Fast Convergence of Quantized Consensus Using Metropolis Chains Over Static and Dynamic Networks

Tamer Başar†, Seyed Rasoul Etesami‡, Alex Olshevsky‡

Abstract—We consider the quantized consensus problem on undirected connected graphs with $n$ nodes, and devise a protocol with fast convergence time to the set of consensus points. Specifically, we show that when the edges of a static network are activated based on Poisson processes with Metropolis rates, the expected convergence time to the set of consensus points is at most $O(n^2 \log n)$. We further show an upper bound of $O(n^2 \log^2 n)$ for the expected convergence time of the same protocol over connected time-varying networks. These bounds are better than all previous convergence times for randomized quantized consensus.

Index Terms—Quantized consensus, consensus convergence time, Metropolis chains.

I. INTRODUCTION

There has been much recent interest in the design of control protocols for distributed systems, motivated in part by the need to develop protocols for networks of autonomous agents characterized by the lack of centralized information and time-varying connectivity. A canonical problem within the field is the so-called average consensus problem, wherein a group of agents must agree on the average of their initial values while interacting with neighbors in a (possibly time-varying) network. Protocols for consensus problems must be distributed, relying only on local information at each node, and robust to unexpected changes in topology.

It is well-understood by now that protocols for the consensus play an important role in a number of more sophisticated multi-agent tasks. We mention distributed optimization [2], [3], coverage control [4], formation control [5], [6], cooperative statistical inference [7], [8], power control [9], load balancing [10], epidemic routing [11], as examples of control and coordination problems with proposed solutions relying crucially on consensus.

Simplicity and efficiency are two complementary goals in the design of distributed protocols for consensus and other multi-agent problems. On the one hand, one would like to design a simple protocol which is easy to implement and which has little or reliance on global knowledge about the system. On the other hand, one typically wants to achieve a global goal quickly and efficiently. Interestingly, there often appears to be a trade-off between convergence speed and locality and simplicity of the algorithm, see [12], [13], [14], [15].

Our work is motivated by the observation that often working with real-valued variables in multi-agent control is neither necessary nor efficient. Indeed, limited memory and storage at each agent often forces the variables kept by each agent to lie in a discrete set. We therefore consider the quantized consensus problem, a variation on the consensus problem where the values of each agent are constrained to be integers lying within a certain range. Previous literature on this problem includes [16], [17], [18], [19], [20], [21], [22], [23], [24], [25], [26].

More precisely, we will consider a set of $n$ “agents” or “nodes”, each beginning with an integer value in a certain range $[l, L]$. These nodes can communicate with their neighbors in a certain undirected, connected communication graph (which may be time-varying). The typical goal of computing the average must now be revised since the average may not be an integer. Thus we define a collection of integer values to belong to the “consensus set” if any two integers are at most one apart. We would like to design a protocol which brings the collection of integers at the nodes to the consensus set as fast as possible.

We note that the quantized consensus problem restricts both transmissions to neighbors as well as the values stored by the nodes to be integers. This is related but mathematically distinct from the related problem of “consensus with finite data rate” wherein nodes store real numbers but transmissions are quantized. We refer the reader to [27], [28], [29], [30], [31] for literature on this related problem.

There has been a considerable amount of work on quantized consensus since its introduction in [16], and we make no attempt to provide here a complete survey of the literature. Rather, we summarize the literature which focuses on worst-case convergence times in terms of the number of nodes $n$, which is the focus of the present paper.

The original paper [16] contained upper bounds for a natural quantized consensus protocol on a variety of common graphs. A few years later, the work of [32] proposed a quantized consensus protocol with upper bound of $O(n^3)$ on the expected convergence time on any fixed graph. For
dynamic graphs, [32] obtained a convergence time scaling of $O(n^3)$. The paper [26] obtained an upper bound of $O(n^3)$, but only on complete graphs. Recently, a paper by the first two authors of the current paper [17] showed that a certain “unbiased” quantized consensus protocol has maximum expected convergence time of $O(nmD \log n)$ on static networks, where $m$ and $D$ denote the number of edges and the diameter of the network, respectively. Unbiasedness means that each step of the protocol was based on choosing a random edge in the network, and all edges were equally likely to be chosen. A corresponding bound of $O(n^2 m_{\max} D_{\max} \log^2 n)$ on the expected convergence time of unbiased quantized consensus on connected time-varying networks was given in the follow-up work by the same authors [33], where $m_{\max}$ and $D_{\max}$ denote, respectively, the maximum number of edges and the maximum diameter in the sequence of time-varying networks.

