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 [20] 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 [35], [7] and recent [38], [37], [4] 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. [4] on randomized gossip algorithms motivated a fury of subsequent research and now gossip algorithms appear in many applications, including distributed data fusion in sensor networks [38], load balancing [6] and clock synchronization [11]. For a survey of selected relevant work prior to 2010, we refer the reader to the work of Dimakis et al. [8]. For more recent results on randomized gossip algorithms we suggest [40], [17], [28], [20], [24], [1]. See also [9], [2], [29], [14].

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 $\mathcal{V} = \{1, 2, \ldots, n\}$ 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 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

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. Our protocols are based on recently proposed ideas for the acceleration of randomized Kaczmarz methods for solving consistent linear systems [22] 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. 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 $[n] := \{1, 2, \ldots, n\}$. Boldface upper-case
letters denote matrices; \( \mathbf{I} \) is the identity matrix. By \( \mathcal{L} \) we denote the solution set of the linear system \( \mathbf{A} \mathbf{x} = \mathbf{b} \), where \( \mathbf{A} \in \mathbb{R}^{m \times n} \) and \( \mathbf{b} \in \mathbb{R}^m \). Throughout the paper, \( \mathbf{x}^* \) is the projection of \( \mathbf{x}^0 \) onto \( \mathcal{L} \) (that is, \( \mathbf{x}^* \) is the solution of the best approximation problem; see equation (5)). An explicit formula for the projection of \( \mathbf{x} \) onto set \( \mathcal{L} \) is given by

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

A matrix that often appears in our update rules is

\[
\mathbf{H} := \mathbf{S} (\mathbf{S}^\top \mathbf{A} \mathbf{A}^\top \mathbf{S})^{+} \mathbf{S}^\top,
\]

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

In the convergence analysis we use \( \lambda_{\text{min}}^{+} \) to indicate the smallest nonzero eigenvalue, and \( \lambda_{\text{max}}^{+} \) 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, \( \mathbf{x}_i^k = (x_i^1, \ldots, x_i^n) \in \mathbb{R}^n \) represents the vector with the private values of the \( i \)th node 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. In [20], 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 the Randomized Kaczmarz and Randomized Block Kaczmarz, popular methods for solving linear systems, work as gossip algorithms when applied to a special system encoding the underlying network. In [22], several classes of stochastic optimization algorithms enriched with \textit{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 \( \mathbf{A} \mathbf{x} = \mathbf{b} \) with many equations. The (deterministic) Kaczmarz method for solving consistent linear systems was originally introduced by Kaczmarz in 1937 [15]. 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 Strohmer and Vershynin [32]. This result sparked renewed interest in design of randomized methods for solving linear systems [25], [26], [10], [23], [39], [27], [31], [18]. More recently, Gower and Richtárik [12] 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 [12] for solving the consistent linear system \( \mathbf{A} \mathbf{x} = \mathbf{b} \) has the form

\[
x^{k+1} = x^k - \mathbf{A}^\top \mathbf{S}_k (\mathbf{S}_k^\top \mathbf{A} \mathbf{A}^\top \mathbf{S}_k)^{+} \mathbf{S}_k^\top (\mathbf{A} x^k - \mathbf{b})
\]

\[
(1) = x^k - \mathbf{A}^\top \mathbf{H}_k (\mathbf{A} x^k - \mathbf{b}),
\]

where in each iteration matrix \( \mathbf{S}_k \) is sampled afresh from an arbitrary distribution \( \mathcal{D} \). In [12] it was shown that many popular algorithms for solving linear systems, including RK method and randomized coordinate descent method can be cast as special cases of the above update by choosing\(^1\) 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 \( e_i \in \mathbb{R}^m \) be the \( i \)th unit coordinate vector in \( \mathbb{R}^m \) 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 (2) simplifies to

\[
x^{k+1} = x^k - \mathbf{A}^\top \mathbf{A} x^k - \mathbf{b},
\]

\[
(3)
\]

- **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 (2) simplifies to

\[
x^{k+1} = x^k - \mathbf{A}_C^\top (\mathbf{A}_C \mathbf{A}_C^\top)^{+} (\mathbf{A}_C \mathbf{x}^k - \mathbf{b}_C). \quad (4)
\]

In this paper we are interested in two particular extensions 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 [22]. Before we describe these two algorithms, let us summarize the main connections between the Kaczmarz methods for solving linear systems and gossip algorithms, as presented in [20].

