

Accelerated Gossip via Stochastic Heavy Ball Method

Nicolas Loizou
School of Mathematics,
The University of Edinburgh
Edinburgh, Scotland, UK
n.loizou@sms.ed.ac.uk

Peter Richtárik
KAUST, Saudi Arabia
The University of Edinburgh, United Kingdom
MIPT, Russia
peter.richtarik@kaust.edu.sa

Abstract—In this paper we show how the stochastic heavy ball method (SHB)—a popular method for solving stochastic convex and non-convex optimization problems—operates as a randomized gossip algorithm. In particular, we focus on two special cases of SHB: the Randomized Kaczmarz method with momentum and its block variant. Building upon a recent framework for the design and analysis of randomized gossip algorithms [19] we interpret the distributed nature of the proposed methods. We present novel protocols for solving the average consensus problem where in each step all nodes of the network update their values but only a subset of them exchange their private values. Numerical experiments on popular wireless sensor networks showing the benefits of our protocols are also presented.

Index Terms—Average Consensus Problem, Linear Systems, Networks, Randomized Gossip Algorithms, Randomized Kaczmarz, Momentum, Acceleration

I. INTRODUCTION

Average consensus is a fundamental problem in distributed computing and multi-agent systems. It comes up in many real world applications such as coordination of autonomous agents, estimation, rumour spreading in social networks, PageRank and distributed data fusion on ad-hoc networks and decentralized optimization. Due to its great importance there is much classical [34], [6] and recent [36], [37], [3] work on the design of efficient algorithms/protocols for solving it.

One of the most attractive classes of protocols for solving the average consensus are gossip algorithms. The development and design of gossip algorithms was studied extensively in the last decade. The seminal 2006 paper of Boyd et al. [3] on randomized gossip algorithms motivated a flurry of subsequent research and now gossip algorithms appear in many applications, including distributed data fusion in sensor networks [36], load balancing [5] and clock synchronization [10]. For a survey of selected relevant work prior to 2010, we refer the reader to the work of Dimakis et al. [7]. For more recent results on randomized gossip algorithms we suggest [38], [40], [16], [28], [19], [13]. See also [8], [1], [27].

The main goal in the design of gossip protocols is for the computation and communication to be done as quickly and efficiently as possible. In this work, our focus is precisely this. We design randomized gossip protocols which converge to consensus fast.

A. The average consensus problem

In the average consensus (AC) problem, we are given an undirected connected network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with node set

$\mathcal{V} = \{1, 2, \dots, n\}$ and edges \mathcal{E} . Each node $i \in \mathcal{V}$ “knows” a private value $c_i \in \mathcal{R}$. The goal of AC is for every node of the network to compute the average of these private values, $\bar{c} := \frac{1}{n} \sum_i c_i$, in a distributed fashion. That is, the exchange of information can only occur between connected nodes (neighbors).

B. Main Contributions

In our work we present a new class of randomized gossip protocols where in each iteration all nodes of the network update their values but only a subset of them exchange their private information. The protocols that we present are based on recently proposed ideas for the acceleration of randomized Kaczmarz methods for solving consistent linear systems [21] where the addition of a momentum term was shown to provide practical speedups over the vanilla Kaczmarz methods.

Further, we explain the connection between gossip algorithms for solving the average consensus problem, Kaczmarz-type methods for solving consistent linear systems, and stochastic gradient descent and stochastic heavy ball methods for solving stochastic optimization problems. We show that essentially all these algorithms behave as gossip algorithms when used to solve carefully structure problems.

Finally, we explain in detail the gossip nature of two recently proposed fast Kaczmarz-type methods: the randomized Kaczmarz with momentum (mRK), and its block variant - the randomized block Kaczmarz with momentum (mRBK). We present a detailed comparison of our proposed gossip protocols with existing popular randomized gossip protocols and through numerical experiments we show the benefits of our methods.

C. Structure of the paper

This work is organized as follows. Section II introduces the important technical preliminaries and the necessary background for understanding of our methods. A new connection between gossip algorithms, Kaczmarz methods for solving linear systems and stochastic gradient descent (SGD) for solving stochastic optimization problems is also described. In Section III the two new accelerated gossip protocols are presented. Details of their behaviour and performance are also explained. Numerical evaluation of the new gossip

protocols is presented in Section IV. Finally, concluding remarks are given in Section V.

D. Notation

The following notational conventions are used in this paper. We write $[n] := \{1, 2, \dots, n\}$. Boldface upper-case letters denote matrices; \mathbf{I} is the identity matrix. By \mathcal{L} we denote the solution set of the linear system $\mathbf{A}x = b$, where $\mathbf{A} \in \mathcal{R}^{m \times n}$ and $b \in \mathcal{R}^m$. Throughout the paper, x^* is the projection of x^0 onto \mathcal{L} (that is, x^* is the solution of the best approximation problem; see equation (5)). An explicit formula for the projection of x onto set \mathcal{L} is given by

$$\Pi_{\mathcal{L}}(x) := \arg \min_{x' \in \mathcal{L}} \|x' - x\| = x - \mathbf{A}^\top (\mathbf{A}\mathbf{A}^\top)^\dagger (\mathbf{A}x - b).$$

A matrix that often appears in our update rules is

$$\mathbf{H} := \mathbf{S}(\mathbf{S}^\top \mathbf{A}\mathbf{A}^\top \mathbf{S})^\dagger \mathbf{S}^\top, \quad (1)$$

where $\mathbf{S} \in \mathcal{R}^{m \times q}$ is a random matrix drawn in each step of the proposed methods from a given distribution \mathcal{D} , and \dagger denotes the Moore-Penrose pseudoinverse. Note that \mathbf{H} is a random symmetric positive semi-definite matrix.

In the convergence analysis we use λ_{\min}^+ to indicate the smallest nonzero eigenvalue, and λ_{\max} for the largest eigenvalue of matrix $\mathbf{W} = \mathbb{E}[\mathbf{A}^\top \mathbf{H}\mathbf{A}]$, where the expectation is taken over $\mathbf{S} \sim \mathcal{D}$. Finally, $x^k = (x_1^k, \dots, x_n^k) \in \mathcal{R}^n$ represents the vector with the private values of the n nodes of the network at the k^{th} iteration while with x_i^k we denote the value of node $i \in [n]$ at the k^{th} iteration.

II. BACKGROUND-TECHNICAL PRELIMINARIES

Our work is closely related to two recent papers:

- [19], where a new perspective on randomized gossip algorithms is presented. In particular, a new approach for the design and analysis of randomized gossip algorithms is proposed and it was shown how that Randomized Kaczmarz and Randomized Block Kaczmarz, popular methods for solving linear systems, work as gossip algorithm when applied to a special system encoding the underlying network.
- [21], where several classes of stochastic optimization algorithms enriched with *heavy ball momentum* were analyzed. Among the methods studied are: stochastic gradient descent, stochastic Newton, stochastic proximal point and stochastic dual subspace ascent.

In the rest of this section we present the main results of the above papers, highlighting several connections. These results will be later used for the development of the new randomized gossip protocols.

