} = \mathbf{A}^\top\mathbf{A}$. We have $\mathbf{L}_{ii} = d_i$ (degree of vertex i), $\mathbf{L}_{ij} = \mathbf{L}_{ji} = -1$ if $(i,j) \in \mathcal{E}$ and $\mathbf{L}_{ij} = 0$ otherwise. A simple computation reveals that for any $x \in \mathbb{R}^n$ we have

$$x^\top \mathbf{L} x = \sum_{e=(i,j)\in\mathcal{E}} (x_i - x_j)^2.$$

Let $\tilde{\mathbf{A}}$ be the $n(n-1)/2 \times n$ matrix corresponding to the complete graph $\tilde{\mathcal{G}}$ on $\mathcal{V}$. Let $\tilde{\mathbf{L}} = \tilde{\mathbf{A}}^\top\tilde{\mathbf{A}}$ be its Laplacian. We have $\tilde{\mathbf{L}}_{ii} = n-1$ for all i and $\tilde{\mathbf{L}}_{ij} = -1$ for $i \neq j$. So, $\tilde{\mathbf{L}} = n\mathbf{I} - \mathbf{1}\mathbf{1}^\top$. Then

$$x^\top\tilde{\mathbf{L}}x = n\|x\|^2 - \left(\sum_{i=1}^n x_i\right)^2 = \sum_{(i,j)} (x_i - x_j)^2.$$

Inequality (11) can therefore be recast as follows:

$$x^\top(n\mathbf{I} - \mathbf{1}\mathbf{1}^\top)x \leq x^\top\beta(\mathcal{G})\mathbf{L}x, \qquad x \in \mathbb{R}^n.$$

Let $\beta = \beta(\mathcal{G})$. Note that both $\tilde{\mathbf{L}}$ and $\beta\mathbf{L}$ are Hermitian thus have real eigenvalues and there exist an orthonormal basis of their eigenvectors. Suppose that $\{x_1, \ldots x_n\}$ are eigenvectors of βL corresponding to eigenvalues $\lambda_1(\beta\mathbf{L}), \lambda_2(\beta\mathbf{L}) \ldots, \lambda_n(\beta\mathbf{L})$. Without loss of generality assume that these eigenvectors form an orthonormal basis and $\lambda_1(\beta\mathbf{L}) \geq \cdots \geq \lambda_n(\beta\mathbf{L})$

Clearly, $\lambda_n(\beta\mathbf{L}) = 0$, $x_n = \mathbf{1}/\sqrt{n}$, and $\lambda_{n-1}(\beta\mathbf{L}) = n$. Lemma 14 states that eigenvalues of $\tilde{\mathbf{L}}$ are $\{0, n, n, \ldots, n\}$.

One can easily see that eigenvector corresponding to zero eigenvalue of $\tilde{\mathbf{L}}$ is x_n. Note that eigenvectors $x_1, \ldots, x_{n-1}$ generate an eigenspace corresponding to eigenvalue n of $\tilde{\mathbf{L}}$.

Consider some $x = \sum_{i=1}^n c_i x_i$, $c_i \in \mathbb{R}$ for all i. Then we have

$$x^\top\tilde{\mathbf{L}}x = \sum_{i=1}^n \lambda_i\left(\tilde{\mathbf{L}}\right) c_i^2 \leq \sum_{i=1}^n \lambda_i\left(\beta\mathbf{L}\right) c_i^2 = x^\top\beta\mathbf{L}x,$$

which concludes the proof.

B.2 Two Lemmas

We first establish two lemmas which will be needed to prove Theorem 5.

Lemma 15. *Assume that edge* $e = (i,j)$ *is selected in iteration* t *of Algorithm 1. Then*

$$D(y^{t+1}) - D(y^t) = \tfrac{1}{4}(x_i^t - x_j^t)^2. \tag{22}$$

Proof. We have $y^{t+1} = y^t + \lambda^t f_e$ where λ^t is chosen so that $D(y^{t+1}) - D(y^t)$ is maximized. Applying Lemma 3, we have

$$D(y^{t+1}) - D(y^t) = \max_{\lambda} -\lambda(x_i^t - x_j^t) - \lambda^2 = \tfrac{1}{4}(x_i^t - x_j^t)^2.$$

□

Lemma 16. *Let $x \in \mathbb{R}^n$ such that $\frac{1}{n}\sum_i x_i = \bar{c}$. Then*

$$\frac{1}{2}\|\bar{c}\mathbf{1} - x\|^2 \leq \frac{1}{2\alpha(\mathcal{G})} \sum_{e=(i,j)\in\mathcal{E}} (x_i - x_j)^2. \tag{23}$$

Proof.

$$\begin{aligned} \frac{1}{2}\|\bar{c}\mathbf{1} - x\|^2 &\overset{(2)}{=} \frac{1}{4n}\sum_{i=1}^n\sum_{j=1}^n (x_j - x_i)^2 = \frac{1}{2n}\sum_{(i,j)} (x_j - x_i)^2 \\ &\overset{(11)}{\leq} \frac{\beta(\mathcal{G})}{2n} \sum_{e=(i,j)\in\mathcal{E}} (x_i - x_j)^2 \overset{\text{Lemma 4}}{=} \frac{1}{2\alpha(\mathcal{G})} \sum_{e=(i,j)\in\mathcal{E}} (x_i - x_j)^2 \end{aligned}$$

□

B.3 Proof of Theorem 5

Having established Lemmas 15 and 16, we can now proceed with the proof of Theorem 5:

$$\begin{aligned} \mathbb{E}\left[D(y^*) - D(y^{t+1}) \mid y^t\right] &= D(y^*) - D(y^t) - \mathbb{E}\left[D(y^{t+1}) - D(y^t) \mid y^t\right] \\ &\overset{(22)}{=} D(y^*) - D(y^t) - \sum_{e=(i,j)\in\mathcal{E}} \frac{1}{4m}(x_i^t - x_j^t)^2 \\ &\overset{(1)}{=} \frac{1}{2}\|\bar{c}\mathbf{1} - x^t\|^2 - \sum_{e=(i,j)\in\mathcal{E}} \frac{1}{4m}(x_i^t - x_j^t)^2 \\ &\overset{(23)}{\leq} \left(1 - \frac{\alpha(\mathcal{G})}{2m}\right)\frac{1}{2}\|\bar{c}\mathbf{1} - x^t\|^2 \\ &\overset{(1)}{=} \left(1 - \frac{\alpha(\mathcal{G})}{2m}\right)\left(D(y^*) - D(y^t)\right) . \end{aligned}$$