In [13], [30], [22], it was shown that even in the case of consistent linear systems with \textit{multiple} solutions, Kaczmarz-type methods converge linearly to one particular solution: the projection of the initial iterate \( \mathbf{x}^0 \) onto the solution set of the linear system. This naturally leads to the formulation of the \textit{best approximation problem}:

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

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

In [20] it was shown how RK and RBK work as gossip algorithms when applied to a special linear system encoding the underlying network.

\textbf{Definition 2.1} ([20]): A linear system \( \mathbf{A} \mathbf{x} = \mathbf{b} \) is called “average consensus (AC) system” when \( \mathbf{A} \mathbf{x} = \mathbf{b} \) is equivalent to saying that \( x_i = x_j \) for all \( (i, j) \in \mathcal{E} \).

\(^1\)In 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.
Note that many linear systems satisfy the above definition. For example, we can choose $b = 0$ and $A \in \mathbb{R}^{E \times n}$ to be the incidence matrix of $G$. 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 $A = L$, where $L$ is the Laplacian of $G$. 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:

$$x^{k+1} = (I - A^\top H_k A)x^k.$$  

(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 ([12], [13]):** 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), $\rho := 1 - \lambda^\text{min} \in [0, 1]$, and $\lambda^\text{min}$ denotes the minimum nonzero eigenvalue of $W := \mathbb{E}[AHA]$.

In [20], 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 [4] ("randomly pick an edge $(i, j) \in E$ and then replace the values stored at vertices $i$ and $j$ by their average") is an instance of RK applied to the linear system $Ax = 0$, where the $A$ is the incidence matrix of $G$. RBK can also be interpreted as a gossip algorithm:

**Theorem 2.3 ([20], RBK as a Gossip Algorithm):** Each iteration of RBK for solving $Ax = 0$ works as follows: 1) Select a random set of edges $S \subseteq E$, 2) Form subgraph $G_k$ of $G$ from the selected edges, 3) For each connected component of $G_k$, 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 [30]. 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

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

(7)

$$f_S(x) := \frac{1}{2}\|Ax - b\|^2_H = \frac{1}{2}(Ax - b)^\top H(Ax - b),$$

(8)

and $H$ is the random symmetric positive semi-definite matrix defined in (1).

Under certain (weak) condition on $D$, the set of minimizers of $f$ is identical to the set of the solutions of the linear system. In [30], problem (7) was solved via Stochastic Gradient Descent (SGD):

$$x^{k+1} = x^k - \omega \nabla f_S(x^k),$$

(9)

and 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 the gradient of the stochastic function (8) is given by

$$\nabla f_S(x) \overset{(8)}{=} A^\top H_k(Ax - b).$$

(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 (2).

The recent works [21], [22] 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 a well known algorithm in the optimization literature for solving stochastic optimization problems, and it is extremely popular in areas such as deep learning [33], [34], [16], [36]. However, even though SHB is used extensively in practice, its theoretical convergence behavior is not well understood. To the best of our knowledge, [21], [22] 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 $D$ from which method samples matrices; stepsize/relaxation parameter $\omega \in \mathbb{R}$; momentum parameter $\beta$.
2: Initialize: $x^0$, $x^1 \in \mathbb{R}^n$
3: for $k = 1, 2, \ldots$ do
4: Draw a fresh $S_k \sim D$
5: Set $x^{k+1} = x^k - \omega \nabla f_{S_k}(x^k) + \beta(x^k - x^{k-1})$
6: end for
7: Output: The last iterate $x^k$