A. Kaczmarz Methods and Gossip Algorithms

Kaczmarz-type methods are very popular for solving linear systems $\mathbf{A}x = b$ with many equations. The (deterministic) Kaczmarz method for solving consistent linear systems was originally introduced by Kaczmarz in 1937 [14]. Despite the fact that a large volume of papers was written on the topic, the first provably linearly convergent variant of the Kaczmarz method—the randomized Kaczmarz Method

(RK)—was developed more than 70 years later: by in 2009 Strohmer and Vershynin [31]. This result sparked renewed interest in design of randomized methods for solving linear systems [23], [24], [9], [22], [39], [25], [30], [17], [26]. More recently, Gower and Richtárik [11] provide a unified analysis for several randomized iterative methods for solving linear systems using a sketch-and-project framework. We adopt this framework in this paper.

In particular the sketch-and-project algorithm proposed in [11] for solving consistent the linear system $\mathbf{A}x = b$ has the form

$$\begin{aligned} x^{k+1} &= x^k - \mathbf{A}^\top \mathbf{S}_k (\mathbf{S}_k^\top \mathbf{A}\mathbf{A}^\top \mathbf{S}_k)^\dagger \mathbf{S}_k^\top (\mathbf{A}x^k - b) \\ &\stackrel{(1)}{=} x^k - \mathbf{A}^\top \mathbf{H}_k (\mathbf{A}x^k - b), \end{aligned} \quad (2)$$

where in each iteration matrix \mathbf{S}_k is sampled afresh from an arbitrary distribution \mathcal{D} . In [11] it was shown that many popular algorithms for solving linear systems, including RK methods and randomized coordinate descent methods can be cast as special cases of the above update by choosing¹ an appropriate distribution \mathcal{D} . The special cases that we are interested in are the randomized Kaczmarz (RK) and its block variant the randomized block Kaczmarz (RBK).

Let $e_i \in \mathcal{R}^m$ be the i^{th} unit coordinate vector in \mathcal{R}^m and let $\mathbf{I}_{:C}$ be column submatrix of the $m \times m$ identity matrix with columns indexed by $C \subseteq [m]$. Then RK and RBK methods can be obtained as special cases of the update rule (2) as follows:

- RK: Let $\mathbf{S}_k = e_i$, where $i = i_k$ is chosen in each iteration independently, with probability $p_i > 0$. In this setup the update rule simplifies to

$$x^{k+1} = x^k - \frac{\mathbf{A}_{:i} x^k - b_i}{\|\mathbf{A}_{:i}\|_2^2} \mathbf{A}_{:i}^\top. \quad (3)$$

- RBK: Let $\mathbf{S} = \mathbf{I}_{:C}$, where $C = C_k$ is chosen in each iteration independently, with probability $p_C \geq 0$. In this setup the update rule simplifies to

$$x^{k+1} = x^k - \mathbf{A}_{:C}^\top (\mathbf{A}_{:C} \mathbf{A}_{:C}^\top)^\dagger (\mathbf{A}_{:C} x^k - b_C). \quad (4)$$

In this paper we are interested in two particular extension of the above methods: the randomized Kaczmarz method with momentum (mRK) and its block variant, the randomized block Kaczmarz with momentum (mRBK), both proposed and analyzed in [21]. Before we describe these two methods, let us summarize the main connections between the Kaczmarz methods for solving linear systems and gossip algorithms, as presented in [19].

In [12], [29], [21], it was shown that even in the case of consistent linear systems with *multiple* solutions, Kaczmarz-type methods converge linearly to one particular solution: the projection of the initial iterate x^0 onto the solution set

¹In order to recover a randomized coordinate descent method, one also needs to perform projections with respect to a more general Euclidean norm. However, for simplicity, in this work we only consider the standard Euclidean norm.

of the linear system. This naturally leads to the formulation of the *best approximation problem*:

$$\min_{x=(x_1, \dots, x_n) \in \mathcal{R}^n} \frac{1}{2} \|x - x^0\|^2 \quad \text{subject to} \quad \mathbf{A}x = b. \quad (5)$$

Above, $\mathbf{A} \in \mathcal{R}^{m \times n}$ and $\|\cdot\|$ is the standard Euclidean norm. By $x^* = \Pi_{\mathcal{L}}(x^0)$ we denote the solution of (5).

In [19] it was shown how RK and RBK work as gossip algorithms when applied to a special linear system encoding the underlying network. In particular the following definition is used:

Definition 2.1 ([19]): A linear system $\mathbf{A}x = b$ is called “average consensus (AC) system” when $\mathbf{A}x = b$ is equivalent to saying that $x_i = x_j$ for all $(i, j) \in \mathcal{E}$.

Note that many linear systems satisfy the above definition. For example, we can choose $b = 0$ and $\mathbf{A} \in \mathcal{R}^{|\mathcal{E}| \times n}$ to be the incidence matrix of \mathcal{G} . In this case, the row of the system corresponding to edge (i, j) directly encodes the constraint $x_i = x_j$. A different choice is to pick $b = 0$ and $\mathbf{A} = \mathbf{L}$, where \mathbf{L} is the Laplacian of \mathcal{G} . Note that depending on what AC system is used, RK and RBK have different interpretations as gossip protocols.

From now on we work with the AC system described in the first example. Since $b = 0$, the general sketch-and-project update rule (2) simplifies to:

$$x^{k+1} = [\mathbf{I} - \mathbf{A}^\top \mathbf{H}_k \mathbf{A}] x^k. \quad (6)$$

The convergence performance of RK and RBK for solving the best approximation problem (and as a result the average consensus problem) is described by the following theorem.

Theorem 2.2 ([11], [12]): Let $\{x^k\}$ be the iterates produced by (2). Then

$$\mathbb{E}[\|x^k - x^*\|^2] \leq \rho^k \|x^0 - x^*\|^2,$$

where x^* is the solution of (5), $\rho := 1 - \lambda_{\min}^+ \in [0, 1]$, and λ_{\min}^+ denotes the minimum nonzero eigenvalue of $\mathbf{W} := \mathbb{E}[\mathbf{A}^\top \mathbf{H} \mathbf{A}]$.

In [19], the behavior of both RK and RBK as gossip algorithms was described, and a comparison with the convergence results of existing randomized gossip protocols was made. In particular, it was shown that the most basic randomized gossip algorithm [3] (“randomly pick an edge $(i, j) \in \mathcal{E}$ and then replace the values stored at vertices i and j by their average”) is an instance of RK applied to the linear system $\mathbf{A}x = 0$, where the \mathbf{A} is the incidence matrix of \mathcal{G} . RBK can also be interpreted as a gossip algorithm:

Theorem 2.3 ([19], RBK as a Gossip Algorithm): Each iteration of RBK for solving $\mathbf{A}x = 0$ works as follows: 1) Select a random set of edges $\mathcal{S} \subseteq \mathcal{E}$, 2) Form subgraph \mathcal{G}_k of \mathcal{G} from the selected edges, 3) For each connected component of \mathcal{G}_k , replace node values with their average.