Taking expectation again, we get the recursion

$$\mathbb{E}\left[D(y^*) - D(y^{t+1})\right] \leq \left(1 - \frac{\alpha(\mathcal{G})}{2m}\right)\mathbb{E}\left[D(y^*) - D(y^t)\right].$$

C Proofs for Section 3

C.1 Proof of Theorem 6

Lemma 17. *Fix $k \geq 0$ and let $R > 0$. Then*

$$\min_{\lambda=(\lambda^0,\ldots,\lambda^k)\in\mathbb{R}^{k+1}} \frac{R + \beta^k}{\alpha^k} = 2\sqrt{\frac{R}{k+1}},$$

and the optimal solution is given by $\lambda^t = \sqrt{\frac{R}{k+1}}$ for all t.

Proof. Define $\phi(\lambda) = \frac{R+\beta^k}{\alpha^k}$. If we write $\lambda = rx$, where $r = \|\lambda\|$ and x is of unit norm, then $\phi(tx) = \frac{R+r^2}{r\langle \mathbf{1}, x\rangle}$. Clearly, for any fixed r, the $x \in \mathbb{R}^{k+1}$ minimizing $x \mapsto \phi(rx)$ is $x = \mathbf{1}/\|\mathbf{1}\|$, where $\mathbf{1}$ is the vector of ones in $\mathbb{R}^{k+1}$. It now only remains to minimize the function $r \mapsto \frac{R+r^2}{r\|\mathbf{1}\|}$. This function is convex and differentiable. Setting the derivative to zero leads to $r = \sqrt{R}$. Combining the above, we get the optimal solution $\lambda = \frac{r}{\|\mathbf{1}\|}\mathbf{1} = \frac{\sqrt{R}}{\|\mathbf{1}\|}\mathbf{1}$. □

Let $e = (i, j)$ be the edge selected at iteration $t \geq 0$. Applying Lemma 3, we see that $D(y^{t+1}) - D(y^t) = \lambda^t |x_i^t - x_j^t| - \left(\lambda^t\right)^2$. Taking expectation with respect to edge selection, we get

$$\mathbb{E}\left[D(y^{t+1}) - D(y^t) \mid y^t\right] = -\left(\lambda^t\right)^2 + \lambda^t \cdot \frac{1}{m} \sum_{e=(i,j)\in\mathcal{E}} |x_i^t - x_j^t|,$$

and taking expectation again and using the tower property, we get the identity $\mathbb{E}\left[D(y^{t+1}) - D(y^t)\right] = -\left(\lambda^t\right)^2 + \lambda^t \cdot \mathbb{E}\left[L^t\right]$. Therefore,

$$\begin{aligned} D(y^*) - D(y^0) &\geq \mathbb{E}\left[D(y^{k+1}) - D(y^0)\right] \\ &= \mathbb{E}\left[\sum_{t=0}^{k} D(y^{t+1}) - D(y^t)\right] \\ &= \sum_{t=0}^{k} \mathbb{E}\left[D(y^{t+1}) - D(y^t)\right] \quad = \quad -\sum_{t=0}^{k}\left(\lambda^t\right)^2 + \sum_{t=0}^{k} \lambda^t \cdot \mathbb{E}\left[L^t\right]. \end{aligned}$$

It remains to reshuffle the resulting inequality to obtain (14).

We can see that part (i) follows directly. Optimality of stepsizes in (ii) is due to Lemma 17. To show (iii) we should state that

$$\alpha^k = \sum_{t=0}^{k} \lambda^k = \sum_{1}^{k+1} \frac{a}{\sqrt{t}} \geq a \int_{t=1}^{k+2} t^{-1/2} dt = 2a\left(\sqrt{k+2} - 1\right)$$

$$\beta^k = \sum_{t=0}^{k} \left(\lambda^k\right)^2 = \sum_{t=1}^{k+1} \frac{a^2}{t} \leq a^2 \int_{1/2}^{k+3/2} t^{-1} dt = a^2\left(\log(k+3/2) + \log(2)\right)$$

The inequality above holds due to the fact that for $t > 1/2$ we have $t^{-1} \leq \int_{t-1/2}^{t+1/2} x^{-1} dx$ since x^{-1} is convex function.

C.2 Proof of Theorem 7

Using Lemma 3 with we have

$$\begin{aligned} \mathbb{E}\left[D(y^{t+1}) - D(y^t) \mid y^t\right] &= -\left(\lambda^t\right)^2 + \lambda^t \frac{1}{m} \sum_{e\in\mathcal{E}} |x_i^t - x_j^t| \\ &= \frac{1}{4m^2}\left(\sum_{e\in\mathcal{E}} |x_i^t - x_j^t|\right)^2 \quad \geq \quad \frac{1}{4m^2} \sum_{e\in\mathcal{E}} \left(x_i^t - x_j^t\right)^2. \end{aligned}$$

Taking the expectation again we obtain

$$\mathbb{E}\left[D(y^{t+1}) - D(y^t)\right] \geq \frac{1}{4m^2}\mathbb{E}\left[\sum_{e\in\mathcal{E}} \left(x_i^t - x_j^t\right)^2\right]. \tag{24}$$

On the other hand, we have

$$\begin{aligned} D(y^{t+1}) - D(y^t) &= \big(D(y^{t+1}) - D(y^*)\big) + \big(D(y^*) - D(y^t)\big) \\ &= \frac{1}{2}\|\bar{c}\mathbf{1} - x^t\|^2 - \frac{1}{2}\|\bar{c}\mathbf{1} - x^{t+1}\|^2 \\ &= \frac{\alpha(\mathcal{G})}{4m^2}\|\bar{c}\mathbf{1} - x^t\|^2 + \left(1 - \frac{\alpha(\mathcal{G})}{2m^2}\right)\frac{1}{2}\|\bar{c}\mathbf{1} - x^t\|^2 - \frac{1}{2}\|\bar{c}\mathbf{1} - x^{t+1}\|^2 \\ &\overset{(23)}{\leq} \frac{1}{4m^2}\sum_{e=(i,j)\in\mathcal{E}}\left(x_i^t - x_j^t\right)^2 + \left(1 - \frac{\alpha(\mathcal{G})}{2m^2}\right)\frac{1}{2}\|\bar{c}\mathbf{1} - x^t\|^2 - \frac{1}{2}\|\bar{c}\mathbf{1} - x^{t+1}\|^2. \end{aligned}$$

