Comparing Asynchronous Distributed Averaging Gossip Algorithms Over Scale-free Graphs

Wěipéng Huáng, Yanyang Wang, Gregory Provan

Abstract

Gossip algorithms are an approach for data dissemination within networks that can compute data aggregates efficiently. Gossip protocol is fascinating due to its capability of powering the classical protocols, its parallel to larger social structures and its feature of full decentralization. We deem computing the distributed average in a network a critical and fundamental procedure for distributed machine learning study. Thus, we are interested in exploring the three well-known gossip algorithms solving this problem, namely Push-pull, Push-sum, and Sum-weight gossip, particularly in asynchronous mode. This paper first attempts to connect the theoretical and the practical work, by proposing modifications to the general algorithms which make them more applicable in practice. Since no thorough empirical comparison of these gossip algorithms has been conducted, we compare three algorithms on two synthetic dataset and one real-world dataset, by the scale-free graph (Barabási-Albert graph). We show that all three algorithms have an extremely high accuracy for approximating the distributed mean although Push-sum would obtain bad result on few occasions. Also, Push-sum is the most efficient algorithm when the mean degree in the graph is small, and Push-sum and Push-pull are both efficient when the graph or the mean degree is large. On the other hand, the experimental results demonstrate that the performance (accuracy and efficiency) of each algorithm could be affected by variously distinct graph properties. Furthermore, we discuss the time complexity of three asynchronous gossip algorithms and prove that they are all bounded by a logarithmic function of the graph size and other properties.

Index Terms

Gossip Protocol; Randomized Algorithms; Distributed Computing; Empirical Study; Scale-free Graph

W Huáng is an independent researcher and was with the Insight Center for Data Analytics, University College Cork, Cork, Ireland. Email: weipeng.huang@outlook.com

Y Wang is an independent researcher. Email: yanyangwang1988@gmail.com

G Provan is with the Department of Computer Science, University College Cork, Cork, Ireland. Email: g.provan@cs.ucc.ie

Comparing Asynchronous Distributed Averaging Gossip Algorithms Over Scale-free Graphs

I. Introduction

COMPUTATION over large networks, such as social networks, web graphs, and wireless sensor networks, requires efficient algorithms given the vast size of networks currently encountered. Gossip algorithms [36] have received significant attention, since they facilitate network-based inference low complexity, high scalability, and high reliability. A profound review [7] discussed that, although the "efficiency" of gossip protocol is controversy, its bright side is still evident: it is still a strong adjunct to the classical distributed approaches, in parallel with large social structures (e.g., mobiles, chats, friend relationship, etc.,), and completely decentralized.

More importantly, there has been a decent amount of research on applying machine learning techniques with gossip protocols. A collection of gossip-based machine learning methods are already investigated, such as K-means Clustering, Gassian Mixture model, Principal Component Analysis, Support Vector Machine, Regression, and Stochastic Optimization etc., [6], [15], [16], [19]–[22], [39], [40], [42]. This fact demonstrates the positive potential of merging machine learning techniques into the fully distributed environment using gossip. The fact that most methods are built on top of gossip averaging motivates this research.

Gossip algorithms for distributed averaging enables each node to communicate with its neighbors in the graph and exchange data. It has been proved that a variety of gossip algorithms converge exponentially for aggregating data, under certain conditions [10]–[12], [36], [44].

There are two main approaches to gossip algorithms. One manner, which we call random, proceeds by having each network node sharing data with a random single other node at each step. For randomized gossip, we have Push-pull gossip (PPG) and Push-sum gossip (PSG). The other main approach, weighted broadcast, proceeds by having nodes share data with all its network neighbors in a weighted fashion. In the paper, we consider Sum-weight gossip (SWG) for this approach.

Although the theoretical principles of (synchronous) gossip averaging algorithms are well known, there has no systematic empirical comparison of the performance of them. Let alone the study of comapring the asynchronous algorithms and the them against the graph properties. We address that need in this article. We conduct our empirical study by examining the performance of the three selected gossip algorithms in asynchronous mode. The simulations are carried out on a random complex network topology, scale-free (Barabási-Albert) graph [1], [4], [8], which is a degree-biased graph topology staying close to real-world structures.

The paper presents the following contributions of our own:

- We conduct the empirical comparison between the three algorithms on scale-free graphs, with regard to the computation accuracy, rounds, and messages transmitted in the network. We also propose a metric to measure the quantity of rounds that do not contribute to the global convergence. The finding shows that all algorithms achieve little approximation error, although PSG has a low probability to perform not sufficiently well. SWG uses the least rounds to converge in all cases; however, the number of rounds for PSG is close to that for PPG falls dramatically as the graph mean degree grows.
- This paper demonstrates that the performance of each algorithm can be affected by several graph features to different extents.
- We propose adjusted practical algorithms for PPG and PSG, as well as a detailed discussion.
- We revise the proof for PPG in [10]–[12] to remove the constraint that the diffusion matrix for the protocol is symmetric. Meanwhile, we also prove the round complexity of asynchronous PSG and SWG.

A. Outline

The rest of the paper is organized as follows. Section II reviews the previous work. Section III defines the problem and elaborates on the notation. Next, Section IV details the protocols and algorithm, and Section V depicts the implementation concerns and the modifications to the algorithms. Later, we describe the network topology in Section VI, and discuss the experimental study in Section VII, Section VIII concludes the paper. Last, the theoretical study lies in Appendix.

II. Related Work

It is essential to examine the performance of the Gossip protocol against the network topologies that can be used to model a variety of real-world systems. Scale-free graphs have the properties close to what a few real-world topologies maintain, such as the World Wide Web [2], metabolic networks [34], and trust cooperation networks [26], etc.

Despite the lack of the empirical comparison between the gossip protocols for distributed averaging, there has been some work for comparing the information dissemination using gossip. [23] compared the message and communication step complexity (round complexity) between a few probabilistic gossip strategies for one single gossip algorithm in the random graph [17] and scale-free graph. In their algorithm, each node uses a probabilistic switch to control if to gossip once it is awaken. This work proposed a strategy in which the high-degree sites send messages in a higher probability than low-degree sites. This approach works better than the common

approaches with respect to the time and message complexity. They also empirically demonstrated that the convergence speed is faster starting in a *hub* node other than in a random node. Hu et al. [29] denoted a few metrics *effectual fanout*, *message complexity*, *reliability*, and *latency* to compare three types of diffusion-oriented gossip algorithms on three network topologies, which are respectively: random graph, random geometric graph, and scale-free graph. [27], [30] proposed a strategy that all the nodes send messages to the high-degree neighbors with a higher probability, and tested it on scale-free graphs. Unfortunately, the above gossip algorithms are merely for message spreading.

For the gossip algorithms tackling *distributed averaging*, Boyd's group provided the theoretical framework for PPG [10]–[12]. When proposing PPG, [33] conducted a fairly detailed empirical study for PPG, including scale-free graphs however the mean degree was limited to 20. [31], [32] presented both theory and simulations for SWG. However, the work missed a detailed description of the experiments and topology configuration. [3] proposed a broadcast-based gossip averaging algorithm. We do not consider this algorithm as it does not guarantee the convergence to the global mean [3], [31], [32].

For the asynchronous gossip algorithms, [24], [25] analyzed the complexity of the asynchronous gossip protocols for the message dissemination. [9] discussed the difficulty of the convergence of an asynchronous gossip algorithm that is not in the selected algorithms in the paper. Besides, there also exists a few research of utilizing the asynchronous gossip for machine learning techniques only [21], [37], [42]. Given the number of gossip-based applications, we feel it neccessary study its underlying framework.

To address the aforementioned shortcomings, we are about to carry out a novel study on the three well-known gossip algorithms that computes the distributed mean, more precisely, in an asynchronous mode. We compare the three algorithms across a set of graphs of a variety of graph properties. Plus, we analyze the time complexity for the asynchronous PSG and SWG, and show the proof. For PPG, we add a simple proof that removes one constraint from what exists in the proof made by [10]–[12].

III. Preliminaries

a) Distributed Averaging Problem: Consider a network that reflects an undirected graph $G = (V, E)$, where V and E denote the vertex set and edge set respectively. Let $\mathbf{x} = \{x_i\}_{i=1\dots n}$, where x_i is the local value held by node i, s.t. $i \in V$. The objective is to compute the distributed mean of the node values $\overline{x} = \frac{1}{n}\sum_i x_i$.

b) Notation: Let n be the number of vertices/sites/nodes, namely the graph order or the graph size, of the network G, s.t. $n = |E|$. Each node also holds a weight w, s.t. $\mathbf{w} = \{w_1, \dots, w_n\}$. Any two nodes i and j are thought to be neighbors if and only if $(i, j) \in E$, and $\mathcal{N}_i$ denotes the neighbor set of node i. We denote the mean degree by $\overline{V}$ where $\overline{V} = \frac{1}{n}\sum_i |\mathcal{N}_i|$. Moreover, let i^* denote that node i is awaken, and let $i \to j$ indicate i sends a message to j.

We denote the transition matrix by $\mathbf{P} = \{p_{i,j}\}_{i,j=1\dots n}$, where $p_{i,j}$ denotes the probability of i contacting j and $p_{i,j} = 0,\ \forall (i, j) \notin E$. Additionally, we denote the weight for transmitting the local data from i to j by $k_{i,j}$. $\mathbf{K} = \{k_{i,j}\}_{i,j=1\dots n}$ is also known as the diffusion matrix, e.g., $\mathbf{x}(t)^{\mathrm{T}}\mathbf{K}(t) = \mathbf{x}(t+1)^{\mathrm{T}}$. Let us also denote the product of the diffusion matrix by $\mathbf{K}(0:t)$, where $\mathbf{K}(0:t) = \mathbf{K}(0)\dots\mathbf{K}(t)$. Let τ denote the stopping condition threshold.

For the expression, we use $\mathbb{1}$ for the indicator function, with the predicate in its subscript. Let $\|\cdot\|^2$ denote the square of vector norm, s.t. $\|\mathbf{x}\|^2 = \mathbf{x}^{\mathrm{T}}\mathbf{x} = \sum_i x_i^2$. Let us also denote the second largest eigenvalue by λ_2. Given a set of samples, we present its mean a with the lower and upper bound b and c for the corresponding .95 confidence interval (CI), by the format "$a\ (b, c)$". Last, ϱ denotes a positive number smaller than 1, and $\lambda_2(\cdot)$ is the second largest eigenvalue of the input matrix.

IV. Gossip Protocol

In the gossip protocol, each node in the network communicates with its neighbor(s) to exchange data. This paper mainly focuses on the distributed averaging problem and the three algorithms for solving it. Regarding the diffusion strategy per round, PPG allows a node to exchange with merely one neighbor per action. In PSG, each node sends a message to a randomly chosen neighbor without waiting for the response, whereas SWG holds a transition matrix that guides each node to disseminate around to multiple nodes [5], [31], [36].

There are various criteria for categorizing the gossip algorithms. In terms of the diffusion strategy, one can classify PPG and PSG as binary gossip, and label SWG as broadcast gossip. On the other hand, PSG and SWG can also be one family as they both are one-way pushing and weight-based averaging protocols, whilst PPG is a two-way transmission protocol. Consequently, we will describe them separately in the following sub-sections.

Prior to the details, we introduce *mass conservation* [31], [36], which is a vital proposition for the three algorithms.