Using the expression for the stochastic gradient (10), the update rule of SHB can be written more explicitly:

$$x^{k+1} = x^k - \omega A^\top H_k(Ax_k - b) + \beta(x^k - x^{k-1}).$$

(11)

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

- RK with momentum (mRK):
  $$x^{k+1} = x^k - \omega \frac{A_i}{\|A_i\|_F^2} (Ax_i - b_i) + \beta(x^k - x^{k-1})$$

- RBK with momentum (mRBK):
  $$x^{k+1} = x^k - \omega A_{C_i}^\top (A_{C_i}A_{C_i})^\dagger(A_{C_i}x_k - b_C) + \beta(x^k - x^{k-1})$$

In [22], 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, [22]]):** Choose $x^0 = x^1 \in \mathbb{R}^n$, let $\{x^k\}_{k=0}^\infty$ be the sequence of random iterates produced by SHB. Let $\lambda^\text{min}$ (resp. $\lambda^\text{max}$) be the smallest nonzero (resp. largest) eigenvalue of $W$. Assume $0 < \omega < 2$ and $\beta \geq 0$ and that the expressions $\alpha_1 := 1 + 3\beta + 2\beta^2 - (\omega(2 - \omega) + \omega\beta)\lambda^\text{min}$ and $\alpha_2 := \beta + 3\beta^2 + 2\beta\lambda^\text{max}$ satisfy $\alpha_1 + \alpha_2 < 1$. Then

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

(12)

and $\mathbb{E}[f(x^k)] \leq q^k \frac{\lambda^\text{max}}{2} (1 + \delta) \|x^0 - x^*\|^2$, where $q = \frac{1}{2}(\alpha_1 + \sqrt{\alpha_1^2 + 4\alpha_2})$ and $\delta = q - \alpha_1$. Moreover, $\alpha_1 \leq q \leq 1$. 

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

1: Parameters: Distribution $D$ from which method samples matrices; stepsize/relaxation parameter $\omega \in \mathbb{R}$; heavy ball/momentum parameter $\beta$.
2: Initialize: $x^0, x \in \mathbb{R}^n$
3: for $k = 1, 2, \ldots$ do
4: Pick an edge $e = (i, j)$ following the distribution $D$
5: The values of the nodes are updated as follows:
   - Node $i$: $x^{k+1}_i = \frac{1}{2}(x^k_i + \beta(x^k_i - x^{k-1}_i)) + \frac{1}{2}\omega(x^k_j - x^{k-1}_j)$
   - Node $j$: $x^{k+1}_j = \frac{1}{2}(x^k_j + \beta(x^k_j - x^{k-1}_j)) + \frac{1}{2}\omega(x^k_i - x^{k-1}_i)$
   - Any other node $\ell$: $x^{k+1}_\ell = x^k_\ell + \beta(x^k_i - x^{k-1}_i)$
6: end for
7: Output: The last iterate $x^k$

Theorem 2.5 (Theorem 4, [22]): Let $\{x^k\}_{k=0}^\infty$ be the sequence of random iterates produced by SHB, started with $x^0 = x^1 \in \mathbb{R}^n$, with relaxation parameter (stepsize) $0 < \omega \leq 1/\lambda_{\text{max}}$ and momentum parameter $(1 - \sqrt{\omega\lambda_{\text{min}}^+})^2 < \beta < 1$. Let $x^* = \Pi_\mathcal{D}(x^0)$. Then there exists a constant $C > 0$ such that for all $k \geq 0$ we have $\|E[x^k - x^*]\|_2^2 \leq \beta^k C$.