B. The Heavy Ball momentum

A detailed study of several (equivalent) *stochastic reformulations* of consistent linear systems was developed in [29]. This new viewpoint facilitated the development and analysis of relaxed variants (with relaxation parameter $\omega \in (0, 2)$) of

the sketch-and-project update (2). In particular, one of the reformulations is the *stochastic optimization problem*

$$\min_{x \in \mathcal{R}^n} f(x) := \mathbb{E}_{\mathbf{S} \sim \mathcal{D}} [f_{\mathbf{S}}(x)], \quad (7)$$

where

$$f_{\mathbf{S}}(x) := \frac{1}{2} \|\mathbf{A}x - b\|_{\mathbf{H}}^2 = \frac{1}{2} (\mathbf{A}x - b)^\top \mathbf{H} (\mathbf{A}x - b), \quad (8)$$

and \mathbf{H} is the random symmetric positive semi-definite matrix defined in (1).

Under certain (weak) condition on \mathcal{D} , the set of minimizers of f is identical to the set of the solutions of the linear system. In [29], problem (7) was solved via Stochastic Gradient Descent (SGD):

$$x^{k+1} = x^k - \omega \nabla f_{\mathbf{S}_k}(x^k). \quad (9)$$

Moreover, a linear rate of convergence was proved despite the fact that f is not necessarily strongly convex and that a fixed stepsize $\omega > 0$ is used.

Observe that that gradient of the stochastic function (8) is given by

$$\nabla f_{\mathbf{S}_k}(x) \stackrel{(8)}{=} \mathbf{A}^\top \mathbf{H}_k (\mathbf{A}x - b) \quad (10)$$

and as a result, it is easy to see that for $\omega = 1$, the SGD update (9) reduces to the sketch-and-project update (equation (2)).

The recent works [20], [21] analyze momentum variants of SGD, with the goal to accelerate the convergence of the method for solving problem (7). SGD with momentum—also known as the stochastic heavy ball method (SHB)—is very popular in the optimization literature for solving stochastic optimization problems, and it is also extremely popular in areas such as deep learning [32], [33], [15], [35]. However, even though SHB is used extensively in practice, its theoretical convergence behavior is not well understood. To the best of our knowledge, [20], [21] are the first that prove linear convergence of SHB in any setting.

The update rule of SHB for solving problem (7) is formally presented in the following algorithm:

Algorithm 1 Stochastic Heavy Ball (SHB)

- 1: **Parameters:** Distribution \mathcal{D} from which method samples matrices; stepsize/relaxation parameter $\omega \in \mathcal{R}$; momentum parameter β .
 - 2: **Initialize:** $x_0, x_1 \in \mathcal{R}^n$
 - 3: **for** $k = 1, 2, \dots$ **do**
 - 4: Draw a fresh $\mathbf{S}_k \sim \mathcal{D}$
 - 5: Set $x^{k+1} = x^k - \omega \nabla f_{\mathbf{S}_k}(x^k) + \beta(x^k - x^{k-1})$
 - 6: **end for**
 - 7: **Output:** The last iterate x^k
-

Note that using the expression of the stochastic gradient (10), the update rule of SHB can be written more explicitly as follows:

$$x^{k+1} = x^k - \omega \mathbf{A}^\top \mathbf{H}_k (\mathbf{A}x^k - b) + \beta(x^k - x^{k-1}). \quad (11)$$

Using the same choice of distribution \mathcal{D} as in equation (3) and (4), we can now obtain momentum variants of RK and RBK:

- RK with momentum (mRK) :

$$x^{k+1} = x^k - \omega \frac{\mathbf{A}_i: x^k - b_i}{\|\mathbf{A}_i\|_2^2} \mathbf{A}_i^\top + \beta(x^k - x^{k-1})$$

- RBK with momentum (mRBK):

$$x^{k+1} = x^k - \omega \mathbf{A}_{C:}^\top (\mathbf{A}_C: \mathbf{A}_{C:}^\top)^\dagger (\mathbf{A}_C: x^k - b_C) + \beta(x^k - x^{k-1})$$

In [21], two main theoretical results describing the behavior of SHB (and as a result also the special cases mRK and mRBK) were presented:

Theorem 2.4 (Theorem 1, [21]): Choose $x^0 = x^1 \in \mathcal{R}^n$. Let $\{x^k\}_{k=0}^\infty$ be the sequence of random iterates produced by SHB. Let λ_{\min}^+ (resp, λ_{\max}) be the smallest nonzero (resp. largest) eigenvalue of \mathbf{W} . Assume $0 < \omega < 2$ and $\beta \geq 0$ and that the expressions $a_1 := 1 + 3\beta + 2\beta^2 - (\omega(2-\omega) + \omega\beta)\lambda_{\min}^+$ and $a_2 := \beta + 2\beta^2 + \omega\beta\lambda_{\max}$ satisfy $a_1 + a_2 < 1$. Then

$$\mathbb{E}[\|x^k - x^*\|^2] \leq q^k (1 + \delta) \|x^0 - x^*\|^2, \quad (12)$$

and

$$\mathbb{E}[f(x^k)] \leq q^k \frac{\lambda_{\max}}{2} (1 + \delta) \|x^0 - x^*\|^2,$$

where $q = \frac{a_1 + \sqrt{a_1^2 + 4a_2}}{2}$ and $\delta = q - a_1$. Moreover, $a_1 + a_2 \leq q < 1$.

Theorem 2.5 (Theorem 4, [21]): Let $\{x^k\}_{k=0}^\infty$ be the sequence of random iterates produced by SHB, started with $x^0 = x^1 \in \mathcal{R}^n$, with relaxation parameter (stepsize) $0 < \omega \leq 1/\lambda_{\max}$ and momentum parameter $(1 - \sqrt{\omega\lambda_{\min}^+})^2 < \beta < 1$. Let $x^* = \Pi_{\mathcal{C}}^{\mathbf{B}}(x^0)$. Then there exists a constant $C > 0$ such that for all $k \geq 0$ we have $\|\mathbb{E}[x^k - x^*]\|^2 \leq \beta^k C$.

Corollary 1: (i) If we choose $\omega = 1$ and $\beta = (1 - \sqrt{0.99\lambda_{\min}^+})^2$, then the iteration complexity becomes: $\tilde{O}(\sqrt{1/\lambda_{\min}^+})$.
(ii) If we choose $\omega = 1/\lambda_{\max}$ and $\beta = (1 - \sqrt{\frac{0.99\lambda_{\min}^+}{\lambda_{\max}}})^2$, then the iteration complexity becomes: $\tilde{O}(\sqrt{\lambda_{\max}/\lambda_{\min}^+})$.

III. RANDOMIZED GOSSIP PROTOCOLS WITH MOMENTUM

Having presented SHB for solving the stochastic optimization problem (7) and describing its sketch-and-project nature (11), let us now describe its behavior as a randomized gossip protocol when applied to solving the AC system $\mathbf{A}x = 0$, where $\mathbf{A} \in |\mathcal{E}| \times n$ is the incidence matrix of the network.

Since $b = 0$, method (11) can be simplified to:

$$x^{k+1} = [\mathbf{I} - \omega \mathbf{A}^\top \mathbf{H}_k \mathbf{A}] x^k + \beta(x^k - x^{k-1}). \quad (13)$$

In the rest of this section we will focus on two special cases of the above method: RK with momentum and RBK with momentum.