Taking the expectation of the above and combining with (24) we obtain the desired recursion

$$\mathbb{E}\left[\|\bar{c}\mathbf{1} - x^{t+1}\|^2\right] \quad \leq \quad \left(1 - \frac{\alpha(\mathcal{G})}{2m^2}\right)\mathbb{E}\left[\|\bar{c}\mathbf{1} - x^t\|^2\right].$$

C.3 Proof of Lemma 8

Let $e = (i, j)$ be the edge selected at iteration t. Applying Lemma 3, we see that

$$D(y^{t+1}) - D(y^t) = \begin{cases} -\frac{\epsilon}{2}(x_i^t - x_j^t) - \frac{\epsilon^2}{4}, & x_i^t - x_j^t \leq -\epsilon \\ \frac{\epsilon}{2}(x_i^t - x_j^t) - \frac{\epsilon^2}{4}, & x_j^t - x_i^t \leq -\epsilon, \\ 0, & \text{otherwise.} \end{cases}$$

This implies that

$$D(y^{t+1}) - D(y^t) \begin{cases} \geq \frac{\epsilon^2}{4}, & \text{if } \Delta_e^t = 1, \\ = 0, & \text{if } \Delta_e^t = 0. \end{cases}$$

Taking expectation in the selection of e, we get

$$\mathbb{E}\left[D(y^{t+1}) - D(y^t) \mid y^t\right] \geq \frac{\epsilon^2}{4} \cdot \mathbb{P}(\Delta_e^t = 1 \mid y^t) + 0 \cdot \mathbb{P}(\Delta_e^t = 0 \mid y^t) = \frac{\epsilon^2}{4}\Delta^t.$$

It remains to take expectation again.

C.4 Proof of Theorem 9

Since for all $k \geq 0$ we have $D(y^k) \leq D(y^*)$, it follows that

$$D(y^*) - D(y^0) \geq \mathbb{E}\left[D(y^k) - D(y^0)\right] = \mathbb{E}\left[\sum_{t=0}^{k-1} D(y^{t+1}) - D(y^t)\right] = \sum_{t=0}^{k-1}\mathbb{E}\left[D(y^{t+1}) - D(y^t)\right].$$

It remains to apply Lemma 8.

C.5 Proof of Lemma 10

Firstly we will compute increase in the dual function value in iteration t:

$$\begin{aligned} D(y^{t+1}) - D(y^t) &= \frac{1}{2}\|\bar{c}\mathbf{1} - x^t\|^2 - \frac{1}{2}\|\bar{c}\mathbf{1} - x^{t+1}\|^2 \\ &= \frac{1}{2}\left((\bar{c} - x_j^t)^2 + (\bar{c} - x_i^t)^2 - (\bar{c} - x_j^{t+1})^2 - (\bar{c} - x_i^{t+1})^2\right) \\ &= -\bar{c}\left(x_j^{t+1} + x_i^{t+1} - x_j^t - x_i^t\right) + \frac{1}{2}\left(\left(x_j^t\right)^2 + \left(x_i^t\right)^2 - \left(x_j^{t+1}\right)^2 - \left(x_i^{t+1}\right)^2\right) \\ &= -\bar{c}\left(w_j^{t_j} + w_i^{t_i}\right) + \frac{1}{2}\left(\left(x_j^t\right)^2 + \left(x_i^t\right)^2\right) \\ &\qquad -\frac{1}{4}\left(\left(x_j^t + x_i^t\right)^2 + 2\left(x_j^t + x_i^t\right)\left(w_j^{t_j} + w_i^{t_i}\right) + \left(w_j^{t_j} + w_i^{t_i}\right)^2\right) \\ &= \frac{1}{4}\left(x_j^t - x_i^t\right)^2 - \left(w_i^{t_i} + w_j^{t_j}\right)\left(\bar{c} + \frac{1}{2}\left(x_j^t + x_i^t\right)\right) - \frac{1}{4}\left(w_i^{t_i} + w_j^{t_j}\right)^2. \end{aligned} \tag{25}$$

Now we want to estimate the expectation of this gap. Our main goal is to find $\mathbb{E}\left[D(y^{t+1}) - D(y^t)\right]$. There are 3 terms in (25). Since expectation is linear, we will evaluate expectations of the 3 terms separately and merge them at the end.

Taking the expectation over the choice of edge and inserted noise in iteration t we obtain

$$\mathbb{E}\left[\frac{1}{4}\left(x_i^t - x_j^t\right)^2 | x^t\right] = \frac{1}{4m}\sum_{e\in\mathcal{E}}\left(x_i^t - x_j^t\right)^2. \tag{26}$$

Thus we have

$$\begin{aligned}
&\mathbb{E}\left[D(y^{t+1}) - D(y^t) - \frac{1}{4}\left(x_i^t - x_j^t\right)^2 \mid x^t\right] \\
&\overset{(26)}{=} \mathbb{E}\left[D(y^{t+1}) - D(y^t) \mid x^t\right] - \frac{1}{4m}\sum_{e\in\mathcal{E}}\left(x_i^t - x_j^t\right)^2 \\
&= \mathbb{E}\left[D(y^*) - D(y^t) \mid x^t\right] + \mathbb{E}\left[D(y^{t+1}) - D(y^*) \mid x^t\right] - \frac{1}{4m}\sum_{e\in\mathcal{E}}\left(x_i^t - x_j^t\right)^2 \\
&\overset{(1)}{=} \frac{1}{2}\|\bar{c}\mathbf{1} - x^t\|^2 - \mathbb{E}\left[\frac{1}{2}\|\bar{c}\mathbf{1} - x^{t+1}\|^2 \mid x^t\right] - \frac{1}{4m}\sum_{e=(i,j)\in\mathcal{E}}\left(x_i^t - x_j^t\right)^2 \\
&\overset{(23)}{\leq} \frac{1}{2}\|\bar{c}\mathbf{1} - x^t\|^2 - \mathbb{E}\left[\frac{1}{2}\|\bar{c}\mathbf{1} - x^{t+1}\|^2 \mid x^t\right] - \frac{\alpha(\mathcal{G})}{4m}\|\bar{c}\mathbf{1} - x^t\|^2 \\
&= \left(1 - \frac{\alpha(\mathcal{G})}{2m}\right)\frac{1}{2}\|\bar{c}\mathbf{1} - x^t\|^2 - \mathbb{E}\left[\frac{1}{2}\|\bar{c}\mathbf{1} - x^{t+1}\|^2 \mid x^t\right] \\
&\overset{(1)}{=} \left(1 - \frac{\alpha(\mathcal{G})}{2m}\right)\left(D(y^*) - D(y^t)\right) - \mathbb{E}\left[D(y^*) - D(y^{t+1}) \mid y^t\right].
\end{aligned}$$