Proposition 1 (Mass Conservation)**.** *At any time t during the process, the sum of the values in the network remains unaltered and the weights sum up to n. That is, $\forall t : \sum x(t) = n\overline{x}$ and $\sum w(t) = n$ hold.*

A. Push-Pull Gossip

In this approach, node i pushes one message to one arbitrary neighbor j each round. Meanwhile, each node keeps replying a PULL message to the PUSH sender. Therefore, its diffusion matrix keeps changing over at all times, as one node selects one random neighbor to proceed at each round. However, at any t, there is

$$\mathbf{x}(t+1)^{\mathrm{T}} = \mathbf{x}(t)^{\mathrm{T}}\mathbf{K}(t) = \mathbf{x}(0)^{\mathrm{T}}\mathbf{K}(0:t)\,. \tag{1}$$

Note that, the symmetricity of $\mathbf{K}$ enables that $\mathbf{x}(t) = \mathbf{K}(t)\mathbf{x}(t)$. Right stochasticity of $\mathbf{K}(t)$ for any t indicates

$\mathbf{K}(t)\mathbf{1} = \mathbf{1}$ is also valid. The vector $\mathbf{x}$ (at any time t) sums up to $n\overline{x}$.

$$\forall t: \begin{cases} \mathbf{x}(t+1)^{\mathrm{T}}\mathbf{1} = \mathbf{x}(t)^{\mathrm{T}}\mathbf{K}(t)\mathbf{1} = \mathbf{x}(t)^{\mathrm{T}}\mathbf{1} \\ \sum x_i(t) = \sum x_i(0) = n\overline{x}\,. \end{cases} \tag{2}$$

All node values converge to the global mean $\overline{x}$ eventually. The process should be considered as converged when $\exists t', \forall t \geq t', \mathbf{x}(t) \approx \mathbf{x}(t')$. Moreover, $\mathbf{K}(t)$ is doubly stochastic in this approach, since node i receives a message from neighbor j will send back a message to j. We then have $k_{j,i} = k_{i,i} = 0.5$.

1) Algorithm: PPG concurrently executes two processes - one is active for disseminating PUSH messages, and another is passively listening to the returning PULL messages. Algorithm 1 depicts the basic version of PPG [33], [43]. In the algorithm, RANDOM($\mathcal{N}_i$) returns a random neighbor of the node i, and the function CONVERGED($\cdot$) is for checking the convergence locally (Algorithm 5). Upon the receipt of a message, the node updates its local value by summing up the local value and the incoming value.

Algorithm 1 PUSH-PULL GOSSIP

At node i

procedure DIFFUSE
  $j \leftarrow$ RANDOM($\mathcal{N}_i$)
  send PUSH(x_i) to j
end procedure
procedure RECEIVE ▷ passive thread
  receive a message msg from j
  if $msg.status ==$ PUSH **then**
    send PULL(x_i) to j
  end if
  $x_i \leftarrow (x_i + msg.x_j)/2$
end procedure

a) Complexity: All nodes will converge to the global average after $O(\frac{\log \epsilon}{\log \varrho})$ rounds for some arbitrarily small positive value ϵ and some positive value $\varrho < 1$ (see [10]–[12]). Following the approach in [10]–[12], λ could be the second largest eigenvalue of the expected diffusion matrix, and has been proved to be strictly smaller than 1. Consequently the mean rounds in the process is then $O(\frac{\log \epsilon}{n \log \varrho})$. The message passing complexity is simply the same as its time complexity, as a node only passes one message to the neighbor per round. It is notable that, the complexity function is still a logarithmically increasing function and hence enables the scalability of the algorithm.

B. Push-Sum Gossip

PSG designs that each node i sends a message to a random neighbor, without waiting for any response message. Subsequently, it updates its value by summing up all incoming data. Notably, in one certain time slot, a node is possible to receive multiple messages.

In the protocol, each node holds a value x, and a weight w which will be initialized as 1. Instead of the value x, a node i will use the sum value s to exchange with neighbors, where $s_i(0) = x_i(0)$. Every node starts a round by sending the weighted data $\frac{1}{2}(w_i, s_i)$ to a randomly chosen neighbor j and itself Afterwards, the node updates its weight and value by summing up all the received weights and values, i.e., $w_i(t) = \sum_j w_j(t)\mathbb{1}_{t;j\to i}$. The estimate of $\mathbf{x}$ is defined as

$$\mathbf{x}(t) = \left\{ \frac{s(t)_1}{w(t)_1} \quad \dots \quad \frac{s(t)_n}{w(t)_n} \right\}. \tag{3}$$

Also, the following equations hold:

$$\mathbf{s}(t+1)^{\mathrm{T}} = \mathbf{s}(t)^{\mathrm{T}}\mathbf{K}(t) = \mathbf{s}(0)^{\mathrm{T}}\mathbf{K}(0:t) \tag{4}$$

$$\mathbf{w}(t+1)^{\mathrm{T}} = \mathbf{w}(t)^{\mathrm{T}}\mathbf{K}(t) = \mathbf{w}(0)^{\mathrm{T}}\mathbf{K}(0:t)\,. \tag{5}$$

$\mathbf{K}(t)$ is only row-stochastic in most cases. The algorithm initializes with $\mathbf{w}(0) = \mathbf{1}$, and $\mathbf{s}(0) = \mathbf{x}$. Analogously to Eq. (2) in PPG, we have

$$\forall t: \begin{cases} \sum x_i(t) = \sum \frac{s_i(0)}{w_i(0)} = n\overline{x} \\ \sum w_i(t) = n\,. \end{cases} \tag{6}$$

1) Algorithm: The algorithm was proposed in [36], and its underlying version is presented in Algorithm 2. At each round, a node i diffuses only one message, and sums its local s_i and w_i with all incoming data if there is any.

Algorithm 2 PUSH-SUM GOSSIP

At node i

repeat
  $j \leftarrow$ RANDOM($\mathcal{N}_i$)
  send $(s_i/2, w_i/2)$ to j and i
  receive $\{(s_j, w_j) : j \to i\}$
  $(s_i,\ w_i) \leftarrow \left(\sum_j s_j,\ \sum_j w_j\right)$
until CONVERGED()

2) Complexity: [36] showed that the average rounds is bounded in $O(\log n)$. However, since the same analysis technique in [10]–[12] can be applied to this algorithm, we believe that the the average rounds is also bounded by $O(\frac{\log \epsilon}{n \log \varrho})$. We prefer $O(\frac{\log \epsilon}{n \log \varrho})$, which characterize the algorithm more (refer to Section A-B2). The message cost is then intuitively identical to the rounds taken in the process.

C. Sum-Weight Gossip

SWG is a representative of weighted gossip. Each node i, in SWG, maintains a non-negative one-sum weighted adjacency vector $\mathbf{k}_i$, and broadcast the weighted data $k_{i,j}(w_i, s_i)$ to each corresponding neighbor j per action. [36] first suggested synopsis gossip - the prototype which was later coined "weighted gossip" [5]. Benezit et al [5] also proved that the algorithms based on the framework are guaranteed to converge to the global average value. We study the Sum-Weight Gossip (SWG) for this protocol, and present it in Algorithm 3 [31], [32].

We regard SWG as a generalization of PSG [36]. In particular, they share exactly the same properties as shown in Eqs. (3) to (6). The only difference is that SWG broadcasts messages while PSG disseminates only one message at each time. It is not mandatory for $\mathbf{K}(t)$ to be column-stochastic, although it must be right stochastic.

1) Algorithm: Each node i keeps a codebook $\mathbf{k}_i$ maintaining the share distribution over its neighbors. A node disseminates $|\mathcal{N}_i|$ messages, and then updates the local s and w exactly as how a node updates in PSG. Like the mechanism in PSG, every node will preserve a portion of data for itself when gossiping, i.e., $w_{i,i} > 0$.

Algorithm 3 SUM-WEIGHT GOSSIP

At node i

repeat
  for all $j \in \{j : k_{i,j} > 0\}$ **do**
    send $(k_{i,j}s_i,\ k_{i,j}w_i)$ to node j
  end for
  receive $\{(s_j, w_j) :\ j \to i\}$
  $(s_i,\ w_i) \leftarrow \left(\sum_j s_j,\ \sum_j w_j\right)$
until CONVERGED()

a) Complexity: The process will also stop after $O(\frac{\log \epsilon}{n \log \varrho})$ rounds. However, its number of messages is co-governed by the mean degree in the graph. The expected message complexity of the network is $O(\frac{\overline{V} \log \epsilon}{n \log \varrho})$ that can be significantly worse than the complexity of PPG and PSG.

V. IMPLEMENTATION ANALYSIS

We use this section to document the challenges we have encountered when implementing the algorithms. Section V-A is for discussing the problem of obtaining zero-error. Section V-B depicts the local convergence check and its necessity. With respect to each challenge, we comment on the algorithms facing the challenge and our solutions. We propose changes for PPG and PSG to handle practice as the discussion proceeds.

A. Accuracy Concerns

Theoretically, the errors of all the algorithms tend to 0 as the rounds tend to infinity, disregarding the message passing losses. However, the error may easily grow in real-world applications, due to the violation of *mass conservation*.

Message sending failure is a straight factor that distorts *mass conservation* and is more severe to PSG and SWG than to PPG. The failure downgrades the accuracy of PPG only when there are PULL message loss since PUSH message loss will not trigger the receipts to send back a PULL message. For such as case, the current round can be ignored. Assuming the number of lost messages are fixed, the message loss occurs later has a lower expected negative impact to the accuracy less than that occurs earlier in the computation The value in later rounds are closer to be converged value hence the discrepency, $|x_i - \overline{x}|$, will be relatively smaller. The loss of the mass is therefore more likely to be smaller than that occurs in early rounds.

For PPG, *mass conservation* will be distorted if a node i receives PUSH messages from other sites but node j, when it waits only for the pull message from node j. One can devise that the node i blocks itself and only accepts the PULL message from the node j after pushing to j. However, this design is likely to raise deadlocks during the process.

Algorithm 4 PRACTICAL PUSH-PULL GOSSIP

At node i

$busy \leftarrow false$ ▷ indicates if the node is busy
procedure DIFFUSE
  repeat
    $j \leftarrow$ RANDOM($\mathcal{N}_i$)
    send PUSH(msg_i) to j
    $busy \leftarrow true$
  until CONVERGED()
end procedure

procedure RECEIVE(msg) ▷ passive thread
  $j \leftarrow msg.sender$
  if $msg.status =$ BUSY **then**
    SLEEP(Δt) ▷ throttling
  else if $msg.status =$ PUSH $\ \&\ busy$ **then**
    send BUSY(msg_i) to j
  else
    if $msg.status =$ PUSH **then**
      send PULL(msg_i) to j
    else if $msg.status =$ PULL **then**
      $busy \leftarrow false$
    else
      raise error
    end if
    $x_i \leftarrow (x_i + msg.x_j)/2$
  end if
end procedure

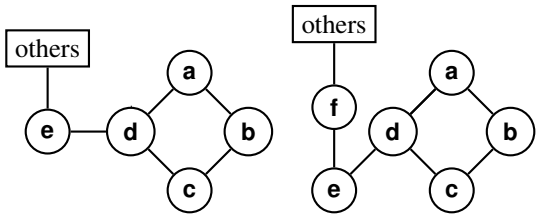
For instance, we assume there are three nodes $\{i, j, k\}$ such that $i \to j$, $j \to k$, and $k \to i$ happen simultaneously. It clearly forms a circular wait chain between the three nodes. Hence, in addition to the design above, our implementation consider using a BUSY message for notifying the latter PUSH sender(s) to seek a new target, as the receiver has already been waiting for a PULL message. A practical version of Push-Pull gossip (P-PPG) is laid out in Algorithm 4. This mechanism is not perfect as it will generate undesirable extra message communication; however, our empirical outcome show that the quantity added is acceptable. Fortunately, this problem does not arise in PSG and SWG, as they are one-way pushing protocols.