Algorithm 2 mRK: Randomized Kaczmarz with momentum as a gossip algorithm

- 1: **Parameters:** Distribution \mathcal{D} from which method samples matrices; stepsize/relaxation parameter $\omega \in \mathcal{R}$; heavy ball/momentum parameter β .
 - 2: **Initialize:** $x_0, x_1 \in \mathcal{R}^n$
 - 3: **for** $k = 1, 2, \dots$ **do**
 - 4: Pick an edge $e = (i, j)$ following the distribution \mathcal{D}
 - 5: The values of the nodes are updated as follows:
 - Node i : $x_i^{k+1} = \frac{2-\omega}{2}x_i^k + \frac{\omega}{2}x_j^k + \beta(x_i^k - x_i^{k-1})$
 - Node j : $x_j^{k+1} = \frac{2-\omega}{2}x_j^k + \frac{\omega}{2}x_i^k + \beta(x_j^k - x_j^{k-1})$
 - Any other node ℓ : $x_\ell^{k+1} = x_\ell^k + \beta(x_\ell^k - x_\ell^{k-1})$
 - 6: **end for**
 - 7: **Output:** The last iterate x^k
-

A. Randomized Kaczmarz Gossip with momentum

When RK is applied to solve an AC system $\mathbf{A}x = 0$, one recovers the famous pairwise gossip algorithm [3]. Algorithm 2 describes how the relaxed variant of randomized Kaczmarz with momentum behaves as a gossip algorithm. See also Figure (1) for a graphical illustration of the method.

Remark 3.1: Note that in the special case with $\beta = 0$ (no momentum parameter) only the two nodes of edge $e = (i, j)$ update their values. In this case the two nodes do not update their values to their exact average but to a convex combination that depends on the stepsize $\omega \in (0, 2)$. To obtain the pairwise gossip algorithm of [3], we should further choose $\omega = 1$.

Distributed Nature of the Algorithm: There are many ways to implement mRK in a distributed fashion. Here we highlight a few:

- *Asynchronous pairwise broadcast gossip:* In this protocol each node $i \in \mathcal{V}$ of the network \mathcal{G} has a clock that ticks at the times of a rate 1 Poisson process. The inter-tick times are exponentially distributed, independent across nodes, and independent across time. This is equivalent to a global clock ticking at a rate n Poisson process which wakes up an edge of the network at random. In particular, in this implementation mRK works as follows: In the k^{th} iteration (time slot) the clock of node i ticks and node i randomly contact one of its neighbors and simultaneously broadcast a signal to inform the nodes of the whole network that is updating (this signal does not contain any private information of node i). The two nodes (i, j) share their information and update their private values following the update rule of Algorithm 2 while all the other nodes updating their values using their own information. In each iteration only one pair of nodes exchange their private values.
- *Synchronous Pairwise gossip:* In this protocol a single global clock is available to all nodes. The time is assumed to be slotted commonly across nodes and in each time slot only a pair of nodes of the network is randomly activated and exchange their information following the update rule of Algorithm 2. The remaining not activated nodes update their values using their own

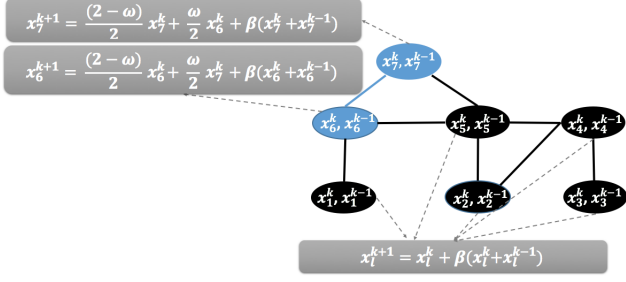


Fig. 1: Example of how mRK works as gossip algorithm. In the presented network the edge that connects nodes 6 and 7 is randomly selected. The pair of nodes exchange their information and update their values following the update rule of the Algorithm 2 while the rest of the nodes, $\ell \in [5]$, update their values using only their own previous private values.

last two private values. Note that this implementation of mRK comes with the disadvantage that requires a central entity which choose the activate pair of nodes in each step.

- *Asynchronous Pairwise gossip with common memory:* Note that the update rule of the nodes of the active pair in Algorithm 2 can rewritten as follows:

$$x_i^{k+1} = x_i^k + \beta(x_i^k - x_i^{k-1}) + \frac{\omega}{2}(x_j^k - x_i^k)$$

$$x_j^{k+1} = x_j^k + \beta(x_j^k - x_j^{k-1}) + \frac{\omega}{2}(x_i^k - x_j^k)$$

In particular observe that in their update rule they have the expression $x_i^k + \beta(x_i^k - x_i^{k-1})$ which is precisely the update of all non activate nodes of the network. Thus if we assume that the nodes share a common counter that counts how many iterations take place and each node i saves also the last iterate k_i that it was activated then the algorithm can work in distributed fashion as follows:

Let us denote the number of total iterations (common counter) that becomes available to the activate nodes of each step as K and let us define with $i_k = K - k_i$ the number of iterations between the current iterate and the last time that the i^{th} node is picked (iteration k_i) then the update rule of the Algorithm 2 can be equivalently expressed as:

- Pick an edge $e = (i, j)$ at random following the distribution \mathcal{D} .
- The private values of the nodes of the network are updated as follows:

$$x_i^{k+1} = i_k [x_i^k + \beta(x_i^k - x_i^{k-1})] + \frac{\omega}{2}(x_j^k - x_i^k)$$

$$x_j^{k+1} = j_k [x_j^k + \beta(x_j^k - x_j^{k-1})] + \frac{\omega}{2}(x_i^k - x_j^k)$$

$$k_i = k + 1$$

$$k_j = k + 1$$

Any other node ℓ : $x_\ell^{k+1} = x_\ell^k$

B. Connection with the accelerated gossip algorithm

In the randomized gossip literature there in one particular method closely related to our approach. It was first proposed in [4] and its analysis under strong conditions was presented in [16]. In this paper local memory is exploited by installing shift registers at each agent. In particular we are interested in the case of just two registers where the first stores the agent's current value and the second the agent's value before the latest update. The algorithm can be described as follows. Suppose that edge $e = (i, j)$ is chosen at time k . Then,

- Node i : $x_i^{k+1} = \omega \left(\frac{x_i^k + x_j^k}{2} \right) + (1 - \omega)x_i^{k-1}$
- Node j : $x_j^{k+1} = \omega \left(\frac{x_i^k + x_j^k}{2} \right) + (1 - \omega)x_j^{k-1}$
- Any other node ℓ : $x_\ell^{k+1} = x_\ell^k$

where $\omega \in [1, 2)$. The method was analyzed in [16] under a strong assumption on the probabilities of choosing the pair of nodes that as the authors mentioned is unrealistic in practical scenarios, and for networks like the random geometric graphs. At this point we should highlight that the results presented in [21] hold for essentially any distribution \mathcal{D} and as a result such a problem cannot occur.