A faster upper bound on convergence time and only for static networks was given in the recent paper [34], where a protocol was provided whose expected convergence time in general static networks is $O(n^3 \log n)$. As of the time of writing this paper, the upper bounds of $O(n^3 \log n)$ and $O(n^2 m_{\max} D_{\max} \log^2 n)$, respectively, are the fastest-known protocols for randomized quantized consensus over static and dynamic networks known to us in the previous literature [33], [34]. We also note that a deterministic version of the quantized-consensus algorithm was introduced in [35], where the authors showed a convergence time of $O(n^2)$ for such dynamics over static networks. However, unlike the randomized quantized consensus protocols considered here as well as in [34], [17], [33], adopting the protocols from [35] to time-varying graphs appears to be impossible.

Our main results in the current paper are a further improvement of the convergence rate of quantized consensus. Our main innovation is to design a protocol wherein nodes cooperate to perform updates on edges connecting them at so-called Metropolis rates (to be defined later). We show that this protocol results in an expected time of $O(n^2 \log n)$ until it reaches the consensus set on any fixed connected undirected graph, and an expected time of $O(n^2 \log^2 n)$ to reach the consensus set over any sequence of time-varying connected undirected graphs.

It is instructive to compare the rates we obtain here with those in the previous literature. On fixed graphs, our essentially quadratic convergence time is an improvement on the essentially cubic convergence time of [34] by a factor of $n$. Over dynamic graphs, the best previous bound of $O(n^2 m_{\max} D_{\max} \log^2 n)$ from [33] is at worst $O(n^3 \log^2 n)$, and our essentially quadratic upper bound appears to be an improvement by a factor of $n^{3/2}$. However, this comparison is somewhat misleading, since [33] performs one update per unit time whereas our protocol performs $O(n)$ updates per unit time (as we will show later). It is therefore more correct to assert that our protocol is an improvement by a factor of $n^{3/2}$ over the previously best convergence time on dynamic graphs.

This paper is organized as follows. In Section [II], we formulate the quantized consensus problem and discuss some of its properties. In Section [III] we prove our main result for the static networks, namely the upper bound of $O(n^2 \log n)$ for the expected convergence time. In Section [IV] we prove our main result for dynamic networks, namely the upper bound of $O(n^2 \log^2 n)$ for the expected convergence time. In Section [V] we provide upper bounds for our protocol over some specific networks. In Section [VI] we provide some simulation results and compare our theoretical bounds to the performance achieved in some concrete fixed and time-varying networks. We conclude the paper by identifying some future directions of research in Section [VII].

Notation: We let $[n] = \{1, 2, \ldots, n\}$. For an undirected graph $G = (V, E)$ we let $N(x)$ be the set of neighbors of $x$, and furthermore, let $d(x)$ denote the degree of $x$ i.e., $d(x) = |N(x)|$. Also, we let $G \times G = (V \times V, E')$ be the Cartesian product of $G$, with the set of nodes $(x, y) \in V \times V$ and the set of edges $E'$ such that $\{(x, y), (r, s)\} \in E'$ if and only if $x = r, s \in N(y)$ or $y = s, r \in N(x)$. For a given vector $v$, we denote its $i$th entry by $v_i$ and its transpose by $v'$. We say that a matrix $A$ with nonnegative entries is stochastic if each row of $A$ sums to 1. If both $A$ and $A'$ are stochastic, we say that $A$ is doubly stochastic. Given a matrix $A$ with real eigenvalues, we denote its maximum and minimum eigenvalues by $\alpha_{\max}(A)$ and $\alpha_{\min}(A)$, respectively. Finally, given an electric network with resistances $\{r_{e}\}_{e \in E}$ on its links, we let $R(x \leftrightarrow y)$ denote the effective resistance between the nodes $x$ and $y$.