Using Theorem 2.5 and by a proper combination of the stepsize and the momentum parameter $\beta$, SHB enjoys an accelerated linear convergence rate in mean. [22]

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

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 algorithm with momentum (Algorithm 2) can be rewritten as follows:

$\begin{align*}
x^{k+1}_i &= x^k_i + \beta(x^k_i - x^{k-1}_i) + \frac{1}{\omega}(x^k_j - x^k_i) \\
x^{k+1}_j &= x^k_j + \beta(x^k_j - x^{k-1}_j) + \frac{1}{\omega}(x^k_i - x^k_j)
\end{align*}$

In the rest of this section we focus on two special cases of (13): RK with momentum and RBK with momentum.

A. Randomized Kaczmarz Gossip with momentum

When RK is applied to solve an AC system $Ax = 0$, one recovers the famous pairwise gossip algorithm [4]. 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: In the special case that $\beta = 0$ (zero momentum) only the two nodes of edge $e = (i, j)$ update their values. In this case the two selected 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 [4], one should further choose $\omega = 1$.

Distributed Nature of the Algorithm: Here we highlight a few ways to implement mRK in a distributed fashion. Asynchronous pairwise broadcast gossip: In this protocol each node $i \in \mathcal{V}$ of the network $G$ 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 $\delta$ 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 last two private values. Note that this implementation of mRK comes with the disadvantage that requires a central entity which in each step requires to choose the activated pair of nodes.

Asynchronous pairwise gossip with common counter: Note that the update rule of the selected pair of nodes $(i, j)$ in Algorithm 2 can be rewritten as follows:

$\begin{align*}
x^{k+1}_i &= x^k_i + \beta(x^k_i - x^{k-1}_i) + \frac{1}{\omega}(x^k_j - x^k_i) \\
x^{k+1}_j &= x^k_j + \beta(x^k_j - x^{k-1}_j) + \frac{1}{\omega}(x^k_i - x^k_j)
\end{align*}$

In particular observe that the first part of the above expressions $x^k_1 + \beta(x^k_i - x^{k-1}_i)$ is exactly the same with the update rule of the non activate nodes at $k^{th}$ step. Thus, if we assume that all nodes share a common counter that keeps track of the current iteration count, and if each node $i \in \mathcal{V}$ remembers also the last iteration counter $k_i$ when it was activated, then step 5 of Algorithm 2 takes the form:

$\begin{align*}
x^{k+1}_i &= i_k [x^k_i + \beta(x^k_i - x^{k-1}_i)] + \frac{1}{\omega}(x^k_j - x^k_i) \\
x^{k+1}_j &= j_k [x^k_j + \beta(x^k_j - x^{k-1}_j)] + \frac{1}{\omega}(x^k_i - x^k_j) \\
k_i &= k + 1; \\
\text{Any other node } \ell: x^{k+1}_\ell &= x^k_\ell
\end{align*}$

where $i_k = k - k_j$ (the $k^{th}$ node is activated.

\footnote{We speculate that a completely distributed synchronous gossip algorithm that finds pair of nodes in a distributed manner without any additional computational burden can be design following the same procedure proposed in Section III.C of [4].}
B. Connection with the accelerated gossip algorithm

In the randomized gossip literature there is one particular method closely related to our approach. It was first proposed in [5] and its analysis under strong conditions was presented in [17]. In this paper local memory is exploited by installing shift registers at each agent. In particular we are interested in the case of 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 \( \ell \): \( x_\ell^{k+1} = x_\ell^k \)

where \( \omega \in [1,2] \). The method was analyzed in [17] 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 [22] hold for essentially any distribution \( \mathcal{D} \) and as a result such a problem cannot occur.

Note also that if 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 \( \ell \): \( x_\ell^{k+1} = x_\ell^k \)

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 Algorithm 2 the not selected nodes of the network do not remain idle but instead update their values using their own previous information.

By defining the momentum matrix \( \mathbf{B} = \text{Diag}(\beta_1, \beta_2, \ldots, \beta_n) \), the above closely related algorithms can be expressed, in vector form, as:

\[
x^{k+1} = \frac{2}{\omega} (x_i^k - x_j^k)(e_i - e_j) + \mathbf{B}(x^k - x^{k-1}).
\]

In particular, in mRK every diagonal element of matrix \( \mathbf{B} \) is equal to \( \omega - 1 \), while in the algorithm of [5], [17] all the diagonal elements are zeros except the two values that correspond to nodes \( i \) and \( j \) that are equal to \( \beta_i = \beta_j = \omega - 1 \).