B. Local Convergence Concerns

Previous study generally suggested to run the gossip algorithms to a certain number of rounds. Given the theory, the algorithms are thought to be converged surely. However, we view this approach is not ideal for the real-world application, since we observe that the convergence rounds for each algorithm diversify. Also, in a distributed network, checking the local convergence complies more with the self-interested discipline for the agents.

Implementing the convergence check requires extra considerations. It is not ideal for a node to either stop too early or too late. Stopping too early may not lead to a convergence, while terminating too late may generate redundant rounds. The difficulty lies in that a local node possesses little knowledge

of the global convergence. We rule that, if the local value change is smaller than an extremely small value ϵ during τ rounds, the node is thought to be converged (Algorithm 5). However, even if the condition is met, it may be the case that the nodes in a cycle do not receive any updates from the outside, but exchange inside the cycle over τ times. In this case, the nodes in the cycle are not truly converged, although they seem to be. For example, Fig. 1a shows a case which may lead to quick local but not global convergence. Assuming at some point, nodes a, b, c, d, e have the same local value, the nodes in the cycle keep exchanging messages, but node e is not involved with certain probability. Therefore, node a, b, c, and d are likely to converge rapidly, if τ is small. In particular, more nodes are connected in such cycle, the quicker these nodes stop processing due to the local misjudging the global convergence. Fig. 1b depicts an even worse case, by which the updates from the others will be harder to be passed to the cycle.



(a) A bad case (b) An even worse case

Fig. 1. Local convergence examples that lead to a loss of mass

Algorithm 5 Local Convergence Check

At node i
$i.c$ is initialized to be 0 ▷ a local counter

function CONVERGED
  $i.c \leftarrow i.c + 1$ **if** $\left|\frac{x_i(t-1)-x_i(t)}{x_i(t-1)}\right| < \epsilon$ **else** 0
  return $true$ **if** $i.c > \tau$ **else** $false$
end function

Considering the reason, setting a larger τ straightforwardly diminishes the likelihood of the final misjudgment for a node. The idea behind is that if the probability of one single misjudgment of convergence for a node is around q, and then the probability of misjudgment will be q^τ. It implies that there is a trade-off between the accuracy and the stopping threshold τ, and we believe there must be other techniques that can apply. Nevertheless, we regard optimizing the stopping scenarios as an important future research topic.

We pay extra attention to PSG, as PSG is the most sensitive to the convergence out of the three algorithms. It is probable that at least one node remains not contacted in a certain period, due to the nature of the algorithm, especially in the real-world network which are degree-biased. Such a node will terminate as it deems itself "converged". To prevent this, the general solution is to enlarge the stopping threshold τ. One cannot simply cancel the convergence checking for the nodes not receiving anything. This raw method will occasionally hang the process, as one node's neighbors may all have terminated but itself remains active. It is also necessary to add extra sleeping time for each node that receives nothing. That is, sleeping enlarges the chance for a node to receive a message in its next round. The practical algorithm is detailed in Algorithm 6. There is an option to prevent the deadlock that each node sends the message to notify the neighbors that it has converged. However it adds a lot more message cost, and hence not considered at the stage.

Algorithm 6 PRACTICAL PUSH-SUM GOSSIP

At node i

repeat
  $j \leftarrow$ RANDOM($\mathcal{N}_i$)
  send $(s_i/2, w_i/2)$ to j and i
  $\mathcal{M} \leftarrow \{(s_j, w_j) : \ j \rightarrow i\}$
  if $\mathcal{M} \neq \varnothing$ **then**
    $(s_i, \ w_i) \leftarrow \left(\sum_j s_j, \ \sum_j w_j\right)$
  else
    SLEEP(Δt)
  end if
until CONVERGED()

VI. BARABÁSI-ALBERT GRAPH

To explore more opportunities for gossip-based applications, it is crucial to examine the performance of the algorithms in the networks that are close to the real-world topologies. Albert and Barabási [1] empirically demonstrated that most large topologies in real-world are scale free and their degree distribution follows a power law for large mean degree k. Hence, we study the scale-free (Barabási-Albert) graph [4]. Let us denote the graph by $\mathcal{B}(n, m)$, where n denotes the number of nodes, and m is the number of edges that each new node should connect to the existing graph with.

The Barabási-Albert model has two ingredients, *growth* and *preferential attachment*. It initializes with a *clique* containing a small number of nodes, say m_0, where $m \leq m_0 \ll n$. A *clique* is a complete subgraph relative to the entire graph. Moving forward, it links a new node to $m(\leq m_0)$ existing nodes in the existing graph repeatedly. A new node will be allied with node i with a probability Π dependent on the degree d_i, such that

$$\Pi(d_i) = \frac{d_i}{\sum_j d_j}$$

Polynikis [41] also claimed that $\overline{V} \approx 2m$, and $\overline{V}$ does not depend on the number of nodes n in the graph.

VII. EMPIRICAL STUDY

In this section, we first discuss the experimental setup and comment on the empirical findings.

A. Data

The simulations evaluate the algorithm performance on two sets of synthetic data and one real-world dataset. The two sets of synthetic data are generated from the Gaussian distributions regarding a range of graph size. They are

TABLE I
THE (MEAN, STANDARD DEVIATION) PAIRS OF THE DATASETS

	Graph Size					
Dataset	200	400	600	800	1000	5000
N1	(101.73, 1054.74)	(104.08, 954.65)	(104.76, 1011.62)	(106.88, 1028.75)	(99.54, 967.69)	(100.39, 1007.12)
N2	(112.48, 96.79)	(97.78, 97.03)	(95.00, 103.85)	(103.55, 102.44)	(101.64, 101.25)	(100.76, 100.70)
YT	(8.93, 28.88)	(5.70, 19.48)	(7.90, 31.35)	(9.43, 68.00)	(7.70, 52.57)	(8.59, 68.66)

1) N1: drawn from $\mathcal{N}(10^2, 10^6)$,
2) N2: sampled by $\mathcal{N}(10^2, 10^4)$.

We are interested in examining the performance of and the comparison between algorithms, hence consider the anonymized YouTube friendship data (YT), provided by [38]. YouTube is an active video sharing website wherein the users can connect together or simply interact with others, etc. Since the entire dataset is large, we sample the data to fit each simulation scenario.

B. Graphs

The experiments are repeated on many random graphs with various graph orders and graph parameters, generated by the Python NetworkX package [28]. For each graph size and each parameter setting, we create 5 random graphs for each single set of parameter configuration. E.g., we randomize 5 graphs for the setting $n = 200$ and $w = 10$ for the scale-free graphs. A graph will be re-generated if it is a disconnected graph.

C. Measurement Metrics

1) Waste Round: By the nature of the algorithms, we easily notice that there must be certain rounds that do not raise any local value change towards the convergence. The principle of this metric is indeed to reflect the inefficiency. Thus, the rounds for reaching τ are also counted as waste rounds, as they comply with "inefficiency". It also represents a round that wastes the computational energy. Hence, we define a term *waste round*, for further understanding the effectiveness of the algorithms

Definition 1 (Waste Round)**.** *In any averaging gossip algorithms, a waste round for a node is a round during which a node has no value change after updating with the incoming messages.*

It is worth noting that the rounds for deciding if it is approaching convergence will be also counted, as we found that the stopping thresholds vary for maintaining the high accuracy of different algorithms. Therefore, we penalize those ones which demand a larger τ. Our finding shows that the number of waste rounds are not ignorable in both PPG and PSG, while SWG is solely slightly affected. Generally, we always desire to obtain a solution that solves distributed problems efficiently in energy consumption, computation and transmission. This is a metric that allows us to further optimize the algorithms in the future.

2) Performance Metrics: This paper sticks to the simple but instructive metrics, which are shown as follows.

- *mean rounds* ($\mathcal{R}$): the mean rounds for reaching the consensus over all nodes
- *mean waste rounds* ($\mathcal{W}$): the waste rounds per node during the process
- *mean messages* ($\mathcal{M}$): the average messages sent by each node
- *mean least absolute percent error* ($\mathcal{L}$):

$$\mathcal{L} = \frac{1}{n} \sum_i \left| \frac{x_i - \overline{x}}{\overline{x}} \right|$$

The number of rounds and messages are associated with the time and message efficiency respectively. We use $\mathcal{L}$ instead of mean least squares error over all nodes which was used in the theoretical analysis in Appendix, as minimizing them are equivalent in this task. Choosing $\mathcal{L}$ is for better illustration in the figures. Finally, we would like to clarify that, for instance, $\mathcal{R}$ is the mean rounds over all nodes, whilst $\overline{\mathcal{R}}$ is the mean value of the mean rounds with respect to multiple simulations.

In regard to the graph properties, we focus on the following ones:

- *degree* (V): we consider the mean, variance, minimum, and maximum of degree, respectively denoted by $\overline{V}$, $Var(V)$, $Min(V)$, and $Max(V)$.
- *clustering coefficient* (C): the fraction of possible edges linking the node's neighbors through that existing edges between the node's neighbors. It will be studied with the mean, variance, minimum, and maximum. It uses the identical naming convention as that for V.
- *eccentricity* (E): the maximum graph distance of reaching any node from any distinct node in a connected graph. We consider the mean, variance, minimum (namely radius) and maximum (diameter).
- *density* (DEN): the rate of number of edges over possible number of edges.

, For more details, please refer to [18], [46]

D. Sensitivity Analysis

Fig. 2 shows the sensitivity analysis of the stopping condition τ. Note that we denote the graph of order 200 by $G(200)$, in the rest of the paper. The plots illustrate the .95 CI of $\mathcal{L}$ for each algorithm running on the three datasets for $\mathcal{B}(200, 10)$, given a range of the thresholds. Since $m = 5$ may be too extreme, we stick to the graph with 200 nodes and $m = 10$. We found that the numeric scales and the trends for each dataset

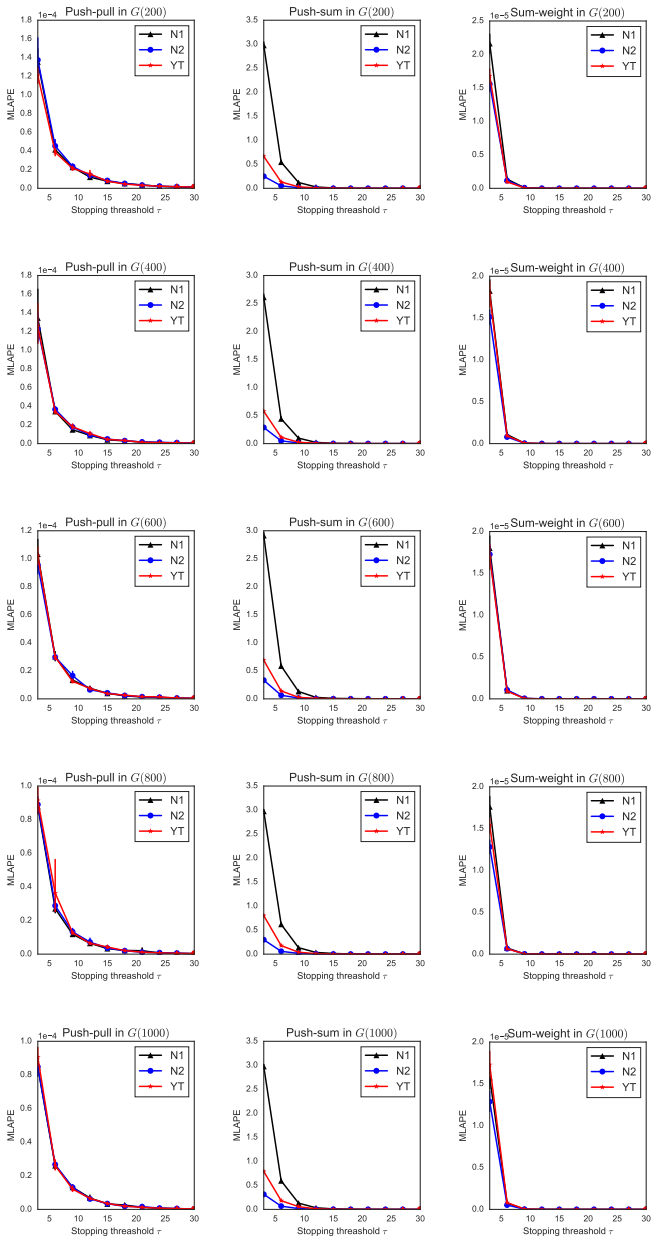


Fig. 2. Sensitivity Analysis for Stopping Threshold

and graph size are very similar. Hence, we show the analysis on merely one graph of each graph order.