Taking the full expectation of the above and using tower property, we get

$$\mathbb{E}\left[D(y^{t+1}) - D(y^t) - \frac{1}{4}\left(x_i^t - x_j^t\right)^2\right] \leq \left(1 - \frac{\alpha(\mathcal{G})}{2m}\right)\mathbb{E}\left[D(y^*) - D(y^t)\right] - \mathbb{E}\left[D(y^*) - D(y^{t+1})\right]. \tag{27}$$

Lemma 18. *Suppose that we run Algorithm 4 for t iterations and t_i denotes the number of times that some edge corresponding to node i was selected during the algorithm.*

1. *$v_i^{t_i}$ and t_j are independent for all (i.e., not necessarily distinct) i, j.*
2. *$v_i^{t_i}$ and $\phi_j^{t_j}$ are independent for all (i.e., not necessarily distinct) i, j.*
3. *$w_i^{t_i}$ and $w_j^{t_j}$ have zero correlation for all $i \neq j$.*
4. *x_j^t and $\phi_i^{t_i} v_i^{t_i}$ have zero correlation for all (i.e., not necessarily distinct) i, j.*

Proof. 1. Follows from the definition of v_i^t.

2. Follows from the definition of v_i^t.

3. Note that we have $w_i^{t_i} = \phi_i^{t_i} v_i^{t_i} - \phi_i^{t_i-1} v_i^{t_i-1}$ and $w_j^{t_j} = \phi_j^{t_j} v_j^{t_j} - \phi_j^{t_j-1} v_j^{t_j-1}$. Clearly, $v_i^{t_i}$ and $w_j^{t_j}$ have zero correlation. Similarly $v_i^{t_i-1}$ and $w_j^{t_j}$ have zero correlation. Thus, $w_i^{t_i}$ and $w_j^{t_j}$ have zero correlation.

4. Clearly, x_j^t is a function initial state and all instances of random variables up to the iteration t. Thus, $v_i^{t_i}$ is independent to x_j^t from the definition. Thus, x_j^t and $\phi_i^{t_i} v_i^{t_i}$ have zero correlation.

□

Now we are going to take the expectation of the second term of (25). We will use the "tower rule" of expectations in the form $\mathbb{E}\left[\mathbb{E}\left[\mathbb{E}\left[X \mid Y, Z\right] \mid Y\right]\right] = \mathbb{E}\left[X\right]$, where X, Y, Z are random variables. In particular, we get

$$\mathbb{E}\left[-\left(w_i^{t_i} + w_j^{t_j}\right)\left(\overline{c} + \frac{1}{2}\left(x_j^t + x_i^t\right)\right)\right] = \mathbb{E}\left[\mathbb{E}\left[\mathbb{E}\left[-\left(w_i^{t_i} + w_j^{t_j}\right)\left(\overline{c} + \frac{1}{2}\left(x_j^t + x_i^t\right)\right) \mid e^t, x^t\right] \mid x^t\right]\right].$$

In the equation above, e^t denotes an edge selected at in the iteration t. Let us first calculate the inner most expectation on the right hand side of the above identity:

$$\begin{aligned}
&\mathbb{E}\left[-\left(w_i^{t_i} + w_j^{t_j}\right)\left(\overline{c} + \frac{1}{2}\left(x_j^t + x_i^t\right)\right) \mid e^t, x^t\right] \\
&\overset{(*^1)}{=} \mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1} + \phi_j^{t_j-1}v_j^{t_j-1} - \phi_i^{t_i}v_i^{t_i} - \phi_j^{t_j}v_j^{t_j}\right)\left(\overline{c} + \frac{1}{2}\left(x_j^t + x_i^t\right)\right) \mid e^t, x^t\right] \\
&\overset{(*^2)}{=} \mathbb{E}\left[\overbrace{\left(\phi_i^{t_i-1}v_i^{t_i-1} + \phi_j^{t_j-1}v_j^{t_j-1}\right)}^{\text{constant}}\overbrace{\left(\overline{c} + \frac{1}{2}\left(x_j^t + x_i^t\right)\right)}^{\text{constant}} \mid e^t, x^t\right] \\
&\qquad + \mathbb{E}\left[\left(-\phi_i^{t_i}v_i^{t_i} - \phi_j^{t_j}v_j^{t_j}\right)\overbrace{\left(\overline{c} + \frac{1}{2}\left(x_j^t + x_i^t\right)\right)}^{\text{constant}} \mid e^t, x^t\right] \\
&= \left(\phi_i^{t_i-1}v_i^{t_i-1} + \phi_j^{t_j-1}v_j^{t_j-1}\right)\left(\overline{c} + \frac{1}{2}\left(x_j^t + x_i^t\right)\right) \\
&\qquad + \mathbb{E}\left[\left(-\phi_i^{t_i}v_i^{t_i} - \phi_j^{t_j}v_j^{t_j}\right)|e, x^t\right]\left(\overline{c} + \frac{1}{2}\left(x_j^t + x_i^t\right)\right) \to 0 \\
&\overset{L.18}{=} \left(\phi_i^{t_i-1}v_i^{t_i-1} + \phi_j^{t_j-1}v_j^{t_j-1}\right)\left(\overline{c} + \frac{1}{2}\left(x_j^t + x_i^t\right)\right) \\
&= \left(\phi_i^{t_i-1}v_i^{t_i-1} + \phi_j^{t_j-1}v_j^{t_j-1}\right)\overline{c} + \frac{1}{2}\left(\phi_i^{t_i-1}v_i^{t_i-1} + \phi_j^{t_j-1}v_j^{t_j-1}\right)\left(x_j^t + x_i^t\right),
\end{aligned}$$

where $(*^1)$ means definition of $w_i^{t_i}$ and $(*^2)$ means linearity of expectation.