II. PROBLEM FORMULATION

In this section, we assume we are given a fixed, undirected, connected graph $G = (V, E)$ without self-loops. We adopt the convention of using $n$ for the number of nodes in this graph, i.e., $n = |V|$. The Metropolis Markov chain on this graph is then defined as follows.

Let $\mathcal{M}$ be a square matrix whose $ij$th entry is defined as

$$M_{ij} = \begin{cases} \frac{1}{\max(d(i), d(j))}, & \text{if } i \neq j, j \in N(i) \\ 1 - \sum_{j \in N(i)} M_{ij}, & \text{if } i = j \\ 0, & \text{otherwise,} \end{cases} \quad (1)$$

Note that $\mathcal{M}$ is symmetric, nonnegative, and doubly stochastic. We refer to the Markov chain which transitions according to $\mathcal{M}$ as the Metropolis chain. Moreover, given nodes $x, y \in V$, the hitting time $H^x_m(x, y)$ is the expected time until the Metropolis chain with initial position at node $x$ reaches node $y$ (quantities associated with the Metropolis chain will generally be denoted with an “$m$” superscript). For future use and convenience, we adopt the notation $\lambda_{ij} = 1/\max(d(i), d(j))$ whenever $\{i, j\} \in E, i \neq j$ and $\lambda_{ij} = 0$ otherwise.

A. Quantized Metropolis dynamics

We next introduce a continuous time quantized process based on Metropolis weights, whose behavior will turn out to be related to the Metropolis chain. For each link $\{i, j\} \in$
\(E\), \(i \neq j\) of the graph \(G\), we consider a Poisson process of rate \(\lambda_{ij}\). Each node \(i\) begins with an integer value \(x_i(0)\) which lies in the range \([l, L]\). Each time the process corresponding to an edge \(\{i, j\} \in E\) registers an arrival, the two nodes \(i, j\) perform the quantized consensus update from [16]:

\[
x_i(t^+) = \begin{cases} 
    x_i(t) - 1, & \text{if } x_i(t) > x_j(t) \\
    x_i(t) + 1, & \text{if } x_i(t) < x_j(t) \\
    x_i(t), & \text{if } x_i(t) = x_j(t),
\end{cases}
\]

and likewise for node \(j\). In other words, each \(x_i(t)\) is an integer-valued jump process whose jumps occur whenever an arrival occurs at any edge incident on \(i\).

Note that the above update rule allows for the possibility that \(x_i(t^+) = x_i(t)\). In this case, we will say that the update at time \(t\) was trivial. Furthermore, observe that if \(|x_i(t) - x_j(t)| = 1\), then the update of Eq. (2) will cause nodes \(i\) and \(j\) to swap values, i.e., \(x_i(t^+) = x_j(t), x_j(t^+) = x_i(t)\). In this case, we will also say that the update was trivial. If neither of these two cases has occurred during an update, we will say that the update was non-trivial.

Note that the quantized Metropolis dynamics can be implemented in a distributed manner. Furthermore, since \(\sum_{j \in N(i)} \lambda_{ij} \leq 1\), we have that \(\sum_{(i,j) \in E} \lambda_{ij} = O(n)\), and so the Metropolis quantized consensus protocol performs \(O(n)\) updates per unit time.