Remark 3.2: The shift register algorithm of [17] and Algorithm 2 can be seen as the two limit cases of the update rule (14). As we mentioned, the shift register method [17] uses only two non-zero diagonal elements in \( \mathbf{B} \), 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.

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:

\[
x^{k+1} = (1 - \omega)x^k + \beta(x^k - x^{k-1}).
\]

where \( \mathbf{I} - \mathbf{A}^\top \mathbf{H}_k \mathbf{A} \) 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 \( q \leq n \) 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 \( \mathcal{V}_r \), such that \( \mathcal{V} = \cup_{r \in [q]} \mathcal{V}_r \) and \( |\mathcal{V}| = \sum_{r=1}^q |\mathcal{V}_r| \) 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 subgraph \( \mathcal{G}_k \)) update their values via \( x_\ell^{k+1} = x_\ell^k + \beta(x_\ell^k - x^{k-1}) \).
property, while existing accelerated gossip algorithms [5], [17] 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:** 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 $\sum_{i=1}^n x^k_i = \sum_{i=1}^n c_i$ for any $k \geq 0$ and as a result, $\frac{1}{n} \sum_{i=1}^n x^k_i = \bar{c}$.

**Proof:** We prove the result for the more general Algorithm 3. Assume that in the $k^{th}$ step of the method $q$ connected components are formed. Let the set of the nodes of each connected component be $V^k_r$ so that $\mathcal{V} = \bigcup_{r=1}^{q} V^k_r$ and $|\mathcal{V}| = \sum_{r=1}^q |V^k_r|$ for any $k > 0$. Thus:

$$\sum_{i=1}^n x^{k+1}_i = \sum_{i\in V^k_r} x^{k+1}_i + \cdots + \sum_{i\in V^k_q} x^{k+1}_i \quad (18)$$

Let us first focus, without loss of generality, on connected component $r \in [q]$ and simplify the expression for the sum of its nodes: $\sum_{i\in V^k_r} x^{k+1}_i = \sum_{i\in V^k_r} (\omega x^k_i + (1 - \omega) \sum_{j \in V^k_r \setminus \{i\}} x^k_j + \beta (x^k_i - x^{k-1}_i))$.

Recall that in the simple pairwise gossip algorithm the path averaging proposed in [3] and the clique gossiping [19]. 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 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 the simple RBK as a special case (when $\beta = 0$), 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, $\sum_{i=1}^n x^k_i = \sum_{i=1}^n x^0_i$, $\forall k \geq 1$. The original pairwise gossip algorithm proposed in [4] satisfied the mass preservation property, while existing accelerated gossip algorithms [5], [17] 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:** 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 $\sum_{i=1}^n x^k_i = \sum_{i=1}^n c_i$ for any $k \geq 0$ and as a result, $\frac{1}{n} \sum_{i=1}^n x^k_i = \bar{c}$.

**Proof:** We prove the result for the more general Algorithm 3. Assume that in the $k^{th}$ step of the method $q$ connected components are formed. Let the set of the nodes of each connected component be $V^k_r$ so that $\mathcal{V} = \bigcup_{r=1}^{q} V^k_r$ and $|\mathcal{V}| = \sum_{r=1}^q |V^k_r|$ for any $k > 0$. Thus:

$$\sum_{i=1}^n x^{k+1}_i = \sum_{i\in V^k_r} x^{k+1}_i + \cdots + \sum_{i\in V^k_q} x^{k+1}_i \quad (18)$$

Let us first focus, without loss of generality, on connected component $r \in [q]$ and simplify the expression for the sum of its nodes: $\sum_{i\in V^k_r} x^{k+1}_i = \sum_{i\in V^k_r} (\omega x^k_i + (1 - \omega) \sum_{j \in V^k_r \setminus \{i\}} x^k_j + \beta (x^k_i - x^{k-1}_i))$.