Note also that in the case that we choose $\beta = \omega - 1$ in the update rule of Algorithm 2, then our method is simplified to:

- Node i : $x_i^{k+1} = \omega \left(\frac{x_i^k + x_j^k}{2} \right) + (1 - \omega)x_i^{k-1}$
- Node j : $x_j^{k+1} = \omega \left(\frac{x_i^k + x_j^k}{2} \right) + (1 - \omega)x_j^{k-1}$
- Any other node ℓ : $x_\ell^{k+1} = \omega x_\ell^k + (1 - \omega)x_\ell^{k-1}$

In order to apply Theorem 2.4, we need to assume that $0 < \omega < 2$ and $\beta = \omega - 1 \geq 0$ which also means that $\omega \in [1, 2)$. Thus for $\omega \in [1, 2)$ and momentum parameter $\beta = \omega - 1$ it is easy to see that our approach is very similar to the shift-register algorithm. Both methods update the selected pair of nodes in the same way. However, in our case the other nodes of the network do not remain idle but instead also update their values using their own previous information.

Using a diagonal momentum $n \times n$ matrix $\mathbf{B} = \text{Diag}(\mathbf{B}_{11}, \mathbf{B}_{22}, \dots, \mathbf{B}_{nn})$, the two algorithms above can be expressed as:

$$x^{k+1} = x^k - \omega \left(\frac{x_i^k - x_j^k}{2} (e_i - e_j) \right) + \mathbf{B}(x^k - x^{k-1}). \quad (14)$$

In particular, in our algorithm every element on the diagonal is equal to $\beta = \omega - 1$, while in [4] all values on the diagonal are zeros except for the two values $\mathbf{B}_{ii} = \mathbf{B}_{jj} = \omega - 1$.

Remark 3.2: The shift register case and our algorithm can be seen as two limit cases of the update rule (14). In particular, note that the shift register method uses only two non-zero diagonal elements in matrix \mathbf{B} , while our method has a full diagonal. We believe that further methods can be developed in the future by exploring the cases where more than two but not all elements of the diagonal matrix \mathbf{B} are non-zero. It might be possible to obtain better convergence if one carefully chooses these values based on the network topology. We leave this as an open problem for future research.

Algorithm 3 Randomized Block Kaczmarz Gossip with momentum

- 1: **Parameters:** Distribution \mathcal{D} from which method samples matrices; stepsize/relaxation parameter $\omega \in \mathcal{R}$; heavy ball/momentum parameter β .
- 2: **Initialize:** $x^0, x^1 \in \mathcal{R}^n$
- 3: **for** $k = 1, 2, \dots$ **do**
- 4: Select a random set of edges $\mathcal{S} \subseteq \mathcal{E}$
- 5: Form subgraph \mathcal{G}_k of \mathcal{G} from the selected edges
- 6: Node values are updated as follows:
 - For each connected component \mathcal{V}_r^k of \mathcal{G}_k , replace the values of its nodes with:

$$x_i^{k+1} = \omega \left[\frac{\sum_{j \in \mathcal{V}_r^k} x_j^k}{|\mathcal{V}_r^k|} \right] + (1 - \omega)x_i^k + \beta(x_i^k - x_i^{k-1}) \quad (16)$$

- Any other node ℓ : $x_\ell^{k+1} = x_\ell^k + \beta(x_\ell^k - x_\ell^{k-1})$
- 7: **end for**
 - 8: **Output:** The last iterate x^k
-

C. Randomized block Kaczmarz gossip with momentum

Recall that Theorem 2.3 says how RBK (with no momentum and no relaxation) can be interpreted as a gossip algorithm. Now we use this result to explain how relaxed RBK with momentum works. Note that the update rule of RBK with momentum can be rewritten as follows:

$$x^{k+1} \stackrel{(13)}{=} \omega(\mathbf{I} - \mathbf{A}^\top \mathbf{H}_k \mathbf{A})x^k + (1 - \omega)x^k + \beta(x^k - x^{k-1}), \quad (15)$$

where $(\mathbf{I} - \mathbf{A}^\top \mathbf{H}_k \mathbf{A})x^k$ is the update rule of RBK (6).

Thus, in analogy to the simple RBK, in the k^{th} step, a random set of edges is selected and a number (say q) connected components are formed as a result. This includes the connected components that belong to both sub-graph \mathcal{G}_k and also the singleton connected components (nodes outside the \mathcal{G}_k). Let us define the set of the nodes that belong in the $r \in [q]$ connected component at the k^{th} step \mathcal{V}_r^k , such that $\mathcal{V} = \bigcup_{r=\{1,2,\dots,q\}} \mathcal{V}_r^k$ and $|\mathcal{V}| = \sum_{r=1}^q |\mathcal{V}_r^k|$ for any $k > 0$.

Using the update rule (15), Algorithm 3 shows how mRBK is updating the private values of the nodes of the network (see also Figure 2 for the graphical interpretation).

Note that in the update rule of mRBK the nodes that are not attached to a selected edge (do not belong in the sub-graph \mathcal{G}_k) update their values via $x_\ell^{k+1} = x_\ell^k + \beta(x_\ell^k - x_\ell^{k-1})$. By considering these nodes as singleton connected components their update rule is exactly the same with the nodes of sub-graph \mathcal{G}_k . This is easy to see as follows:

$$\begin{aligned} x_\ell^{k+1} &= \omega \left[\frac{\sum_{j \in \mathcal{V}_r^k} x_j^k}{|\mathcal{V}_r^k|} \right] + (1 - \omega)x_\ell^k + \beta(x_\ell^k - x_\ell^{k-1}) \\ &= \omega \left[\frac{x_\ell^k}{1} \right] + (1 - \omega)x_\ell^k + \beta(x_\ell^k - x_\ell^{k-1}) \\ &= x_\ell^k + \beta(x_\ell^k - x_\ell^{k-1}) \end{aligned} \quad (17)$$

Remark 3.3: Note that in the special case that only one edge is selected in each iteration ($\mathbf{S}_k \in \mathcal{R}^{m \times 1}$) the update rule of mRBK is simplified to the update rule of mRK. In

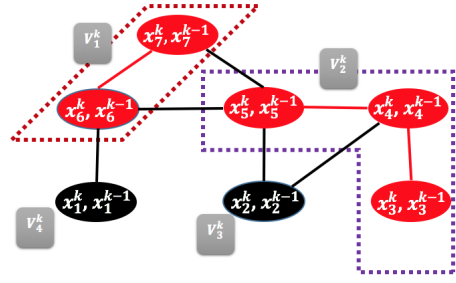


Fig. 2: Example of how the mRBK method works as gossip algorithm. In the presented network in the k^{th} iteration the red edges are randomly chosen and they form subgraph \mathcal{G}_k (from the red edges) and also four connected component. In this figure V_1^k and V_2^k are the two connected components that belong in the subgraph \mathcal{G}_k while V_3^k and V_4^k are the singleton connected components. Then the nodes update their values by communicate with the other nodes of their connected component using the update rule (16). For example the node number 5 that belongs in the connected component V_2^k will update its value using the values of node 4 and 3 that also belong in the same component as follows: $x_5^{k+1} = \omega \left[\frac{x_3^k + x_4^k + x_5^k}{3} \right] + (1 - \omega)x_5^k + \beta(x_5^k - x_5^{k-1})$.

this case the sub-graph \mathcal{G}_k is the pair of the two selected edges.