Now we take the expectation of last expression above with respect to choice of edge on t-th iteration. We obtain

$$\begin{aligned}
&\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1} + \phi_j^{t_j-1}v_j^{t_j-1}\right)\overline{c} + \frac{1}{2}\left(\phi_i^{t_i-1}v_i^{t_i-1} + \phi_j^{t_j-1}v_j^{t_j-1}\right)\left(x_j^t + x_i^t\right)|x^t\right] \\
&\overset{(*^2)}{=} \frac{1}{2}\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1} + \phi_j^{t_j-1}v_j^{t_j-1}\right)\left(x_j^t + x_i^t\right)|x^t\right] + \mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1} + \phi_j^{t_j-1}v_j^{t_j-1}\right)\overline{c}|x^t\right] \to 0 \\
&\overset{L.18}{=} \frac{1}{2}\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1} + \phi_j^{t_j-1}v_j^{t_j-1}\right)\left(x_j^t + x_i^t\right)|x^t\right] \\
&\overset{(*^3)}{=} \frac{1}{2m}\sum_{e\in\mathcal{E}}\left(\phi_i^{t_i-1}v_i^{t_i-1} + \phi_j^{t_j-1}v_j^{t_j-1}\right)\left(x_j^t + x_i^t\right) \\
&= \frac{1}{2m}\sum_{e\in\mathcal{E}}\left(\phi_i^{t_i-1}v_i^{t_i-1}x_i^t + \phi_j^{t_j-1}v_j^{t_j-1}x_j^t\right) + \frac{1}{2m}\sum_{e\in\mathcal{E}}\left(\phi_i^{t_i-1}v_i^{t_i-1}x_j^t + \phi_j^{t_j-1}v_j^{t_j-1}x_i^t\right) \\
&\overset{(*^4)}{=} \frac{1}{2m}\sum_{i=1}^{n} d_i\phi_i^{t_i-1}v_i^{t_i-1}x_i^t + \frac{1}{2m}\sum_{e\in\mathcal{E}}\left(\phi_i^{t_i-1}v_i^{t_i-1}x_j^t + \phi_j^{t_j-1}v_j^{t_j-1}x_i^t\right),
\end{aligned}$$

where $(*^3)$ means definition of expectation and $(*^4)$ means change of the summation order.

Lemma 19.

$$\mathbb{E}\left[\phi_i^{t_i-1}v_i^{t_i-1}x_i^t\right] = \frac{1}{2}\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}\right)^2\right]. \tag{28}$$

Proof.

$$\begin{aligned}
&\mathbb{E}\left[\phi_i^{t_i-1}v_i^{t_i-1}x_i^t\right] \\
&\overset{(*^5)}{=} \mathbb{E}\left[\phi_i^{t_i-1}v_i^{t_i-1}\left(\left(x_i^t - \frac{\phi_i^{t_i-1}v_i^{t_i-1}}{2}\right) + \frac{\phi_i^{t_i-1}v_i^{t_i-1}}{2}\right)\right] \\
&\overset{(*^2)}{=} \mathbb{E}\left[\phi_i^{t_i-1}v_i^{t_i-1}\left(x_i^t - \frac{\phi_i^{t_i-1}v_i^{t_i-1}}{2}\right)\right] + \frac{1}{2}\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}\right)^2\right] \\
&\overset{(*^6)}{=} \mathbb{E}\left[\phi_i^{t_i-1}v_i^{t_i-1}\left(\frac{x_i^{t_i-1} + x_l^{t_l^0} + w^{t_i-1} + w^{t_l^0} - \phi_i^{t_i-1}v_i^{t_i-1}}{2}\right)\right] + \frac{1}{2}\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}\right)^2\right] \\
&\overset{(*^1)}{=} \mathbb{E}\left[\phi_i^{t_i-1}v_i^{t_i-1}\left(\frac{x_i^{t_i-1} + x_l^{t_l^0} + \phi_i^{t_i-1}v_i^{t_i-1} - \phi_i^{t_i-2}v_i^{t_i-2} + \phi_i^{t_l^0}v_l^{t_l^0} - \phi_i^{t_l^0-1}v_l^{t_l^0-1} - \phi_i^{t_i-1}v_i^{t_i-1}}{2}\right)\right] \\
&\qquad + \frac{1}{2}\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}\right)^2\right] \\
&= \mathbb{E}\left[\phi_i^{t_i-1}v_i^{t_i-1}\left(\frac{x_i^{t_i-1} + x_l^{t_l^0} + \phi_i^{t_l^0}v_l^{t_l^0} - \phi_i^{t_l^0-1}v_l^{t_l^0-1} - \phi_i^{t_i-2}v_i^{t_i-2}}{2}\right)\right] + \frac{1}{2}\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}\right)^2\right] \\
&\overset{L.18}{=} \cancel{\mathbb{E}\left[\phi_i^{t_i-1}v_i^{t_i-1}\right]}\,0\,\mathbb{E}\left[\left(\frac{x_i^{t_i-1} + x_l^{t_l^0} + \phi_i^{t_l^0}v_l^{t_l^0} - \phi_i^{t_l^0-1}v_l^{t_l^0-1} - \phi_i^{t_i-2}v_i^{t_i-2}}{2}\right)\right] \\
&\qquad + \frac{1}{2}\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}\right)^2\right] \\
&= \frac{1}{2}\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}\right)^2\right],
\end{aligned}$$

where in step $(*^5)$ we add and subtracting $\frac{\phi_i^{t_i-1}v_i^{t_i-1}}{2}$. In the step $(*^6)$ we denote by l a node such that that the noise $\phi_i^{t_i-1}v_i^{t_i-1}$ was added to the system when the edge (i,l) was chosen (we do not consider $t_i = 0$ since in this case the Lemma 19 trivially holds).