Given $\tau = 3$, PSG shows a random error level with respect to different graph size. We see that the errors drop dramatically as τ increases. Certain cases show that in a 10^{-4} level, a larger threshold for PSG may not be strictly decreasing. However, PPG and SWG obtain extremely low error within 10^{-4} level in decreasing order as τ enlarges. Compared to others, PSG is the most stopping condition sensitive algorithm. All errors approach 0 as τ grows; hence, it implies that $\lim_{\tau\to\infty} \mathcal{L} = 0$ regarding every algorithm and graph.

As a result, we select $\tau = 3$ for PPG and SWG, and $\tau = 24$ for PSG for retaining an error less than 10^{-3}. These values are decided from the worst cases among all sensitivity analysis simulations.

E. Empirical Result

1) Accuracy: This section is used to review the accuracy of the three algorithms. We mark that median is more suitable than mean for demonstrating the accuracy, since the most of $\mathcal{L}$ are close to 0 and the "outlier" outcomes will make the overall performance look terribly bad. Moreover, in comparing the gossip algorithms, the likelihood of getting unexpected outcome is more important and expressive the accuracy value. However, we still show the mean of $\mathcal{L}$ in the Supplementary Document (SD). We conclude that all three algorithms can have exceedingly accurate approximation to the global mean.

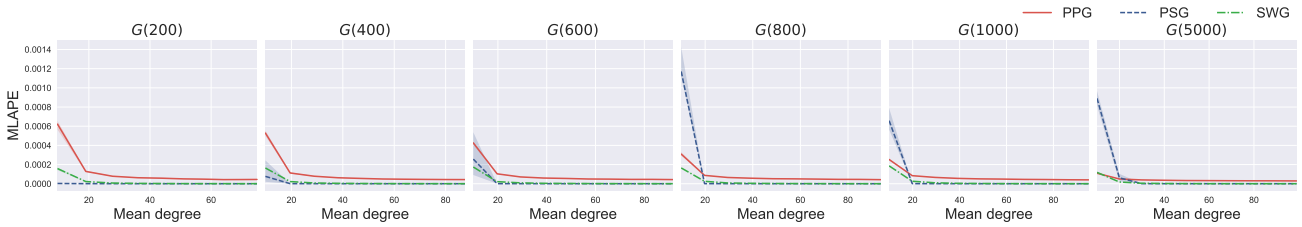
Fig. 3 depicts the median of $\mathcal{L}$ of the algorithms against the mean degree in any graph size. The plots are with .95 CI. One can see that $\mathcal{L}$ for each algorithms are below a certain low level, and decreases approximately exponentially. With the median metric, PSG has the lowest $\mathcal{L}$. However, PSG has the most unstable $\mathcal{L}$ when considering the mean instead of median, shown in the SD. Nonetheless, SWG is always a very solid performer in regard to accuracy.

There exists a slight rise for PPG around mean degree 65 in $G(200)$ with data N1. We believe that all the performance metrics of these randomized algorithms are co-affected by multiple factors rather just than just mean degree in the graph.

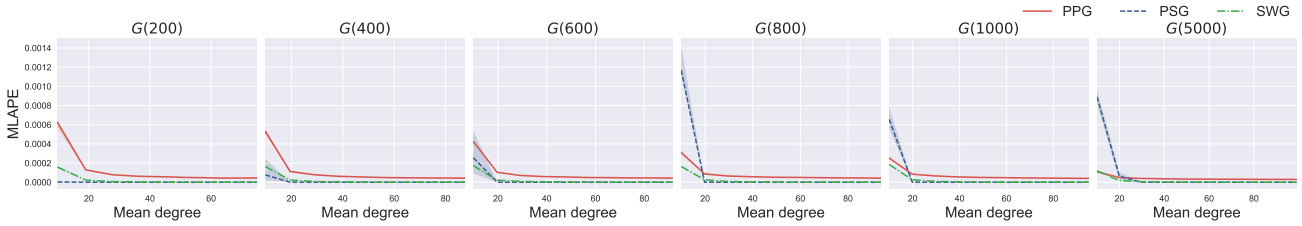
2) Round Consumption: In this section, we look into the rounds required for each algorithm to reach the consensus. The .95 CI of the rounds for each algorithm against various graph size and mean degrees are shown in Fig. 4. Our experiments depict that SWG has consistently low round usage, PSG is better than PPG when the graph size is smaller than 1000, and PPG have better round efficiency once the graph is larger.

The comparison shows that for smaller graph and worse connectivity, SWG has the least executing rounds while PPG requires way more convergence rounds. It also shows that as the graph size grows, the number of rounds needed for PPG to converge is getting close to that for PSG, and slightly better than PSG in $G(1000)$, with mean degree larger than 80. In the case of $G(5000)$, PPG is apparently the best performer with extremely small error, low time and low message cost The separate round plots for PSG and SWG are shown in supplementary material, and demonstrate that the rounds both need for convergence decline exponentially by the rise of the average number of the neighbors in the graph.

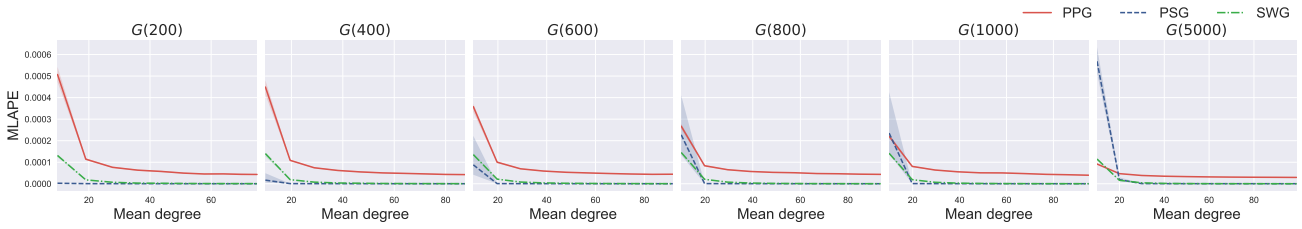
The mean waste rounds of each process are displayed in Fig. 5. It shows that the $\overline{W}$ for PPG and PSG drop exponentially over the graph mean degree, following their patterns of $\overline{R}$. We realize that the mean waste rounds of PSG reaches a high number > 40 in all of our simulations, which accounts for about 70% of the total mean rounds. This implies the improving potential of the algorithm. We also find that the mean waste round rate (Fig XXX in supporting files) of SWG lifts exponentially however the number keeps around 3-ish constantly but the rounds also reduces in an exponential trend. It is close to τ for SWG - which indicates that round wasting is not significant in SWG.



(a) YT



(b) N1



(c) N2

Fig. 3. Accuracy performance

3) Message Cost: Fig. 6 is used to display the comparison of the mean messages for each node transferred within various processes. In summary, PSG and PPG consume much less messages for convergence relative to SWG. In the worst case throughout all simulations, SWG disseminate 10 times more messages than PSG and PPG. Even though, its best case is that SWG and PPG diffuse close number of messages, and this usually happens when the mean degree in the graph is small. However, the messages used for PSG and PPG are decreasing as the graph degrees enlarge. Not surprisingly, the message usage of SWG scales against $\overline{V}$ of the graph - we have already known that its expected complexity is $\overline{V}\log(y)$ for some $y > 1$.

We use the yellow curve (labeled as Push-pull effective) in each figure to present the message quantity ignoring the BUSY messages, which is only counted in PPG. For the simulations in $G(5000)$, the mean effective messages of PPG is lower than the mean messages of PSG; however, $\overline{M}$ of PPG is slightly higher than that of PSG. Although in the simulations on $G(5000)$, PPG is more time efficient than PSG, PSG is slightly better with regard to the message complexity. The mean of the amount of BUSY messages is greater given larger graphs, even though it is not strictly increasing when the mean degree enlarges in the graphs.

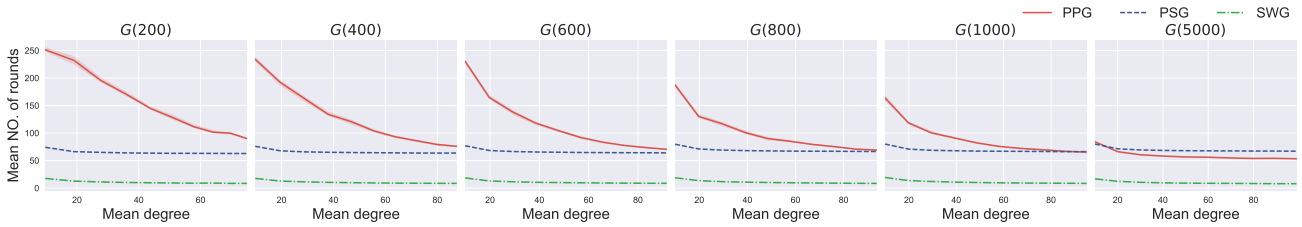
The illustration implies that our modification for retaining the *mass conservation* is fair as it only adds a limited number of messages. Despite of the increase of the BUSY message quantity, the mean rounds still decrease.

F. Graph Properties

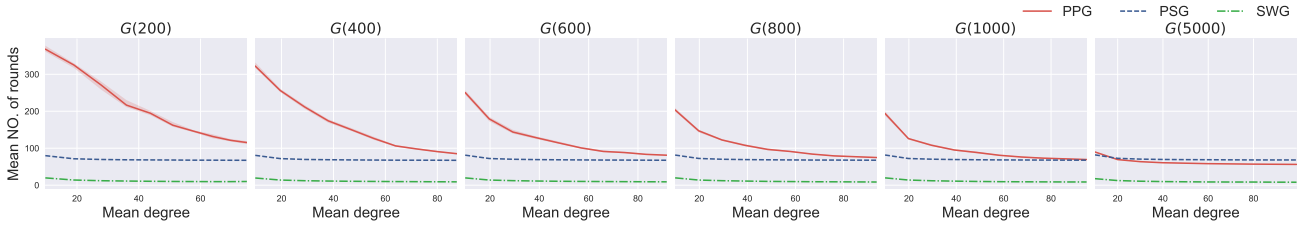
In this section, we denote a metric instability rate by $\mathcal{I}$, for the rate of the simulations having $\mathcal{L} > 10^{-4}$. As we notice that $\mathcal{L}$ is only either arbitrarily bad or exceedingly good, we believe the rate of getting bad accuracy is more valuable. In summary, the metrics associated with degree and clustering coefficient are more correlated to a range of performance metrics.