B. Relevant Existing Results

First we note that, since the updating rules given in Eq. (2) preserve the average value, we have that under the quantized Metropolis dynamics \(\bar{x}(t) = \sum_{i=1}^{n} x_i(t)/n\) does not change with time and equals \(\bar{x}(0)\) for all \(t \geq 0\). Moreover, it has been shown earlier in [16] that such dynamics will converge with probability 1 to the consensus set,

\[
C = \left\{ x \mid x_i \in \{ \lfloor \bar{x}(0) \rfloor, \lceil \bar{x}(0) \rceil + 1 \}, \sum_{i=1}^{n} \frac{x_i}{n} = \bar{x}(0), i \in [n] \right\},
\]

and that the Lyapunov function defined by

\[
V(t) = \sum_{i=1}^{n} \left( x_i(t) - \bar{x}(0) \right)^2
\]

will decrease by at least 2 after each nontrivial update. It was also shown in [16] (Lemmas 4 and 5) that, after at most \((L - l)^2 / 8 - n\) nontrivial updates, the dynamics will terminate. Thus a key part of the analysis of such protocols involves bounding the maximum expected time until a nontrivial update happens.

III. Expected Convergence Time over Static Networks

We now begin the analysis of the convergence time of the quantized Metropolis algorithm over a fixed graph \(G\). As mentioned just above, a key step is to bound the expected time until the first nontrivial update takes place. Let us introduce some notation for this quantity: for integers \(j_1, \ldots, j_n\) define \(T(j_1, \ldots, j_n)\) to be the expected time until a nontrivial update takes place when node \(i\) begins with integer value \(j_i\). Then \(T\) is the largest possible \(T(j_1, \ldots, j_n)\) starting from any configuration of integers which does not lie in the consensus set \(C\):

\[
T = \max_{(j_1, \ldots, j_n) \notin C} T(j_1, \ldots, j_n).
\]

Definition 3.1: Consider two random walkers moving based on whether the activated edge in the quantized Metropolis dynamics is incident to them. That is, if one of the walkers is at node \(x\), and if the next edge to register an arrival is incident to \(x\), i.e., if it is \(\{x, y\}\) for some \(y \in N(x)\), then the random walker moves from \(x\) to \(y\). We refer to such a process as the original process.

We denote by \(M^o(x, y)\) the expected “meeting time” of this process, defined to be the expected time until an edge incident to both walkers registers an arrival provided the two walkers started at nodes \(x, y\) with \(x \neq y\) (in general, we will denote quantities associated with this process with an “o” superscript). By convention, we set \(M^o(x, x) = 0\) for all \(x \in V\).

Proposition 3.2: \(T = \max_{x, y \in V} M^o(x, y)\).

Proof: Let \(S\) denote the set of all the vectors of size \(n\) in which one entry is 0, one entry is 2, and the remaining entries are 1. It is immediate that \(T = \max_{(j_1, \ldots, j_n) \in S} T(j_1, \ldots, j_n)\). Now consider any configuration in which all the agents on the graph \(G\) have value 1 except two of them which are 0 and 2. Focusing on the nodes which have a 0 and a 2, we see that they perform a random walk according to the original process, and a nontrivial update occurs at precisely the meeting time.

Based on the above proposition, our next step is to bound \(\max_{x, y \in V} M^o(x, y)\). We will actually find it easier to instead bound a meeting time associated with a slightly different process, which we call the virtual process, defined next.

Definition 3.3: The virtual process is identical to the original process until the two walkers \(x, y\) become each other’s neighbors in \(G\). At that time, the edge connecting them registers arrivals according to a Poisson process of rate \(2\lambda_{xy}\) (i.e., twice the rate in the original process). The meeting time \(M^v(x, y)\) is defined to be the expected time until an edge incident on both random walkers ticks provided the walkers start at positions \(x, y\) with \(x \neq y\) (in general, we will denote quantities associated with this process with a “v” superscript). By convention we set \(M^v(x, x) = 0\) for all \(x \in V\).

Definition 3.4: A function \(h : \Omega \to \mathbb{R}\) is called harmonic at a vertex \(x \in \Omega\) for a Markov chain with transition probability matrix \(P\) if \(h(x) = \sum_{y \in \Omega} P(x, y)h(y)\).

Remark 3.5: Given a nonempty subset \(B \subset \Omega\) and a Markov chain with an irreducible transition matrix \(P\), every harmonic function \(h(x) : \Omega \to \mathbb{R}\) over \(\Omega \setminus B\) which satisfies \(h(x) \geq 0, \forall x \in B\), must be nonnegative over the entire \(\Omega\) (see Proposition 9.1 of [36]).
Definition 3.6: A vertex $\theta$ is called a hidden vertex of the Metropolis chain $M$ if $H^m(\theta, x) \leq H^m(x, \theta), \forall x \in V$.