□

Taking the expectation with respect to the algorithm we obtain

$$\begin{aligned}
&\mathbb{E}\left[-\left(w_i^{t_i} + w_j^{t_j}\right)\left(\overline{c} + \frac{1}{2}\left(x_j^t + x_i^t\right)\right)\right] \\
&\overset{(*^7)}{=} \mathbb{E}\left[\mathbb{E}\left[\mathbb{E}\left[-\left(w_i^{t_i} + w_j^{t_j}\right)\left(\overline{c} + \frac{1}{2}\left(x_j^t + x_i^t\right)\right)|x^t, e\right]|x^t\right]\right] \\
&= \mathbb{E}\left[\frac{1}{2m}\sum_{i=1}^n d_i\phi_i^{t_i-1}v_i^{t_i-1}x_i^t + \frac{1}{2m}\sum_{e\in\mathcal{E}}\left(\phi_i^{t_i-1}v_i^{t_i-1}x_j^t + \phi_j^{t_j-1}v_j^{t_j-1}x_i^t\right)\right] \\
&\overset{(*^2)}{=} \frac{1}{2m}\sum_{i=1}^n d_i\mathbb{E}\left[\phi_i^{t_i-1}v_i^{t_i-1}x_i^t\right] + \frac{1}{2m}\sum_{e\in\mathcal{E}}\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}x_j^t + \phi_j^{t_j-1}v_j^{t_j-1}x_i^t\right)\right] \\
&\overset{L.19}{=} \frac{1}{4m}\sum_{i=1}^n d_i\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}\right)^2\right] + \frac{1}{2m}\sum_{e\in\mathcal{E}}\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}x_j^t + \phi_j^{t_j-1}v_j^{t_j-1}x_i^t\right)\right],
\end{aligned} \tag{29}$$

where $(*^7)$ means tower rule.

Lemma 20.

$$\mathbb{E}\left[\left(\phi_i^{t_i} v_i^{t_i} + \phi_j^{t_j} v_j^{t_j}\right)^2 |x^t, e^t\right] = \sigma_i^2 \phi_i^{2t_i} + \sigma_j^2 \phi_j^{2t_j}. \tag{30}$$

Proof. Since we have $\mathbb{E}\left[\left(\phi_i^{t_i} v_i^{t_i} + \phi_j^{t_j} v_j^{t_j}\right) |x^t, e^t\right] = 0$, and also for any random variable X: $\mathbb{E}\left[X^2\right] = \mathbb{V}\left(X\right) + \mathbb{E}\left[X\right]^2$, we only need to compute the variance:

$$\mathbb{V}\left(\phi_i^{t_i} v_i^{t_i} + \phi_j^{t_j} v_j^{t_j}\right) = \mathbb{V}\left(\phi_i^{t_i} v_i^{t_i}\right) + \mathbb{V}\left(\phi_j^{t_j} v_j^{t_j}\right) = \left(\phi_i^{t_i}\right)^2 \mathbb{V}\left(v_i^{t_i}\right) + \left(\phi_j^{t_j}\right)^2 \mathbb{V}\left(v_j^{t_j}\right).$$

□

Taking an expectation of the third term of (25) with respect to lastly added noise, the expression $\mathbb{E}\left[\left(w_i^{t_i} + w_j^{t_j}\right)^2 |x^t, e^t\right]$ is equal to

$$\begin{aligned}
&\overset{(*^1)}{=} \mathbb{E}\left[\left(\phi_i^{t_i} v_i^{t_i} + \phi_j^{t_j} v_j^{t_j} - \phi_i^{t_i-1} v_i^{t_i-1} - \phi_j^{t_j-1} v_j^{t_j-1}\right)^2 |x^t, e^t\right] \\
&\overset{(*^2)}{=} \mathbb{E}\left[\left(\phi_i^{t_i} v_i^{t_i} + \phi_j^{t_j} v_j^{t_j}\right)^2 |x^t, e^t\right] - 2\mathbb{E}\left[\overbrace{\left(\phi_i^{t_i} v_i^{t_i} + \phi_j^{t_j} v_j^{t_j}\right)}^{\mathbb{E}\left[\phi_i^{t_i} v_i^{t_i} + \phi_j^{t_j} v_j^{t_j}\right]=0} \overbrace{\left(\phi_i^{t_i-1} v_i^{t_i-1} + \phi_j^{t_j-1} v_j^{t_j-1}\right)}^{\text{constant}} |x^t, e^t\right] \\
&\quad + \overbrace{\mathbb{E}\left[\left(\phi_i^{t_i-1} v_i^{t_i-1} + \phi_j^{t_j-1} v_j^{t_j-1}\right)^2 |x^t, e^t\right]}^{\text{constant}} \\
&\overset{(30)}{=} \sigma_i^2 \phi_i^{2t_i} + \sigma_j^2 \phi_j^{2t_j} + \left(\phi_i^{t_i-1} v_i^{t_i-1} + \phi_j^{t_j-1} v_j^{t_j-1}\right)^2.
\end{aligned}$$

Taking the expectation over e^t we obtain:

$$\begin{aligned}
\mathbb{E}\left[\left(w_i^{t_i} + w_j^{t_j}\right)^2 |x^t\right] &\overset{(*^7)}{=} \mathbb{E}\left[\mathbb{E}\left[\left(w_i^{t_i} + w_j^{t_j}\right)^2 |e^t, x^t\right] |x^t\right] \\
&= \mathbb{E}\left[\sigma_i^2 \phi_i^{2t_i} + \sigma_j^2 \phi_j^{2t_j} + \left(\phi_i^{t_i-1} v_i^{t_i-1} + \phi_j^{t_j-1} v_j^{t_j-1}\right)^2 |x^t\right] \\
&\overset{(*^3)}{=} \frac{1}{m} \sum_{e \in \mathcal{E}} \left(\sigma_i^2 \phi_i^{2t_i} + \sigma_j^2 \phi_j^{2t_j}\right) + \frac{1}{m} \sum_{e \in \mathcal{E}} \left(\phi_i^{t_i-1} v_i^{t_i-1} + \phi_j^{t_j-1} v_j^{t_j-1}\right)^2 \\
&\overset{(*^4)}{=} \frac{1}{m} \sum_{i=1}^{n} d_i \sigma_i^2 \phi_i^{2t_i} + \frac{1}{m} \sum_{e \in \mathcal{E}} \left(\phi_i^{t_i-1} v_i^{t_i-1} + \phi_j^{t_j-1} v_j^{t_j-1}\right)^2.
\end{aligned}$$