Correlation Coefficient is always employed to statistically evaluate the dependence between two variables. Pearson Correlation is used to measure the linear dependence of two variables using the numeric values, whilst Spearman Correlation applies the rank of the values of two variables to evaluate their monotonic relationship [14]. We utilize them to display the relationship between the graph properties and the performance. Two variables gains the strongest correlation when the absolute correlation value hits 1, and are thought to be linearly/monotonically independent given correlation 0. A negative correlation value implies that the two variables affect each other in an opposite direction.

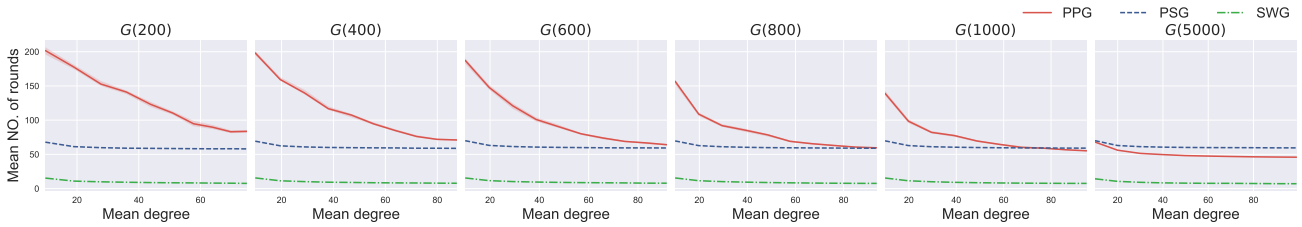
Unfortunately, we cannot draw a conclusion stating that certain variables are the most influencing factors for all the algorithms. Apart from that, the same variable would have different linear and monotonic correlations to those measurement metrics. However, in general, properties about node and eccentricity are important to PPG. The properties related to node, eccentricity and clustering are all factors to PSG, although most of them are weak elements. Finally, SWG is affected by the same kinds as PSG does. The difference is SWG is



(a) YT



(b) N1



(c) N2

Fig. 4. Mean round performance

controlled by node-related properties more significantly than that of cluster-realted, while PSG is the opposite.

a) PPG: The outcome for PPG is shown by Table II. For the mean rounds, larger number of nodes may result in less rounds for convergence. The situation of $\mathcal{M}$ and $\mathcal{W}$ just follow the pattern of $\mathcal{R}$. Also, Graph density and mean clustering coefficient have both small linear and monotonic correlation to $\mathcal{R}$ and $\mathcal{M}$. Besides, the waste rounds and the instability rate will be lowered when the graph has lower degree variance. The variance clustering coefficient is the most influencing factor that have the positive correlations to the four measurements. This fact supports the examples in Fig. 1.

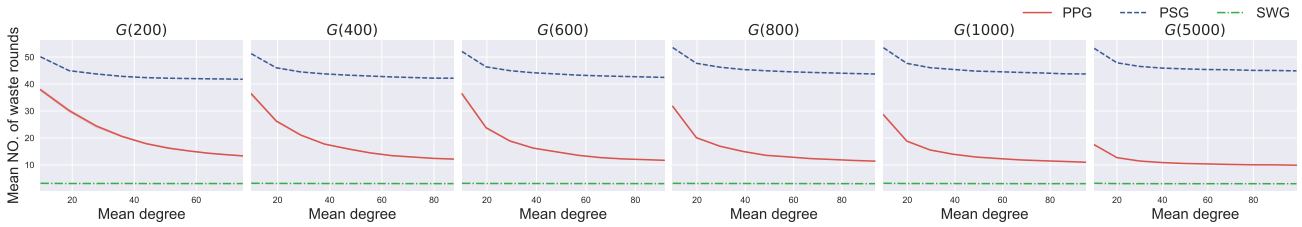
b) PSG: Table III depicts that mean rounds and mean messages are just weakly correlated to the degree, eccentricity, and clustering coefficient, either linearly or monotonically. The quantity of mean waste rounds can be diminished by having higher mean clustering coefficient and higher density. However, PSG may generate more waste rounds in a graph of greater mean eccentricity. With respect to the accuracy, $\mathcal{I}$ can be increased due to a low minimum degree or connectivity. It is intuitive because any node converges too quickly distorts the *mass conservation*, and the node with small degree tend to converge earlier than others. Interestingly, $\mathcal{I}$ demonstrates a relatively strong monotonic dependence to $\overline{C}$ (-0.68), however is almost linearly independent to $\overline{C}$ (-0.02).

We also learn that the factors leading bad performance are all related to eccentricity, and that for the positive outcome stays with minimum clustering coefficient and minimum degree. As PSG is an one-way and single pushing algorithm, it is sensible that $Min(E)$ is a dominant feature since eccentricity computes the distance of spreading information from one node to any others. The mean eccentricity has a relatively strong correlation (0.71) to the accuracy in a negative way, i.e., the larger $\overline{E}$ is, the worse the accuracy PSG could obtain.

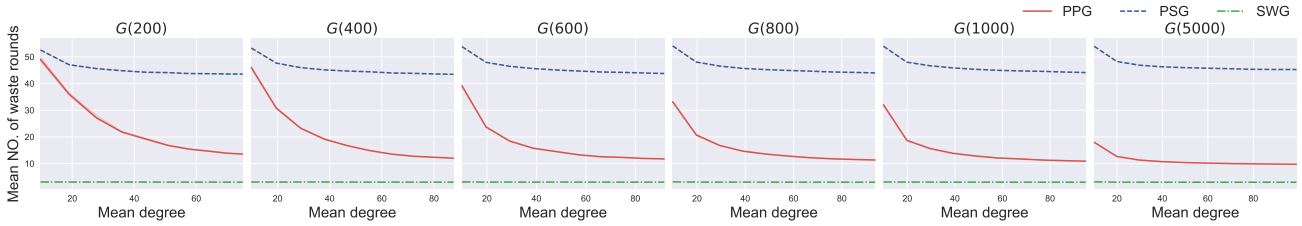
TABLE II
CORRELATION OF GRAPH PROPERTIES AND PERFORMANCE FOR PPG

	Pearson				Spearman			
	$\mathcal{R}$	$\mathcal{M}$	$\mathcal{W}$	$\mathcal{I}$	$\mathcal{R}$	$\mathcal{M}$	$\mathcal{W}$	$\mathcal{I}$
DEN	-0.01	-0.03	-0.14	-0.09	0.07	0.05	-0.05	-0.07
$\overline{V}$	-0.60	-0.60	-0.71	-0.57	-0.72	-0.73	-0.83	-0.81
$Var(V)$	-0.64	-0.63	-0.70	-0.55	-0.82	-0.82	**-0.90**	**-0.91**
$Min(V)$	-0.57	-0.57	-0.66	**-0.63**	-0.66	-0.66	-0.75	-0.83
$Max(V)$	**-0.71**	**-0.69**	**-0.73**	-0.58	**-0.85**	**-0.84**	-0.88	-0.86
$\overline{E}$	0.12	0.14	0.26	0.18	0.20	0.23	0.34	0.34
$Var(E)$	-0.36	-0.35	-0.42	-0.32	-0.38	-0.37	-0.40	-0.37
$Min(E)$	-0.15	-0.14	-0.05	-0.05	-0.14	-0.12	-0.04	0.04
$Max(E)$	-0.07	-0.06	0.03	0.01	-0.08	-0.06	0.02	0.05
$\overline{C}$	0.01	0.00	-0.12	-0.07	0.06	0.04	-0.07	-0.10
$Var(C)$	**0.76**	**0.75**	**0.82**	**0.74**	**0.84**	**0.82**	**0.86**	**0.83**
$Min(C)$	-0.15	-0.16	-0.28	-0.19	-0.21	-0.23	-0.35	-0.30
$Max(C)$	0.35	0.34	0.31	0.27	0.54	0.52	0.52	0.59

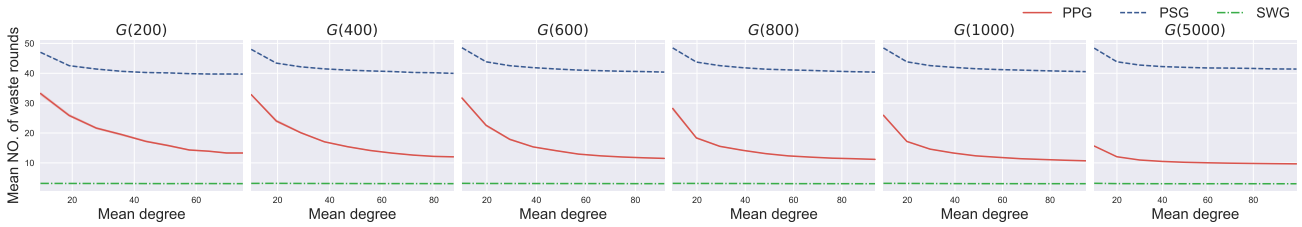
c) SWG: Last, we use Table IV to present the result for SWG. The time efficiency can be upgraded when the graph mean degree grows. The number of messages is greatly related to mean degree with regard to both Pearson and Spearman correlation. That is, despite the great time efficiency, the



(a) YT



(b) N1



(c) N2

Fig. 5. Mean wast round performance

TABLE III
CORRELATION OF GRAPH PROPERTIES AND PERFORMANCE FOR PSG

	Pearson				Spearman			
	$\mathcal{R}$	$\mathcal{M}$	$\mathcal{W}$	$\mathcal{I}$	$\mathcal{R}$	$\mathcal{M}$	$\mathcal{W}$	$\mathcal{I}$
DEN	-0.30	-0.30	-0.50	0.02	-0.36	-0.36	-0.54	-0.65
$\overline{V}$	-0.30	-0.30	-0.51	-0.27	-0.31	-0.31	-0.48	-0.55
$Var(V)$	-0.17	-0.17	-0.31	-0.28	-0.22	-0.22	-0.33	-0.41
$Min(V)$	-0.23	-0.23	-0.40	**-0.55**	-0.26	-0.26	-0.39	**-0.70**
$Max(V)$	-0.03	-0.03	-0.08	-0.12	-0.06	-0.06	-0.11	-0.13
$\overline{E}$	**0.33**	0.33	**0.55**	0.11	**0.39**	**0.39**	**0.58**	**0.71**
$Var(E)$	-0.08	-0.08	-0.17	0.30	-0.06	-0.06	-0.14	-0.02
$Min(E)$	0.25	0.25	0.38	0.36	0.27	0.27	0.34	0.54
$Max(E)$	0.27	**0.27**	0.42	**0.35**	0.29	0.29	0.38	0.57
$\overline{C}$	-0.32	-0.32	-0.51	-0.02	-0.36	-0.36	-0.54	-0.68
$Var(C)$	0.09	0.09	0.18	-0.03	0.07	0.07	0.12	0.05
$Min(C)$	**-0.32**	**-0.32**	**-0.53**	0.05	**-0.39**	**-0.39**	**-0.59**	-0.64
$Max(C)$	-0.12	-0.12	-0.18	0.16	-0.09	-0.09	-0.14	-0.14