Remark 3.7: It is a known fact that the hitting times of a reversible Markov chain satisfy the following transitivity property

$H^m(x, y) + H^m(y, z) + H^m(z, x) = H^m(x, z) + H^m(z, y) + H^m(y, x)$,

see [37] for a proof. It can be shown that a consequence of this is that every reversible Markov chain has at least one hidden vertex [38].

Our first goal is to bound the expected meeting time of a virtual process, which we do in the following lemma. For this, we rely heavily on the techniques used in [17]. The main idea is to argue that the expected meeting time function of the virtual process $M^v(x, y)$ satisfies essentially the same recursion as the function $\Phi^m(\cdot, \cdot): V \times V \to \mathbb{R}$ defined by

$$\Phi^m(x, y) = H^m(x, y) + H^m(y, \theta) - H^m(\theta, y), \quad (4)$$

where $\theta$ denotes a fixed hidden vertex of the Metropolis chain.

Lemma 3.8: For all $x, y \in V$, we have $M^v(x, y) \leq 6n^2$.

Proof: Fix nodes $x$ and $y$ such that $x \neq y$, and define

$$\Lambda_x = \sum_{x \in N(x)} \lambda_{xx}, \quad \Lambda_y = \sum_{y \in N(y)} \lambda_{yy}, \quad \Lambda_{xy} = \Lambda_x + \Lambda_y,$$

An immediate consequence of Remark 3.7 is the symmetry of $\Phi^m(\cdot, \cdot)$:

$$\Phi^m(u, v) = \Phi^m(v, u), \quad \forall u, v \in V.$$

Therefore we can argue that

$$\Phi^m(x, y) = \frac{\Lambda_x}{\Lambda_{xy}} \Phi^m(x, y) + \frac{\Lambda_y}{\Lambda_{xy}} \Phi^m(y, x)$$

$$= \frac{\Lambda_x}{\Lambda_{xy}} \left[ H^m(x, y) + H^m(y, \theta) - H^m(\theta, y) \right]$$

$$+ \frac{\Lambda_y}{\Lambda_{xy}} \left[ H^m(y, x) + H^m(x, \theta) - H^m(\theta, x) \right]$$

$$= \frac{\Lambda_x}{\Lambda_{xy}} \left[ \frac{1}{\Lambda_x} + \sum_{x \in N(x)} \frac{\lambda_{xx}}{\Lambda_x} H^m(x, y) \right]$$

$$+ \frac{\Lambda_y}{\Lambda_{xy}} \left[ \frac{1}{\Lambda_y} + \sum_{y \in N(y)} \frac{\lambda_{yy}}{\Lambda_y} H^m(y, x) \right]$$

$$+ \frac{\Lambda_x}{\Lambda_{xy}} \left[ H^m(y, \theta) - H^m(\theta, y) \right]$$

$$+ \frac{\Lambda_y}{\Lambda_{xy}} \left[ H^m(x, \theta) - H^m(\theta, x) \right],$$

where in the second equality we expanded $H^m(x, y)$ and $H^m(y, x)$. Simplifying the last relation, we obtain

$$\Phi^m(x, y) = 2 \frac{\Lambda_x + \lambda_{xx}}{\Lambda_{xy}} H^m(x, y) + \frac{\Lambda_y + \lambda_{yy}}{\Lambda_{xy}} H^m(y, x)$$

$$+ \frac{\Lambda_x}{\Lambda_{xy}} (H^m(y, \theta) - H^m(\theta, y)) + \frac{\Lambda_y}{\Lambda_{xy}} (H^m(x, \theta) - H^m(\theta, x)),$$

(5)