Remark 3.4: In [19] it was shown that several existing gossip protocols for solving the average consensus problem are special cases of the simple RBK (Theorem 2.3). For example two gossip algorithms that can be cast as special cases of the simple RBK are the path averaging proposed in [2] and the clique gossiping [18]. In path averaging, in each iteration a path of nodes is selected and its nodes update their values to their exact average ($\omega = 1$). In clique gossiping, the network is already divided into cliques and a through a random procedure a clique is activated and the nodes of it update their values to their exact average ($\omega = 1$). Since mRBK contains simple RBK as a special case for $\beta = 0$, we expect that these special protocols can also be accelerated with the addition of momentum parameter $\beta \in (0, 1)$.

D. Mass preservation

One of the key properties of some of the most efficient randomized gossip algorithms is mass preservation. If a gossip algorithm has this property it means that the sum (and as a result the average) of the private values of the nodes remains fixed during the iterative procedure. That is:

$$\sum_{i=1}^n x_i^k = \sum_{i=1}^n x_i^0, \quad \forall k \geq 1.$$

The original pairwise gossip algorithm proposed in [3] satisfied the mass preservation property, while existing accelerated gossip algorithms [4], [16] preserving a scaled sum.

In this section we show that the two proposed protocols presented above also have a mass preservation property. In particular, we prove mass preservation for the case of the block randomized gossip protocol (Algorithm 3) with

momentum. This is sufficient since the Kaczmarz gossip with momentum (mRK) can be cast as special case.

Theorem 3.1: Let us assume that $x^0 = x^1$. That is, the two registers of each node have the same initial value. Then for the Algorithms 2 and 3 we have $\sum_{i=1}^n x_i^k = \sum_{i=1}^n c_i$ for any $k \geq 0$ and as a result, $\frac{1}{n} \sum_{i=1}^n x_i^k = \bar{c}$.

Proof: We prove the result for the more general Algorithm 3. Let assume that in the k^{th} step of the method q connected components are formed. Let us also define the set of the nodes of each connected component as \mathcal{V}_r^k such that $\mathcal{V} = \bigcup_{r=\{1,2,\dots,q\}} \mathcal{V}_r^k$ and $|\mathcal{V}| = \sum_{r=1}^q |\mathcal{V}_r^k|$ for any $k > 0$. Thus:

$$\sum_{i=1}^n x_i^{k+1} = \sum_{i \in \mathcal{V}_1^k} x_i^{k+1} + \sum_{i \in \mathcal{V}_2^k} x_i^{k+1} + \dots + \sum_{i \in \mathcal{V}_q^k} x_i^{k+1} \quad (18)$$

Let us first focus, without loss of generality, on connected component $r \in [q]$ and simplify the expression for the sum of its nodes:

$$\begin{aligned} \sum_{i \in \mathcal{V}_r^k} x_i^{k+1} &\stackrel{(16)}{=} \sum_{i \in \mathcal{V}_r^k} \omega \left(\frac{\sum_{j \in \mathcal{V}_r^k} x_j^k}{|\mathcal{V}_r^k|} \right) + (1 - \omega) \sum_{i \in \mathcal{V}_r^k} x_i^k \\ &\quad + \beta \sum_{i \in \mathcal{V}_r^k} (x_i^k - x_i^{k-1}) \\ &= |\mathcal{V}_r^k| \frac{\omega \sum_{j \in \mathcal{V}_r^k} x_j^k}{|\mathcal{V}_r^k|} + (1 - \omega) \sum_{i \in \mathcal{V}_r^k} x_i^k \\ &\quad + \beta \sum_{i \in \mathcal{V}_r^k} (x_i^k - x_i^{k-1}) \\ &= (1 + \beta) \sum_{i \in \mathcal{V}_r^k} x_i^k - \beta \sum_{i \in \mathcal{V}_r^k} x_i^{k-1} \end{aligned} \quad (19)$$

By substituting this for all $r \in [q]$ into the right hand side of (18) and from the fact that $\mathcal{V} = \bigcup_{r=\{1,2,\dots,q\}} \mathcal{V}_r^k$:

$$\sum_{i=1}^n x_i^{k+1} = (1 + \beta) \sum_{i=1}^n x_i^k - \beta \sum_{i=1}^n x_i^{k-1} \quad (20)$$

Since $x^0 = x^1$, we have $\sum_{i=1}^n x_i^0 = \sum_{i=1}^n x_i^1$, and as a result $\sum_{i=1}^n x_i^k = \sum_{i=1}^n x_i^0$ for all $k \geq 0$. ■

IV. NUMERICAL EVALUATION

We devote this section to experimentally evaluate the performance of the proposed gossip algorithms: mRK and mRBK. In particular we perform three experiments. In the first two we focus on the performance of the mRK while in the last one on its block variant mRBK. In comparing the methods with their momentum variants we use the relative error measure $\|x^k - x^*\|^2 / \|x_0 - x^*\|^2$ where the starting vectors of values $x^0 = x^1 = c$ are taken to be always Gaussian vectors. For all of our experiments the horizontal axis represents the number of iterations.

The networks used in the experiments are the cycle (ring graph), the 2-dimension grid and the randomized geometric graph (RGG) with radius $r = \sqrt{\log(n)/n}$. All the code for the experiments is written in the Julia 0.6.3 programming language.

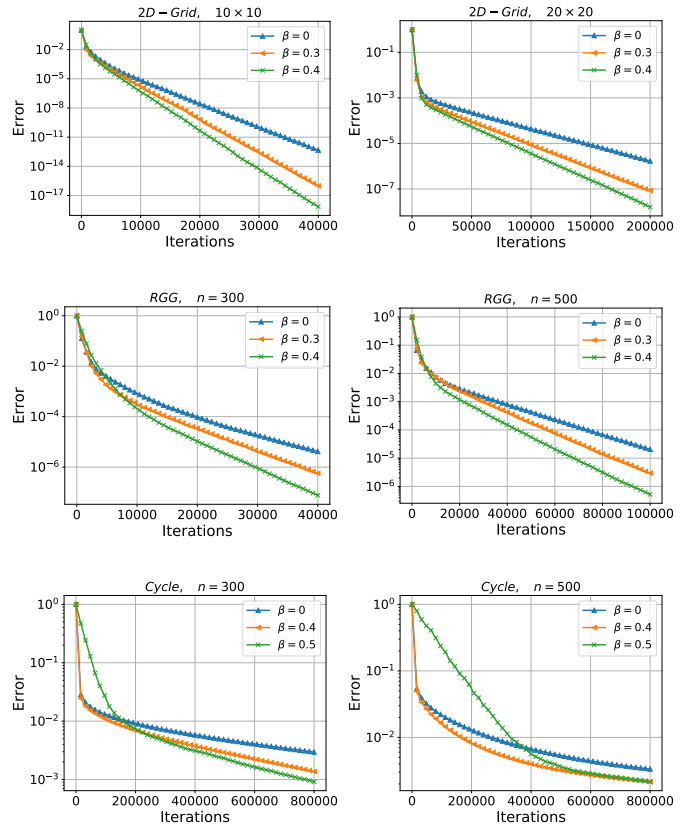


Fig. 3: Performance of mRK for fixed step-size $\omega = 1$ and several momentum parameters β in a cycle, 2-dimension grid and RGG. The choice $\beta = 0$ corresponds to the randomized pairwise gossip algorithm proposed in [3]; The n in the title of each plot indicates the number of nodes of the network. For the grid graph this is $n \times n$.