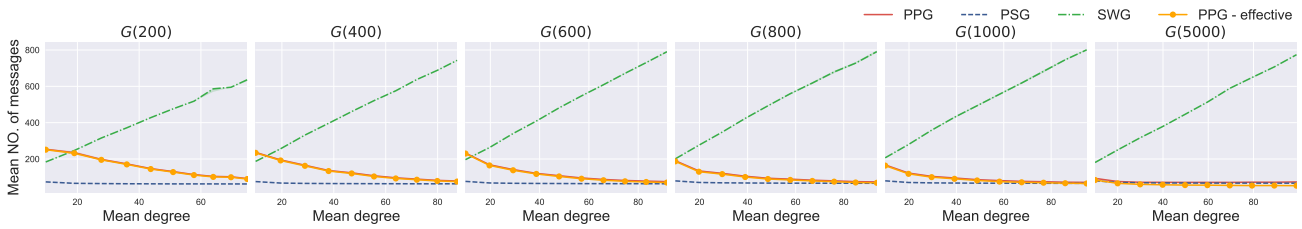
TABLE IV
CORRELATION OF GRAPH PROPERTIES AND PERFORMANCE FOR SWG

	Pearson				Spearman			
	$\overline{\mathcal{R}}$	$\overline{\mathcal{M}}$	$\overline{\mathcal{W}}$	$\mathcal{I}$	$\overline{\mathcal{R}}$	$\overline{\mathcal{M}}$	$\overline{\mathcal{W}}$	$\mathcal{I}$
DEN	-0.31	0.33	0.19	0.00	-0.28	0.34	0.38	0.00
$\overline{V}$	**-0.84**	**0.98**	-0.53	0.00	**-0.87**	**0.97**	-0.58	0.00
$Var(V)$	-0.61	0.73	-0.62	0.00	-0.76	0.82	-0.75	0.00
$Min(V)$	-0.76	0.91	-0.52	0.00	-0.80	0.90	-0.51	0.00
$Max(V)$	-0.42	0.43	**-0.71**	0.00	-0.44	0.45	**-0.79**	0.00
$\overline{E}$	**0.52**	**-0.59**	-0.11	0.00	**0.52**	**-0.61**	-0.15	0.00
$Var(E)$	-0.14	0.10	-0.10	0.00	-0.03	0.04	0.01	0.00
$Min(E)$	0.48	-0.51	-0.09	0.00	0.47	-0.52	-0.07	0.00
$Max(E)$	0.47	-0.53	-0.11	0.00	0.47	-0.54	-0.11	0.00
$\overline{C}$	-0.30	0.33	0.25	0.00	-0.29	0.36	0.38	0.00
$Var(C)$	0.45	-0.43	**0.73**	0.00	0.44	-0.43	**0.79**	0.00
$Min(C)$	-0.41	0.45	0.04	0.00	-0.56	0.63	0.05	0.00
$Max(C)$	0.14	-0.10	0.54	0.00	0.20	-0.17	0.65	0.00

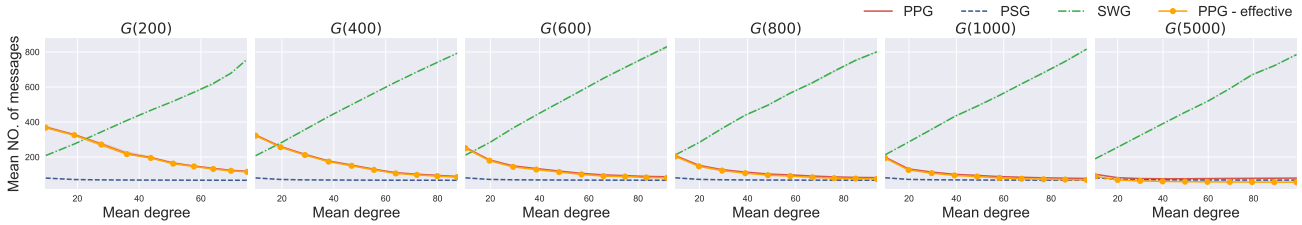
message numbers of node i is dominated by $|\mathcal{N}_i|$, particularly when $|\mathcal{N}_i|$ is large. Therefore, there exists a trade-off between the time and message complexity in this algorithm. Again, the variance of clustering coefficient is also factor that probably lead to more waste rounds, although the exact number of $\mathcal{W}$ is only slightly greater than 3. We can see that with $\mathcal{L} > 10^{-4}$ does not raise a positive instability rate (all are 0). Given that threshold, the correlation to other variables are all 0, which shows that it is totally independent on other variables. This resonates with the fact that SWG achieves negligible error constantly.

VIII. CONCLUSION

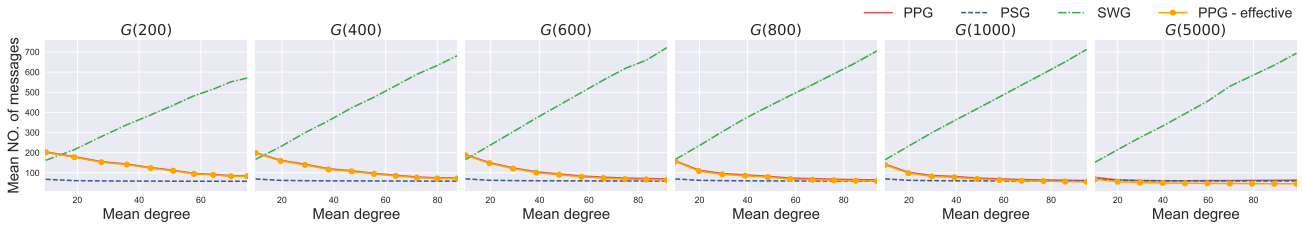
This paper establishes the comparison of three gossip algorithms - PPG, PSG and SWG, which focuses on the accuracy, efficiency in time and message passing. During the course of the comparison, we also demonstrate the stability and scalability of the algorithms. We suggest to employ PSG when operating in a smaller graph (size < 1000 and mean degree < 100). While for a larger and better connected graph, PPG is a the one should be considered. In addition, we examine the correlation between the accuracy/efficiency and the graph



(a) YT



(b) N1



(c) N2

Fig. 6. Mean message performance

properties, and find that there are multiple factors affecting the accuracy and time efficiency. This coincides with the time complexity (for the asynchronous gossip algorithms) which we specify, i.e., ϱ is governed by multiple multiple graph characteristics.

There are a number of potential future directions. First, one may continue studying the relationship between the algorithm performance and the properties of the complex networks, both theoretically and empirically. Apart from that, it is also crucial to optimize the algorithm design, in particular for PSG and PPG. We believe that with better design, PSG would become extremely stable and efficient, through cutting off substantially many waste rounds and lowering the stopping threshold τ. In one sense, creating and analyzing gossip algorithms in directed graphs is also a critical task. Furthermore, we may also keen to test the algorithms on small-world [45] and other complex networks, which have not yet been researched with gossip algorithms.

APPENDIX A
THEORETICAL ANALYSIS

A. Binary Gossip

1) PPG: [10]–[12], [44] made a key assumption that the transition matrix $\mathbf{P}$ is symmetric, and devariated that the expected diffusion matrix (denoted by $\mathbb{E}[\mathbf{W}]$) was doubly stochastic and symmetric. However, a real-world scenario does not necessarily maintain a doubly stochastic and symmetric matrix. Their formulation of the doubly stochastic diffusion matrix depends on that the probability of $i \to j$ and $j \to i$ are equal - it is not always assured in real-world scenarios. Hence we provide an algebraic proof of the push-pull convergence which relates to *waste round* straightforwardly. Although there are many methods proposed to prove the convergence of the gossip protocols [5], [13], [31], [35], [36], we stick to the approach close to those in [10]–[12], [44].

One can define the error vector $\mathbf{z}$ as

$$\mathbf{z} = \mathbf{x} - \overline{x}\mathbf{w}. \tag{7}$$

Henceforth, we have

$$\begin{aligned}\mathbf{z}(t+1)^{\mathrm{T}} &= \mathbf{x}(t+1)^{\mathrm{T}} - \overline{x}\mathbf{w}(t+1)^{\mathrm{T}} \\ &= \mathbf{x}(t)^{\mathrm{T}}\mathbf{K}(t) - \overline{x}\mathbf{w}(t)^{\mathrm{T}}\mathbf{K}(t) \\ &= \left(\mathbf{x}(t)^{\mathrm{T}} - \overline{x}\mathbf{w}(t)^{\mathrm{T}}\right)\mathbf{K}(t) \\ &= \mathbf{z}(t)^{\mathrm{T}}\mathbf{K}(t). \end{aligned} \tag{8}$$

Next, it is trivial to check

$$\|z(t+1)\|^2 = \mathbf{z}(t)^{\mathrm{T}}\mathbf{K}(t)\left[\mathbf{z}(t)^{\mathrm{T}}\mathbf{K}(t)\right]^{\mathrm{T}} \tag{9}$$

$$= \sum_j \left(\sum_i k_{i,j}(t) z_i(t)\right)^2. \tag{10}$$

In PPG, for node u contacting v at the time slot t, we have

$$k_{i,j}(t) = \begin{cases} \frac{1}{2} & i,j \in \{u,v\} \\ 1 & i = j; i,j \notin \{u,v\} \\ 0 & otherwise\,. \end{cases}$$

For simplicity, we use $k_{i,j}$ and z_i instead of $k_{i,j}(t)$ and $z_i(t)$ for the latter equations. Following Eq. (10), we obtain

$$\begin{aligned}\|\mathbf{z}(t+1)\|^2 &= \left(\sum_i k_{i,u}z_i\right)^2 + \left(\sum_i k_{i,v}z_i\right)^2 + \sum_{i\notin\{u,v\}} z_i^2 \\ &= \sum_{i\notin\{u,v\}} z_i^2 + (k_{u,u}z_u + k_{v,u}z_v)^2 \\ &\quad + (k_{v,v}z_v + k_{u,v}z_u)^2 \\ &= 2\left(\frac{z_u+z_v}{2}\right)^2 + \sum_{i\notin\{u,v\}} z_i^2 \\ &\le 2\frac{z_u^2+z_v^2}{2} + \sum_{i\notin\{u,v\}} z_i^2 \\ &= \|\mathbf{z}(t)\|^2.\end{aligned}$$

Clearly, $\|\mathbf{z}(t+1)\|^2 = \|\mathbf{z}(t)\|^2$ iff $z_u = z_v$. It also indicates that $t+1$ is a *waste round*, if $z_u(t) = z_v(t)$. Otherwise it shows that, at every round, $\|\mathbf{z}(t+1)\|^2 < \|\mathbf{z}(t)\|^2$. By recurrence, the mean error over all sites keeps decreasing. Once the process remains unchangeable in a connected graph, it means that all value minuses the global mean are equal.

$$\forall i,j \in V : x_i - \overline{x} = x_j - \overline{x} \implies x_i = x_j.$$

According to *mass conservation*, we deduce that all $x_i = \overline{x}$. Therefore, the process reaches the consensus of the global mean.