Now, using the definition of $\Phi^m(\cdot, \cdot)$, we have that for each neighbor $x_i \in N(x)$,

$$\Phi^m(x_i, y) = H^m(x_i, y) + H^m(y, \theta) - H^m(\theta, y).$$

Multiplying this relation by $\frac{\lambda_{x_i} \Lambda_{xy}}{\Lambda_{xy}}$ and summing over $x_i \in N(x)$, we obtain

$$\sum_{x_i \in N(x)} \frac{\lambda_{x_i} \Lambda_{xy}}{\Lambda_{xy}} \Phi^m(x_i, y) = \sum_{x_i \in N(x)} \frac{\lambda_{x_i} \Lambda_{xy}}{\Lambda_{xy}} H^m(x_i, y)$$

$$+ \frac{\Lambda_x}{\Lambda_{xy}} \left[ H^m(y, \theta) - H^m(\theta, y) \right]. \quad (6)$$

By the same argument, since for any $y_i \in N(y)$,

$$\Phi^m(x, y_i) = \Phi^m(y_i, x) = H^m(y_i, x) + H^m(x, \theta) - H^m(\theta, x),$$

we have

$$\sum_{y_i \in N(y)} \frac{\lambda_{y_i} \Lambda_{xy}}{\Lambda_{xy}} \Phi^m(x, y_i) = \sum_{y_i \in N(y)} \frac{\lambda_{y_i} \Lambda_{xy}}{\Lambda_{xy}} H^m(y_i, x)$$

$$+ \frac{\Lambda_y}{\Lambda_{xy}} \left[ H^m(x, \theta) - H^m(\theta, x) \right]. \quad (7)$$

Substituting (7) and (6) in (5) gives

$$\Phi^m(x, y) = 2 \frac{\Lambda_x + \lambda_{xx}}{\Lambda_{xy}} \Phi^m(x, y) + \frac{\Lambda_y + \lambda_{yy}}{\Lambda_{xy}} \Phi^m(y, x) + \frac{\Lambda_y}{\Lambda_{xy}} \Phi^m(y, x).$$

(8)

On the other hand, we note that the meeting time of the virtual process can be expanded as

$$M^v(x, y) = \frac{1}{\Lambda_{xy}} \sum_{x \in N(x)} \lambda_{xx} M^v(x, y) + \sum_{y \in N(y)} \frac{\lambda_{yy} \Lambda_{xy}}{\Lambda_{xy}} M^v(x, y).$$

(9)

Note that the above equation holds regardless of whether $x$ and $y$ are neighbors, and this fact is the very reason for our introduction of the virtual process.

We therefore see that $\frac{1}{2} \Phi^m(x, y)$ and $M^v(x, y)$ follow the same recursion formula when $x \neq y$. Thus defining $f(x, y) = \frac{1}{2} \Phi^m(x, y) - M^v(x, y)$ we have that for all $x, y \in V, x \neq y$

$$f(x, y) = \sum_{x \in N(x)} \frac{\lambda_{xx}}{\Lambda_{xy}} f(x, y) + \sum_{y \in N(y)} \frac{\lambda_{yy}}{\Lambda_{xy}} f(x, y).$$

Defining $V' = V \times V$ and $V'' = V' \setminus \{(x, x) \mid x \in V\}$, we have that the matrix $Q$ defined as

$$Q((x, y), (r, s)) = \begin{cases} \frac{\lambda_{xx}}{\Lambda_{xy}}, & \text{if } s = y, r \in N(x) \\ \frac{\lambda_{yy}}{\Lambda_{xy}}, & \text{if } r = x, w \in N(y) \\ 0, & \text{Else.} \end{cases}$$

is a stochastic irreducible matrix in $R^{V' \times V'}$. Furthermore, we have that $f(x, y)$ is harmonic over $V''$ and, since $\theta$ is a hidden vertex, we have that for all $x \in V$, $f(x, x) = \frac{1}{2} (H^m(x, \theta) - H^m(\theta, x)) \geq 0$. Therefore, using Remark 3.5 we immediately get $f(x, y) \geq 0$ for all $x, y \in V$, and therefore $M^v(x, y) \leq \frac{1}{2} \Phi^m(x, y)$. Finally, it was shown in [39] that $H^m(x, y) \leq 6n^2$ for all $x, y \in V$, which now immediately implies the current lemma.
Our next step is to argue that the bound we have just derived on \( \max_{x,y} M^*(x,y) \) in the previous lemma holds for \( \max_{x,y} M^v(x,y) \) with an additional multiplicative factor of 2.