Finally, taking the expectation with respect to the algorithm we get

$$
\begin{aligned}
&\mathbb{E}\left[\left(w_i^{t_i}+w_j^{t_j}\right)^2\right] \\
&\quad\overset{(*^7)}{=}\mathbb{E}\left[\mathbb{E}\left[\left(w_i^{t_i}+w_j^{t_j}\right)^2|x^t\right]\right] \\
&\quad\overset{(*^2)}{=}\frac{1}{m}\sum_{i=1}^n d_i\sigma_i^2\mathbb{E}\left[\phi_i^{2t_i}\right]+\frac{1}{m}\sum_{e\in\mathcal{E}}\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}+\phi_j^{t_j-1}v_j^{t_j-1}\right)^2\right] \\
&\quad=\frac{1}{m}\sum_{i=1}^n d_i\sigma_i^2\mathbb{E}\left[\phi_i^{2t_i}\right]+\frac{1}{m}\sum_{e\in\mathcal{E}}\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}\right)^2+\left(\phi_j^{t_j-1}v_j^{t_j-1}\right)^2+2\phi_i^{t_i-1}v_i^{t_i-1}\phi_j^{t_j-1}v_j^{t_j-1}\right] \\
&\quad\overset{(*^4)}{=}\frac{1}{m}\sum_{i=1}^n d_i\sigma_i^2\mathbb{E}\left[\phi_i^{2t_i}\right]+\frac{1}{m}\sum_{i=1}^n d_i\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}\right)^2\right]+\frac{2}{m}\sum_{e\in\mathcal{E}}\mathbb{E}\left[\phi_i^{t_i-1}v_i^{t_i-1}\phi_j^{t_j-1}v_j^{t_j-1}\right] \\
&\quad\overset{L.18}{=}\frac{1}{m}\sum_{i=1}^n d_i\sigma_i^2\mathbb{E}\left[\phi_i^{2t_i}\right]+\frac{1}{m}\sum_{i=1}^n d_i\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}\right)^2\right]+\frac{2}{m}\sum_{e\in\mathcal{E}}\cancelto{0}{\mathbb{E}\left[\phi_i^{t_i-1}v_i^{t_i-1}\right]}\cancelto{0}{\mathbb{E}\left[\phi_j^{t_j-1}v_j^{t_j-1}\right]} \\
&\quad=\frac{1}{m}\sum_{i=1}^n d_i\sigma_i^2\mathbb{E}\left[\phi_i^{2t_i}\right]+\frac{1}{m}\sum_{i=1}^n d_i\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}\right)^2\right]. \qquad (31)
\end{aligned}
$$

Combining (25) with (27), (29) and (31) we obtain

$$
\begin{aligned}
\mathbb{E}\left[D(y^*)-D(y^{t+1})\right] &\leq \left(1-\frac{\alpha(\mathcal{G})}{2m}\right)\mathbb{E}\left[D(y^*)-D(y^t)\right]-\frac{1}{4m}\sum_{i=1}^n d_i\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}\right)^2\right] \\
&\quad-\frac{1}{2m}\sum_{e\in\mathcal{E}}\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}x_j^t+\phi_j^{t_j-1}v_j^{t_j-1}x_i^t\right)\right] \\
&\quad+\frac{1}{4m}\sum_{i=1}^n d_i\sigma_i^2\mathbb{E}\left[\phi_i^{2t_i}\right]+\frac{1}{4m}\sum_{i=1}^n d_i\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}\right)^2\right] \\
&= \left(1-\frac{\alpha(\mathcal{G})}{2m}\right)\mathbb{E}\left[D(y^*)-D(y^t)\right]+\frac{1}{4m}\sum_{i=1}^n d_i\sigma_i^2\mathbb{E}\left[\phi_i^{2t_i}\right] \\
&\quad-\frac{1}{2m}\sum_{e\in\mathcal{E}}\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}x_j^t+\phi_j^{t_j-1}v_j^{t_j-1}x_i^t\right)\right],
\end{aligned}
$$

which concludes the proof.

C.6 Proof of Theorem 11

Lemma 21. *After t iterations of algorithm 4 we have*

$$
\mathbb{E}\left[\phi_i^{2t_i}\right]=\left(1-\frac{d_i}{m}\left(1-\phi_i^2\right)\right)^t. \qquad (32)
$$

Proof.

$$
\begin{aligned}
\mathbb{E}\left[\phi_i^{2t_i}\right] &= \sum_{j=0}^t\mathbb{P}(t_i=j)\phi_i^{2j} = \sum_{j=0}^t\binom{t}{j}\left(\frac{m-d_i}{m}\right)^{t-j}\left(\frac{d_i}{m}\phi_i^2\right)^j \\
&= \left(\frac{m-d_i}{m}+\frac{d_i}{m}\phi_i^2\right)^t = \left(1-\frac{d_i}{m}\left(1-\phi_i^2\right)\right)^t.
\end{aligned}
$$

□

Lemma 22. *Random variables $\phi_i^{t_i-1}v_i^{t_i-1}$ and x_j^t are nonegatively correlated, i.e.*

$$\mathbb{E}\left[\phi_i^{t_i-1}v_i^{t_i-1}x_j^t\right] \geq 0. \tag{33}$$

Proof. Denote $R_{i,j}$ to be a random variable equal to 1 if the noise $w_i^{t_i}$ was added to the system when edge (i,j) was chosen and equal to 0 otherwise. We can rewrite the expectation in the following way:

$$\begin{aligned}\mathbb{E}\left[\phi_i^{t_i-1}v_i^{t_i-1}x_j^t\right] &= \overbrace{\mathbb{E}\left[\phi_i^{t_i-1}v_i^{t_i-1}x_j^t \mid R_{i,j}=1\right]}^{\geq 0}\mathbb{P}(R_{i,j}=1)\\ &\quad + \overbrace{\mathbb{E}\left[\phi_i^{t_i-1}v_i^{t_i-1}x_j^t \mid R_{i,j}=0\right]}^{0}\mathbb{P}(R_{i,j}=0) \quad \geq \quad 0.\end{aligned}$$

The inequality $\mathbb{E}\left[\phi_i^{t_i-1}v_i^{t_i-1}x_j^t \mid R_{i,j}=1\right] \geq 0$ holds due to the fact that $\phi_i^{t_i-1}v_i^{t_i-1}$ was added to x_j with the positive sign. □