Without the loss of generosity, we only need to assure that $\sum_j p_{i,j} = 1$. Also, $\sum_i p_{i,*} = 0$ is guaranteed while $\sum_i p_{*,i} = 0$ is not assured. Let i^* denote i is awakened, and $y_{i\to j}$ denote the product of $\mathbf{z}(t+1)^{\mathrm{T}}\mathbf{z}(t+1)$ for node i contacting j.

$$\begin{aligned}\|\mathbf{z}(t+1)\|^2 &= \sum_i \sum_{j\neq i} p(i\to j, i^*) y_{i\to j} \qquad (12)\\ &= \sum_i \sum_{j\neq i} \frac{1}{n} p_{i,j} \left[2\left(\frac{z_i+z_j}{2}\right)^2 + \sum_{k\notin\{i,j\}} z_k^2\right] \\ &\le \sum_i \sum_{j\neq i} \frac{1}{n} p_{i,j} \|\mathbf{z}(t)\|^2 \\ &= \frac{1}{n}\sum_i \|\mathbf{z}(t)\|^2 \\ &= \|\mathbf{z}(t)\|_2.\end{aligned}$$

Assuming any single node does not equal to all others before convergence, we hence obtain

$$\mathbb{E}\left[\|\mathbf{z}(t+1)\|^2\right] < \|\mathbf{z}(t)\|^2. \qquad (13)$$

Thus, we have $\mathbb{E}\left[\mathbf{z}(t+1)^{\mathrm{T}}\mathbf{z}(t+1)\right] = \lambda^t \mathbf{z}(0)^{\mathrm{T}}\mathbf{z}(0)$, where $\varrho < 1$. Based on Markov inequality, there is:

$$\mathbb{P}\left(\frac{\|\mathbf{z}(t+1)\|^2}{\|\mathbf{z}(0)\|^2} \ge \epsilon\right) \le \frac{\lambda^t}{\epsilon} \le \epsilon \implies t \le \frac{2\log\epsilon}{\log\varrho}. \qquad (14)$$

To conclude, the mean round complexity is then bounded by $O(\frac{2}{n} \times \frac{2\log\epsilon}{\log\varrho}) = O(\frac{\log\epsilon}{n\log\varrho})$, as one t in PPG has actually two rounds for two nodes, i.e. one for each.

B. Weighted Gossip

In the following section, we analyze SWG first and then PSG.

1) SWG: We apply the asynchronous setting in [5] to SWG - $\forall t:\ k_{i,j}(t) = w_{i,j}(t)$, $k_{j,j}(t)\mathbb{1}_{j\neq i} = 1$ and other entries in $\mathbf{W}$ equal to 0. Also, $\mathbb{E}[\mathbf{K}] = \sum_i p(i\to j, i^*)\mathbf{K}_{i\to j}$. We have postulated that each node is randomly awakened, and thus have $p(i^*) = 1/n$ for any i. The expected transition matrix is shown as follows:

$$\mathbb{E}\left[k_{i,j}\right] = \begin{cases} \frac{1}{n}p_{i,j}w_{i,j} & i \neq j \\ \frac{1}{n}p_{i,i}w_{i,i} + \sum_{j\neq i}\frac{1}{n} & otherwise\,. \end{cases}$$

Letting $\mathbb{E}[\mathbf{K}] = \mathcal{Q} + \frac{n-1}{n}\mathbf{I}$, the auxiliary matrix $\mathbf{Q}$ is defined by

$$q_{i,j} = \begin{cases} \frac{1}{n}w_{i,j} & (i,j)\in E \\ 0 & others\,. \end{cases} \qquad (15)$$

Considering Eq. (9), we write

$$\begin{aligned}&\mathbf{z}(t)^{\mathrm{T}}\mathbb{E}[\mathbf{K}(t)]\left\{\mathbf{z}(t)^{\mathrm{T}}\mathbb{E}[\mathbf{K}(t)]\right\}^{\mathrm{T}} \\ &= \mathbf{z}(t)^{\mathrm{T}}\left(\mathcal{Q}(t) + \frac{n-1}{n}\mathbf{I}\right)\left(\mathcal{Q}(t)^{\mathrm{T}} + \frac{n-1}{n}\mathbf{I}\right)\mathbf{z}(t) \\ &= \mathbf{z}(t)^{\mathrm{T}}\mathcal{Q}(t)\mathcal{Q}(t)^{\mathrm{T}}\mathbf{z}(t) + \frac{n-1}{n}\mathbf{z}(t)^{\mathrm{T}}\mathcal{Q}(t)^{\mathrm{T}}\mathbf{z}(t) \\ &\quad + \frac{n-1}{n}\mathbf{z}(t)^{\mathrm{T}}\mathcal{Q}(t)\mathbf{z}(t) + \frac{(n-1)^2}{n^2}\|\mathbf{z}\|^2. \qquad (16)\end{aligned}$$

We are going to analyze the terms in turn. First, we have $\mathbf{z}(t)^{\mathrm{T}}\mathcal{Q}(t)^{\mathrm{T}}\mathbf{z}(t) = \mathbf{z}(t)^{\mathrm{T}}\mathcal{Q}(t)\mathbf{z}(t)$, given the fact that the final outcome of both sides are a singular. It satisfies that

$$[\mathcal{Q}(t)\mathbf{z}(t)]^{\mathrm{T}}\mathbf{z}(t) = \mathbf{z}(t)^{\mathrm{T}}[\mathcal{Q}(t)\mathbf{z}(t)].$$

$n\mathcal{Q}(t)$ is a stochastic matrix and thus its largest eigenvalue is 1 and largest eigenvector is $\mathbf{1}$. Provided the premise, *Rayleigh quotient* is bounded by the second largest eigenvalue λ_2. The inequality follows,

$$\mathbf{z}(t)^{\mathrm{T}}\mathcal{Q}(t)^{\mathrm{T}}\mathbf{z}(t) \le \frac{1}{n}\lambda_2(n\mathcal{Q}(t))\mathbf{z}(t)^{\mathrm{T}}\mathbf{z}(t) \le \frac{1}{n}\|\mathbf{z}(t)\|^2. \qquad (17)$$

Let us denote $\mathbf{A}(t) = \mathcal{Q}(t)\mathcal{Q}(t)^{\mathrm{T}}$. $\mathbf{A}$ may have the largest eigenvalue greater than 1, and therefore the corresponding eigenvector is not $\mathbf{1}$. We cannot apply *Rayleigh quotient* to it. We stick to Eq. (10). Again, we omit the time stamp in the following deduction.

$$\begin{aligned}n^2\mathbf{z}^{\mathrm{T}}\mathbf{A}\mathbf{z} &= \sum_j\sum_i q_{i,j}^2 z_i^2 + \sum_i\sum_{j\neq i}(q_{i,j}q_{j,i}z_iz_j) \\ &= \sum_i\left(\sum_j q_{i,j}^2 + 1 - 1\right)z_i^2 \\ &\quad + \sum_i\sum_{j\neq i}\left[\left(q_{i,j}k_{j,i} + 1 - 1\right)z_iz_j\right] \\ &= \left(\sum_i z_i\right)^2 + \sum_i\left(\sum_j q_{i,j}^2 - 1\right)z_i^2 \\ &\quad + \sum_i\sum_{j\neq i}\left[\left(q_{i,j}q_{j,i} - 1\right)z_iz_j\right]\end{aligned}$$

$$= \sum_i \left(\sum_j q_{i,j}^2 - 1 \right) z_i^2 + \sum_i \sum_{j \neq i} \left[\left(q_{i,j} q_{j,i} - 1 \right) z_i z_j \right]. \quad (18)$$

Provided that $0 \leq q_{i,j} = w_{i,j} < 1$ and $\sum_j k_{i,j} = 1$, there exists

$$\arg\min_{a,b} q_{a,b} q_{b,a} - 1 \leq q_{i,j} q_{j,i} - 1 \leq \arg\max_{a,b} q_{a,b} q_{b,a} - 1.$$

According to Corollary 1, for some $-1 \leq \tilde{c} < 0$, we have

$$\sum_i \sum_{j \neq i} \left[\left(q_{i,j} q_{j,i} - 1 \right) z_i z_j \right] = \tilde{c} \sum_i \sum_{j \neq i} z_i z_j \quad s.t. \sum_i \sum_{j \neq i} z_i z_j \neq 0.$$

$\sum_i \sum_{j \neq i} z_i z_j \neq 0$ holds when the algorithm is not yet converged. Otherwise we have

$$-\sum_i z_i^2 = \left(\sum_i z_i \right)^2 - \sum_i z_i^2 = \sum_i \sum_{j \neq i} z_i z_j \leq 0.$$

This implies that the error between the global mean the local mean at each node is 0, by which the process is thought to be converged.

Therefore, Section A-B1 is then extended to

$$\begin{aligned} &\sum_i \left(\sum_j q_{i,j}^2 - 1 \right) z_i^2 + \tilde{c} \sum_i \sum_{j \neq i} z_i z_j \\ &= \sum_i \left(\sum_j q_{i,j}^2 - 1 \right) z_i^2 + \tilde{c} \left[\left(\sum_i z_i \right)^2 - \sum_i z_i^2 \right] \\ &= \sum_i \left(\sum_j q_{i,j}^2 - 1 - \tilde{c} \right) z_i^2 \quad (19a) \\ &< \sum_i z_i^2. \quad (19b) \end{aligned}$$

Eqs. (19a) and (19b) is deducted as follows.

$$\sum_j q_{i,j}^2 - \arg\max_{a,b} q_{a,b} k_{b,a} \leq \sum_j q_{i,j}^2 - 1 - \tilde{c} \leq \sum_j q_{i,j}^2 - \arg\min_{a,b} q_{a,b} q_{b,a}. \quad (20)$$

It is clear that $\frac{1}{n} \leq \sum_j q_{i,j}^2 \leq 1$ subject to $\sum_j q_{i,j} = 1$ and $k_{i,j} > 0$, as a basic optimization problem. Plus, our graph is connected, and thus has no two nodes sending the whole portion of their data to each other. Without the loss of generosity, we let $1 - \arg\max_{a,b} k_{a,b} k_{b,a} < \epsilon$. Hence, we obtain $-\frac{n-1}{n} < \sum_j k_{i,j}^2 - 1 - \tilde{c} \leq 1$. Also, we have known that $\sum_i z_i^2$ is nonnegative, and thus bound it to be $0 \leq \sum_j k_{i,j}^2 - 1 - \tilde{c} < 1$. This concludes that

$$\mathbf{z}(t)^{\mathrm{T}} \mathcal{Q}(t) \mathcal{Q}(t)^{\mathrm{T}} \mathbf{z}(t) \leq \frac{1}{n^2} \|\mathbf{z}(t)\|^2.$$

Refer back to Eq. (16), it follows

$$\begin{aligned} &\mathbf{z}(t)^{\mathrm{T}} \mathbb{E}[\mathbf{K}(t)] \left\{ \mathbf{z}(t)^{\mathrm{T}} \mathbb{E}[\mathbf{K}(t)] \right\}^{\mathrm{T}} \\ &\leq \left(\frac{1}{n^2} + \frac{2(n-1)}{n^2} + \frac{(n-1)^2}{n^2} \right) \|\mathbf{z}(t)\|^2 \\ &\leq \|\mathbf{z}(t)\|^2 \quad (21) \end{aligned}$$

Given the above results, one can conclude that the mean round complexity of SWG is also bounded by $O(\frac{\log \epsilon}{n \log \varrho})$, for some $\varrho < 1$, via the same technique for PPG.

2) PSG: PSG is a special case of weighted gossip, and hence can employ the procedure for the SWG directly. Considering at time t, there is $u \to v$. The weight $\mathbf{W}$ for PSG is then

$$w_{i,j} = \begin{cases} \frac{1}{2} & i,j \in \{u,v\} : i \neq j \vee i = j \\ 1 & i = j \neq u \\ 0 & otherwise. \end{cases} \quad (22)$$

Following that, we derive $\mathcal{Q}$.

$$q_{i,j} = \begin{cases} \frac{1}{2n} p_{i,j} & i \neq j \\ \frac{1}{n} \sum_{j \in \mathcal{N}_i, j \neq i} \frac{1}{2} p_{i,j} = \frac{1}{2n} & i = j \\ 0 & otherwise. \end{cases} \quad (23)$$

Given that $n\mathcal{Q}$ is stochastic, Eq. (17) for PSG holds.