A. Impact of momentum parameter on mRK

Recall that in the simple pairwise gossip algorithm the two nodes that exchange information update their values to their exact average while all the other nodes remain idle. In our framework this method can be cast as special case of mRK when $\beta = 0$ and $\omega = 1$. In this experiment we keep always the stepsize to be $\omega = 1$ which means that the pair of the chosen nodes update their values to their exact average. We show that by choosing a suitable momentum parameter $\beta \in (0, 1)$ we can have faster convergence for all networks under study. See Figure 3 for more details.

B. Comparison with the Shift-Register

In this experiment we compare mRK with the shift register case when we choose the ω and β in such a way in order to satisfy the connection establish in Section III-B. That is, we choose $\beta = \omega - 1$ for any choice of $\omega \in (1, 2)$. Note that in all plots of Figure 4 our algorithm outperform the corresponding shift-register case.

C. Impact of momentum parameter on mRBK

In this experiment our goal is to show that the addition of momentum accelerates the RBK gossip algorithm proposed in [19]. Without loss of generality we choose the block size to be always equal to $\tau = 5$. That is the random matrix

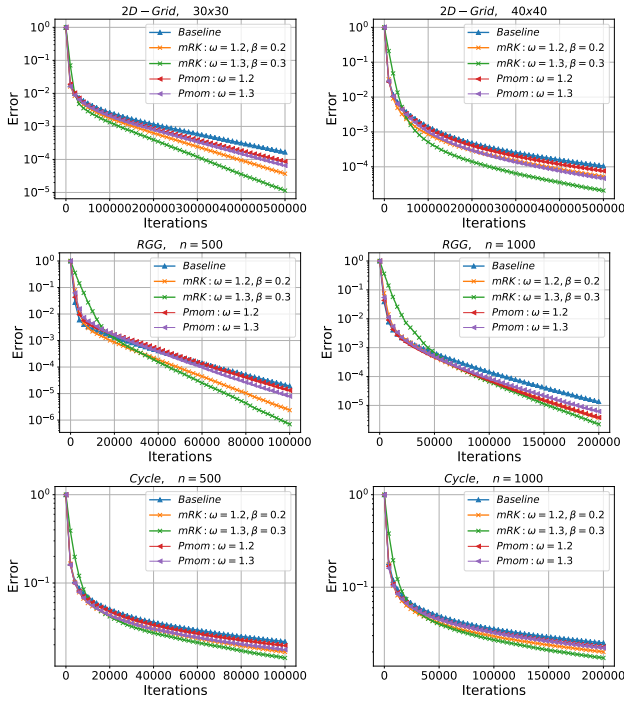


Fig. 4: Comparison of mRK with the pairwise momentum method (Pmom), shift-register algorithm proposed in [16]. In order to have a fair comparison we take always $\beta = \omega - 1$ for our algorithm and the stepsizes are chosen to be either $\omega = 1.2$ or $\omega = 1.3$. The baseline method in the plots denotes the simple not accelerated randomized pairwise gossip algorithm from [3]. The n in the title of each plot indicates the number of nodes of the network. For the grid graph this is $n \times n$.

$S_k \sim \mathcal{D}$ in the update rule of mRBK is always a $m \times 5$ column submatrix of the identity $m \times m$ matrix. Thus, in each iteration 5 edges of the network are chosen to form the subgraph \mathcal{G}_k and the values of the nodes are updated according to Algorithm 3. Note that similar plots can be obtained for any choice of block size. We run all algorithms with fixed stepsize $\omega = 1$. It is obvious that by choosing a suitable momentum parameter $\beta \in (0, 1)$ we have faster convergence than when $\beta = 0$, for all networks under study. See Figure 5 for more details.

V. CONCLUSION AND FUTURE RESEARCH

In this paper we present new accelerated randomized gossip algorithms using tools from numerical linear algebra and the area of randomized Kaczmarz methods for solving linear systems. In particular, using recently developed results on the stochastic reformulation of consistent linear systems we explain how stochastic heavy ball method for solving a specific quadratic stochastic optimization problem can be interpreted as gossip algorithm. To the best of our knowledge, it is the first time that such protocols are presented for solving the average consensus.

We believe that this work opens up many possible future venues of research. For example, using other Kaczmarz-type methods to solve particular linear systems we can obtain novel distributed protocols for solving the average consensus. In addition we believe that the gossip protocols presented

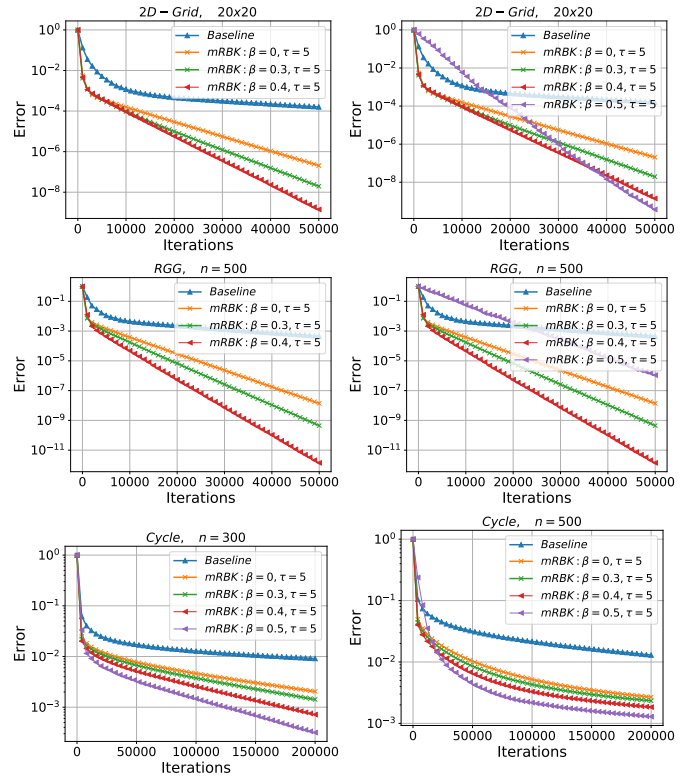


Fig. 5: Comparison of the mRBK with its no momentum variant RBK ($\beta = 0$) proposed in [19]. The stepsize for all methods is $\omega = 1$ and the block size is $\tau = 5$. The baseline method in the plots denote the simple randomized pairwise gossip algorithm (block $\tau = 1$) and is plotted to highlight the benefits of having larger block sizes. The n in the title of each plot indicates the number of nodes of the network. For the grid graph this is $n \times n$.

in this work can be extended to the more general setting of distributed optimization where the goal is to minimize the average of convex functions $(1/n) \sum_{i=1}^n f_i(x)$ in a distributed fashion.