Combining (17) with Lemmas 21 and 22 we obtain

$$\begin{aligned}\mathbb{E}\left[D(y^*)-D(y^{t+1})\right] &\leq \left(1-\frac{\alpha(\mathcal{G})}{2m}\right)\mathbb{E}\left[D(y^*)-D(y^t)\right]+\frac{1}{4m}\sum_{i=1}^n d_i\sigma_i^2\mathbb{E}\left[\phi_i^{2t_i}\right]\\ &\quad -\frac{1}{2m}\sum_{e\in\mathcal{E}}\mathbb{E}\left[\left(\phi_i^{t_i-1}v_i^{t_i-1}x_j^t+\phi_i^{t_j-1}v_j^{t_j-1}x_i^t\right)\right]\\ &\overset{(33)}{\leq} \left(1-\frac{\alpha(\mathcal{G})}{2m}\right)\mathbb{E}\left[D(y^*)-D(y^t)\right]+\frac{1}{4m}\sum_{i=1}^n d_i\sigma_i^2\mathbb{E}\left[\phi_i^{2t_i}\right]\\ &\overset{(32)}{=} \left(1-\frac{\alpha(\mathcal{G})}{2m}\right)\mathbb{E}\left[D(y^*)-D(y^t)\right]+\frac{1}{4m}\sum_{i=1}^n d_i\sigma_i^2\left(1-\frac{d_i}{m}\left(1-\phi_i^2\right)\right)^t\\ &= \left(1-\frac{\alpha(\mathcal{G})}{2m}\right)\mathbb{E}\left[D(y^*)-D(y^t)\right]+\frac{\sum\left(d_i\sigma_i^2\right)}{4m}\psi^t.\end{aligned}$$

The recursion above gives us inductively the following

$$\mathbb{E}\left[D(y^*)-D(y^k)\right] \leq \rho^k\left(D(y^*)-D(y^0)\right)+\frac{\sum\left(d_i\sigma_i^2\right)}{4m}\sum_{t=1}^k\rho^{k-t}\psi^t,$$

which concludes the proof of the theorem.

C.7 Proof of Corollary 12

Note that we have

$$\begin{aligned}\psi^t &= \frac{1}{\sum_{i=1}^n d_i\sigma_i^2}\sum_{i=1}^n d_i\sigma_i^2\left(1-\frac{d_i}{m}\left(1-\left(1-\frac{\gamma}{d_i}\right)\right)\right)^t\\ &= \frac{1}{\sum_{i=1}^n d_i\sigma_i^2}\sum_{i=1}^n d_i\sigma_i^2\left(1-\frac{\gamma}{m}\right)^t \quad = \quad \left(1-\frac{\gamma}{m}\right)^t.\end{aligned}$$

In view of Theorem 11, this gives us the following:

$$\begin{aligned}
\mathbb{E}\left[D(y^*) - D(y^k)\right] &\leq \left(D(y^*) - D(y^0)\right)\rho^k + \frac{\sum\left(d_i\sigma_i^2\right)}{4m}\sum_{t=1}^{k}\rho^{k-t}\psi^t \\
&\leq \left(D(y^*) - D(y^0)\right)\rho^k + \frac{\sum\left(d_i\sigma_i^2\right)}{4m}\sum_{t=1}^{k}\rho^{k-t}\left(1-\frac{\gamma}{m}\right)^t \\
&\leq \left(D(y^*) - D(y^0)\right)\rho^k + \frac{\sum\left(d_i\sigma_i^2\right)}{4m}k\max\left(\rho, 1-\frac{\gamma}{m}\right)^k \\
&\leq \left(D(y^*) - D(y^0) + \frac{\sum\left(d_i\sigma_i^2\right)}{4m}k\right)\max\left(\rho, 1-\frac{\gamma}{m}\right)^k.
\end{aligned}$$

D Notation Glossary

The following notational conventions are used throughout the paper. Boldface upper case letters will denote matrices and boldface lower case letters will denote vectors.

Graphs	
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	an undirected graph with vertices $\mathcal{V}$ and edges $\mathcal{E}$
n	$= \lvert\mathcal{V}\rvert$ (number of vertices)
m	$= \lvert\mathcal{E}\rvert$ (number of edges)
$e = (i,j) \in \mathcal{E}$	edge of $\mathcal{G}$ connecting nodes $i, j \in \mathcal{V}$
d_i	degree of node i
$c \in \mathbb{R}^n$	$= (c_1, \ldots, c_n)$; a vector of private values stored at the nodes of $\mathcal{G}$
$\bar{c}$	$\bar{c} = \frac{1}{n}\sum_i c_i$ (the average of the private values)
$\mathbf{A} \in \mathbb{R}^{m\times n}$	
$\mathbf{L} \in \mathbb{R}^{m\times m}$	$= \mathbf{A}\mathbf{A}^\top$ (Laplacian of $\mathcal{G}$)
$\alpha(\mathcal{G})$	$= \lambda_{\min}^{+}(\mathbf{L})$ (algebraic connectivity of $\mathcal{G}$)
$\beta(\mathcal{G})$	$= n/\alpha(\mathcal{G})$
Randomness	
$\mathbb{E}$	expectation
$\mathbb{P}$	probability
$\mathbb{V}$	variance
v_i^k	random variable from $N(0, \sigma_i^2)$
Optimization	
$P : \mathbb{R}^n \to \mathbb{R}$	primal objective function
$D : \mathbb{R}^m \to \mathbb{R}$	dual objective function (a concave quadratic)
$y \in \mathbb{R}^m$	dual variable
$y^* \in \mathbb{R}^m$	optimal dual variable
$x \in \mathbb{R}^n$	primal variable
$x^* \in \mathbb{R}^n$	$= \bar{c}\mathbf{1}$ (optimal primal variable)
$\mathbf{1}$	a vector of all ones in $\mathbb{R}^n$
Summation	
$\sum_i\sum_j$	sum through all ordered pairs of i and j
$\sum_{(i,j)}$	sum through all unordered pairs of i and j
$\sum_{(i,j)\in\mathcal{E}}$	sum through all edges of $\mathcal{G}$

Table 2: The main notation used in the paper.