Next, we substitute this $\mathcal{Q}$ into Section A-B1, it is then

$$\sum_i \left(\sum_{j,j \neq i} \frac{1}{4} p_{i,j}^2 + \frac{1}{4} - 1 \right) z_i^2 + \sum_i \sum_{j \neq i} \left[\left(\frac{1}{4} p_{i,j} p_{j,i} - 1 \right) z_i z_j \right] \quad (24)$$

Letting $\tilde{c} = \frac{1}{4} p_{i,j} p_{j,i} - 1$,

$$-1 \leq \tilde{c} < -\frac{3}{4}.$$

Also, recall Eq. (19a).

$$\frac{1}{n} \leq \sum_{j,j \neq i} p_{i,j}^2 + \frac{1}{4} - 1 - \tilde{c} \leq \frac{1}{2}.$$

This concludes that

$$\mathbf{z}(t)^{\mathrm{T}} \mathbb{E}[\mathcal{Q}(t)] \mathbb{E}[\mathcal{Q}(t)^{\mathrm{T}}] \mathbf{z}(t) < \frac{1}{n} \mathbf{z}(t)^{\mathrm{T}} \mathbf{z}(t).$$

Finally, we come to the result that the time complexity of PSG is also within $O(\frac{\log \epsilon}{n \log \varrho})$.

Corollary 1. *Given any i and $0 \leq a \leq c_i \leq b$ and $\sum_i d_i \neq 0$ holds, $\exists c : \sum_i c_i d_i = c \sum_i d_i,\ s.t.\ a \leq c \leq b$.*

Proof: Since $\sum_i d_i \neq 0$, we have

$$c = \frac{\sum_i c_i d_i}{\sum_i d_i}$$

If $\sum_i d_i > 0$, $a \sum_i d_i \leq \sum_i c_i d_i \leq b \sum_i d_i$ holds. If $\sum_i d_i < 0$, there comes $b \sum_i d_i \leq \sum_i c_i d_i \leq a \sum_i d_i$. Both cases conclude that $a \leq c \leq b$. ■

ACKNOWLEDGMENT

The authors would like to thank...

REFERENCES

[1] R. Albert and A.-L. Barabási, "Statistical mechanics of complex networks," *Reviews of Modern Physics*, vol. 74, no. January, pp. 47–97, 2002.

[2] R. Albert, H. Jeong, and A.-L. Barabasi, "Internet: Diameter of the World-Wide Web," *Nature*, vol. 401, no. September, pp. 130–131, 1999.

[3] T. Aysal and M. Yildiz, "Broadcast gossip algorithms," *Theory Workshop, 2008.*, pp. 343–347, 2008.

[4] A.-L. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, p. 11, 1999.

[5] F. Bénézit, V. Blondel, P. Thiran, J. Tsitsiklis, and M. Vetterli, "Weighted gossip: Distributed averaging using non-doubly stochastic matrices," *IEEE International Symposium on Information Theory - Proceedings*, no. 0, pp. 1753–1757, 2010.

[6] P. Bianchi, S. Clemencon, G. Morrai, and J. Jakubowicz, "On-line learning gossip algorithm in multi-agent systems with local decision rules," *Proceedings - 2013 IEEE International Conference on Big Data, Big Data 2013*, pp. 6–14, 2013.

[7] K. Birman, "The promise, and limitations, of gossip protocols," *ACM SIGOPS Operating Systems Review*, vol. 41, p. 8, 2007.

[8] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, "Complex networks: Structure and dynamics," *Physics reports*, vol. 424, no. 4, pp. 175–308, 2006.

[9] V. S. Borkar, R. Makhijani, and R. Sundaresan, "Asynchronous gossip for averaging and spectral ranking," *IEEE Journal on Selected Topics in Signal Processing*, vol. 8, no. 4, pp. 703–716, 2014.

[10] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, "Analysis and optimization of randomized gossip algorithms," *43rd IEEE Conference on Decision and Control (CDC) (IEEE Cat. No.04CH37601)*, vol. 5, pp. 0–5, 2004.

[11] ——, "Gossip algorithms: Design, analysis and applications," in *Proceedings - IEEE INFOCOM*, vol. 3, 2005, pp. 1653–1664.

[12] ——, "Randomized gossip algorithms," *IEEE Transactions on Information Theory*, vol. 52, no. 6, pp. 2508–2530, 2006.

[13] J.-Y. Chen and G. Pandurangan, "Optimal Gossip-Based Aggregate Computation," *SIAM Journal on Computing*, vol. 41, no. 3, pp. 455–483, Jan. 2010.

[14] G. W. Corder and D. I. Foreman, *Nonparametric statistics: A step-by-step approach*. John Wiley & Sons, 2014.

[15] G. Di Fatta, F. Blasa, S. Cafiero, and G. Fortino, "Epidemic K-Means clustering," in *Proceedings - IEEE International Conference on Data Mining, ICDM*. Ieee, Dec. 2011, pp. 151–158.

[16] ——, "Fault tolerant decentralised K-Means clustering for asynchronous large-scale networks," pp. 317–329, Mar. 2012.

[17] P. Erdös and A. Rényi, "On random graphs," *Publicationes Mathematicae*, vol. 6, pp. 290–297, 1959.

[18] P. Erdös and A. Rényi, "On the evolution of random graphs," *Publ. Math. Inst. Hungar. Acad. Sci*, vol. 5, pp. 17–61, 1960.

[19] J. Fellus, D. Picard, and P.-H. Gosselin, "Decentralized K-means using randomized gossip protocols for clustering large datasets," in *Proceedings - IEEE 13th International Conference on Data Mining Workshops, ICDMW 2013*, 2013, pp. 599–606.

[20] ——, "Dimensionality reduction in decentralized networks by Gossip aggregation of principal components analyzers," *European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning*, pp. 171–176, 2014.

[21] ——, "Asynchronous gossip principal components analysis," *Neurocomputing*, vol. 169, pp. 262–271, 2015.

[22] K. Flouri, B. Beferull-Lozano, and P. Tsakalides, "Optimal gossip algorithm for distributed consensus svm training in wireless sensor networks," in *2009 16th International Conference on Digital Signal Processing*. IEEE, 2009, pp. 1–6.

[23] B. Garbinato, D. Rochat, and M. Tomassini, "Impact of scale-free topologies on gossiping in ad hoc networks," in *Proceedings - 6th IEEE International Symposium on Network Computing and Applications, NCA 2007*, no. 1, 2007, pp. 269–272.

[24] C. Georgiou, S. Gilbert, R. Guerraoui, and D. R. Kowalski, "On the complexity of asynchronous gossip," *Proceedings of the Annual ACM Symposium on Principles of Distributed Computing*, pp. 135–144, 2008.

[25] ——, "Asynchronous gossip," *Journal of the ACM*, vol. 60, no. 2, pp. 1–42, 2013.

[26] X. Guardiola, R. Guimera, A. Arenas, A. Diaz-Guilera, D. Streib, and L. a. N. Amaral, "Macro- and micro-structure of trust networks," *arXiv preprint cond-mat/0206240*, pp. 1–5, 2002.

[27] R. Gupta, A. C. Maali, and Y. N. Singh, "Adaptive Push-Then-Pull Gossip Algorithm for Scale-free Networks," *arXiv preprint arXiv:1310.5985*, 2013.

[28] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using NetworkX," in *Proceedings of the 7th Python in Science Conference (SciPy2008)*, Pasadena, CA USA, Aug. 2008, pp. 11–15.

[29] R. Hu, J. Sopena, L. Arantes, P. Sens, and I. Demeure, "Fair comparison of gossip algorithms over large-scale random topologies," in *Proceedings of the IEEE Symposium on Reliable Distributed Systems*. IEEE, Oct. 2012, pp. 331–340.

[30] ——, "Efficient dissemination algorithm for scale-Free topologies," in *Proceedings of the International Conference on Parallel Processing*, vol. 6, 2013, pp. 310–319.

[31] F. Iutzeler, P. Ciblat, and W. Hachem, "Analysis of sum-weight-like algorithms for averaging in wireless sensor networks," *IEEE Transactions on Signal Processing*, vol. 61, no. 11, pp. 2802–2814, Jun. 2013.

[32] F. Iutzeler, P. Ciblat, W. Hachem, and J. Jakubowicz, "A New Broacast Based Distributed Averaging Algorithm over Wireless Sensor Networks," *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*, pp. 3117–3120, 2012.

[33] M. Jelasity, A. Montresor, and O. Babaoglu, "Gossip-based aggregation in large dynamic networks," *ACM Transactions on Computer Systems*, vol. 23, no. August, pp. 219–252, 2005.

[34] H. Jeong, S. P. Mason, A.-L. Barabási, and Z. N. Oltvai, "Lethality and centrality in protein networks." *Nature*, vol. 411, no. May, pp. 41–42, 2001.

[35] S. Kashyap, S. Deb, K. V. M. Naidu, R. Rastogi, and A. Srinivasan, "Efficient gossip-based aggregate computation," *Proceedings of the twenty-fifth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems - PODS '06*, p. 308, 2006.

[36] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-based computation of aggregate information," *44th Annual IEEE Symposium on Foundations of Computer Science, 2003. Proceedings.*, pp. 482–491, 2003.

[37] S. Lee and A. Nedic, "Asynchronous Gossip-Based Random Projection Algorithms over Networks," *IEEE Transactions on Automatic Control*, vol. 61, no. 4, pp. 953–968, 2016.

[38] A. Mislove, M. Marcon, K. P. Gummadi, P. Druschel, and B. Bhattacharjee, "Measurement and Analysis of Online Social Networks," in *Proceedings of the 5th ACM/Usenix Internet Measurement Conference (IMC'07)*, San Diego, CA, October 2007.

[39] A. Nikseresht and M. Gelgon, "Gossip-based computation of a Gaussian mixture model for distributed multimedia indexing," in *Multimedia, IEEE Transactions on*, vol. 10, no. 3, Apr. 2008, pp. 385–392.

[40] R. Ormándi, I. Heged, and M. Jelasity, "Gossip Learning with Linear Models on Fully Distributed Data," *Concurrency and Computation: Practice and Experience*, pp. 1–14, 2011.

[41] A. Polynikis, "Random Walks & Scale-Free Networks," *M. Sc. thesis*, no. August, 2006.

[42] S. S. Ram, A. Nedić, and V. V. Veeravalli, "Asynchronous gossip algorithm for stochastic optimization: Constant stepsize analysis," *Recent Advances in Optimization and its Applications in Engineering*, pp. 51–60, 2010.

[43] G. D. M. Serugendo, M.-P. Gleizes, and A. Karageorgos, *Self-organising software: From natural to artificial adaptation*. Springer Science & Business Media, 2011.

[44] D. Shah, "Gossip Algorithms," *Foundations and Trends in Networking*, vol. 3, no. 1, pp. 1–125, 2007.

[45] D. J. Watts and S. H. Strogatz, "Collective dynamics of 'small-world' networks." *Nature*, vol. 393, no. June, pp. 440–442, 1998.

[46] D. B. West, *Introduction to graph theory*. Prentice hall Upper Saddle River, 2001, vol. 2.