REFERENCES

- [1] T.C. Aysal, M.E. Yildiz, A.D. Sarwate, and A. Scaglione. Broadcast gossip algorithms for consensus. *IEEE Trans. Signal Process.*, 57(7):2748–2761, 2009.
- [2] F. Bénézit, A.G. Dimakis, P. Thiran, and M. Vetterli. Order-optimal consensus through randomized path averaging. *IEEE Trans. Inf. Theory*, 56(10):5150–5167, 2010.
- [3] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah. Randomized gossip algorithms. *IEEE Trans. Inf. Theory*, 14(SI):2508–2530, 2006.
- [4] Ming Cao, Daniel A Spielman, and Edmund M Yeh. Accelerated gossip algorithms for distributed computation. In *44th Annual Allerton Conference on Communication, Control, and Computation*, pages 952–959. Citeseer, 2006.
- [5] G. Cybenko. Dynamic load balancing for distributed memory multiprocessors. *J. Parallel Distrib. Comput.*, 7(2):279–301, 1989.
- [6] Morris H DeGroot. Reaching a consensus. *Journal of the American Statistical Association*, 69(345):118–121, 1974.
- [7] A.G. Dimakis, S. Kar, J.M.F. Moura, M.G. Rabbat, and A. Scaglione. Gossip algorithms for distributed signal processing. *Proceedings of the IEEE*, 98(11):1847–1864, 2010.
- [8] A.G. Dimakis, A.D. Sarwate, and M.J. Wainwright. Geographic gossip: Efficient averaging for sensor networks. *IEEE Trans. Signal Process.*, 56(3):1205–1216, 2008.

- [9] Y.C. Eldar and D. Needell. Acceleration of randomized Kaczmarz method via the Johnson–Lindenstrauss lemma. *Numerical Algorithms*, 58(2):163–177, 2011.
- [10] N.M. Freris and A. Zouzias. Fast distributed smoothing of relative measurements. In *Decision and Control (CDC), 2012 IEEE 51st Annual Conference on*, pages 1411–1416. IEEE, 2012.
- [11] R.M. Gower and P. Richtárik. Randomized iterative methods for linear systems. *SIAM. J. Matrix Anal. & Appl.*, 36(4):1660–1690, 2015.
- [12] R.M. Gower and P. Richtárik. Stochastic dual ascent for solving linear systems. *arXiv preprint arXiv:1512.06890*, 2015.
- [13] F. Hanzely, J. Konečný, N. Loizou, P. Richtárik, and D. Grishchenko. Privacy preserving randomized gossip algorithms. *arXiv preprint arXiv:1706.07636*, 2017.
- [14] S. Kaczmarz. Angenäherte auflösung von systemen linearer gleichungen. *Bulletin International de l'Academie Polonaise des Sciences et des Lettres*, 35:355–357, 1937.
- [15] A. Krizhevsky, I. Sutskever, and G.E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, pages 1097–1105, 2012.
- [16] J. Liu, B.D.O. Anderson, M. Cao, and A.S. Morse. Analysis of accelerated gossip algorithms. *Automatica*, 49(4):873–883, 2013.
- [17] J. Liu and S. Wright. An accelerated randomized Kaczmarz algorithm. *Mathematics of Computation*, 85(297):153–178, 2016.
- [18] Yang Liu, Bo Li, Brian Anderson, and Guodong Shi. Clique gossiping. *arXiv preprint arXiv:1706.02540*, 2017.
- [19] N. Loizou and P. Richtárik. A new perspective on randomized gossip algorithms. In *4th IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2016.
- [20] N. Loizou and P. Richtárik. Linearly convergent stochastic heavy ball method for minimizing generalization error. *NIPS-Workshop on Optimization for Machine Learning [arXiv preprint arXiv:1710.10737]*, 2017.
- [21] N. Loizou and P. Richtárik. Momentum and stochastic momentum for stochastic gradient, newton, proximal point and subspace descent methods. *arXiv preprint arXiv:1712.09677*, 2017.
- [22] A. Ma, D. Needell, and A. Ramdas. Convergence properties of the randomized extended Gauss-Seidel and Kaczmarz methods. *SIAM Journal on Matrix Analysis and Applications*, 36(4):1590–1604, 2015.
- [23] D. Needell. Randomized Kaczmarz solver for noisy linear systems. *BIT Numerical Mathematics*, 50(2):395–403, 2010.
- [24] D. Needell and J.A. Tropp. Paved with good intentions: analysis of a randomized block Kaczmarz method. *Linear Algebra Appl.*, 441:199–221, 2014.
- [25] D. Needell, R. Zhao, and A. Zouzias. Randomized block Kaczmarz method with projection for solving least squares. *Linear Algebra and its Applications*, 484:322–343, 2015.
- [26] D. Needell, R. Zhao, and A. Zouzias. Randomized block Kaczmarz method with projection for solving least squares. *Linear Algebra Appl.*, 484:322–343, 2015.
- [27] A. Olshevsky and J.N. Tsitsiklis. Convergence speed in distributed consensus and averaging. *SIAM J. Control Optim.*, 48(1):33–55, 2009.
- [28] Alex Olshevsky. Linear time average consensus on fixed graphs and implications for decentralized optimization and multi-agent control. *arXiv preprint arXiv:1411.4186*, 2014.
- [29] P. Richtárik and M. Takáč. Stochastic reformulations of linear systems: algorithms and convergence theory. *arXiv:1706.01108*, 2017.
- [30] F. Schöpfer and D.A. Lorenz. Linear convergence of the randomized sparse Kaczmarz method. *arXiv preprint arXiv:1610.02889*, 2016.
- [31] T. Strohmer and R. Vershynin. A randomized Kaczmarz algorithm with exponential convergence. *J. Fourier Anal. Appl.*, 15(2):262–278, 2009.
- [32] I. Sutskever, J. Martens, G.E. Dahl, and G.E. Hinton. On the importance of initialization and momentum in deep learning. *ICML (3)*, 28:1139–1147, 2013.
- [33] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. In *CVPR*, pages 1–9, 2015.
- [34] 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.
- [35] A.C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht. The marginal value of adaptive gradient methods in machine learning. *arXiv preprint arXiv:1705.08292*, 2017.
- [36] L. Xiao, S. Boyd, and S. Lall. A scheme for robust distributed sensor fusion based on average consensus. In *Information Processing in Sensor Networks, 2005. IPSN 2005. Fourth International Symposium on*, pages 63–70. IEEE, 2005.
- [37] Lin Xiao and Stephen Boyd. Fast linear iterations for distributed averaging. *Systems & Control Letters*, 53(1):65–78, 2004.
- [38] J.Y. Yu and M.G. Rabbat. Performance comparison of randomized gossip, broadcast gossip and collection tree protocol for distributed averaging. In *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2013 IEEE 5th International Workshop on*, pages 93–96. IEEE, 2013.
- [39] A. Zouzias and N.M. Freris. Randomized extended Kaczmarz for solving least squares. *SIAM. J. Matrix Anal. & Appl.*, 34(2):773–793, 2013.
- [40] A. Zouzias and N.M. Freris. Randomized gossip algorithms for solving Laplacian systems. In *Control Conference (ECC), 2015 European*, pages 1920–1925. IEEE, 2015.