Recall that in the simple pairwise gossip algorithm the path averaging proposed in [3] and the clique gossiping [19]. 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 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 the simple RBK as a special case (when $\beta = 0$), 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, $\sum_{i=1}^n x^k_i = \sum_{i=1}^n x^0_i$, $\forall k \geq 1$. The original pairwise gossip algorithm proposed in [4] satisfied the mass preservation property, while existing accelerated gossip algorithms [5], [17] 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:** 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 $\sum_{i=1}^n x^k_i = \sum_{i=1}^n c_i$ for any $k \geq 0$ and as a result, $\frac{1}{n} \sum_{i=1}^n x^k_i = \bar{c}$.

**Proof:** We prove the result for the more general Algorithm 3. Assume that in the $k^{th}$ step of the method $q$ connected components are formed. Let the set of the nodes of each connected component be $V^k_r$ so that $\mathcal{V} = \bigcup_{r=1}^{q} V^k_r$ and $|\mathcal{V}| = \sum_{r=1}^q |V^k_r|$ for any $k > 0$. Thus:

$$\sum_{i=1}^n x^{k+1}_i = \sum_{i\in V^k_r} x^{k+1}_i + \cdots + \sum_{i\in V^k_q} x^{k+1}_i \quad (18)$$

Let us first focus, without loss of generality, on connected component $r \in [q]$ and simplify the expression for the sum of its nodes: $\sum_{i\in V^k_r} x^{k+1}_i = \sum_{i\in V^k_r} (\omega x^k_i + (1 - \omega) \sum_{j \in V^k_r \setminus \{i\}} x^k_j + \beta (x^k_i - x^{k-1}_i))$.

Recall that in the simple pairwise gossip algorithm the path averaging proposed in [3] and the clique gossiping [19]. 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 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 the simple RBK as a special case (when $\beta = 0$), 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, $\sum_{i=1}^n x^k_i = \sum_{i=1}^n x^0_i$, $\forall k \geq 1$. The original pairwise gossip algorithm proposed in [4] satisfied the mass preservation property, while existing accelerated gossip algorithms [5], [17] preserving a scaled sum.
Fig. 3: Performance of mRK for fixed step-size $\omega = 1$ and several momentum parameters $\beta$ in a 2-dimension grid, random geometric graph (RGG) and a cycle graph. The choice $\beta = 0$ corresponds to the randomized pairwise gossip algorithm proposed in [4]; The $n$ in the title of each plot indicates the number of nodes of the network. For the grid graph this is $n \times n$.

Fig. 4: Comparison of mRK and the pairwise momentum method (Pmom), proposed in [17] (shift-register algorithm of Section III-B). For fair comparison, the momentum parameter of mRK is $\beta = \omega - 1$ and the stepsizes are chosen to be either $\omega = 1.2$ or $\omega = 1.3$. The baseline method is the simple not accelerated randomized pairwise gossip algorithm from [4]. The $n$ in the title of each plot indicates the number of nodes of the network. For the grid graph this is $n \times n$.

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 average consensus problem.

This work opens up many possible future venues for research. For example, using other popular Kaczmarz-type methods to solve specific linear systems related to the underlying network can lead to the development of novel distributed protocols for average consensus. In addition, we speculate that the gossip protocols presented in this work can be extended to the more general setting of multi-agent consensus optimization where the goal is to minimize the average of convex functions \( \frac{1}{n} \sum_{i=1}^{n} f_i(x) \) in a decentralized way [24].

REFERENCES

Fig. 5: Comparison of mRBK with its no momentum variant RBK ($\beta = 0$) proposed in [20]. The stepsize for all methods is $\tau = 5$ and the block size is $\tau = 5$. The baseline method in the plots denotes 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. For the grid graph this is $n \times n$.