**Lemma 3.9:** \( \max_{x,y} M^v(x,y) \leq 12n^2 \).

**Proof:** The main idea is that, as we have already shown in Lemma 3.8 that \( \max_{x,y \in V} M^v(x,y) \leq 6n^2 \), the virtual process and the original process are identical until the two random walkers are each other’s neighbors, and that when that happens, the probability of meeting at the next transition is \( \frac{\lambda_{uw}}{\lambda_{x,y}} \) in the original process and \( \frac{\lambda_{uw}}{\lambda_{x,y}} \) in the virtual process, and the former is at least half of the latter.

To make this formal, we argue as follows. Fix \( x, y \) with \( x \neq y \) and initialize both an original process and a virtual process with initial positions of the walkers being \( x \) and \( y \). Furthermore, let us couple these processes to move identically until the first time when the walkers are each other’s neighbors; clearly, this has no effect on the distribution of either process. When the walkers are each other’s neighbors for the first time, say at nodes \( u \) and \( w \), let us split the edge connecting them of rate \( 2\lambda_{uw} \) in the virtual process into two distinct edges of rate \( \lambda_{uw} \). We will make the first of these, as well as the edges going from \( u \) and \( w \) to their neighbors, register arrivals in the virtual process exactly when the corresponding edges in the original process register arrivals. Again, this has no effect on the distribution of either process.

Now, for these coupled processes, let us denote by \( T^v \) the first time when the random walkers in the virtual processes meet (recall this means that an edge incident on both of them registers an arrival) and \( T^v \) the first time when the random walkers in the original process meet. We have that \( T^v = T^v \) if the first edge of rate \( \lambda_{uw} \) registers an arrival before the second edge of rate \( \lambda_{uw} \). This happens with probability 1/2. If this does not happen, \( T^v = T^v + 1 \). Thus:\[
M^v(x,y) \leq \frac{1}{2} M^v(x,y) + \frac{1}{2} (M^v(x,y) + \max_{u,w} M^v(u,w))
\]
which implies that \( \max_{x,y} M^v(x,y) \leq 2 \max_{x,y} M^v(x,y) \leq 12n^2 \), where the last step made use of Lemma 3.8.

**Remark 3.10:** Let \( \tau^v(x,y) \) be the meeting time of two random walkers in the original process, who start at \( x \) and \( y \). In view of the previous lemma, we immediately have that:\[
\mathbb{P}(\tau^v(x,y) > t) \leq e^{-\frac{t}{12c^2n^2}}.
\]

Indeed, this follows by dividing the time interval of length \( t \) into \( 12c^2n^2 \) intervals and applying Markov’s inequality in each one. For a rigorous justification, see Chapter 2.4.3 of [37] whose argument applies verbatim here.

Now we are in a position to state the main result of this section, which is an upper bound for the expected convergence time of the quantized Metropolis protocol.

**Theorem 3.11:** The expected time until the quantized Metropolis dynamics reaches a consensus set is \( O(n^2 \log n) \).

**Proof:** Consider the time it takes for the Lyapunov function \( V(t) = \max_i x_i(t) - \min_i x_i(t) \) to shrink by at least 1 starting from any non-consensus configuration. We will next argue that this time is \( O(n^2 \log n) \), which will prove the statement of the theorem.

Indeed, let \( S_{\text{max}} = \arg \max_i x_i(t) \), \( S_{\text{min}} = \arg \min_i x_i(t) \). Note that focusing on the value of a pair of nodes, one node in \( S_{\text{max}} \) and one in \( S_{\text{min}} \), we see that they perform a random walk according to the original process until one of them participates in a non-trivial update. Thus the probability that \( V(t') = V(t) \) for \( t' > t \) is upper bounded by the probability that there exist some \( i \in S_{\text{max}}, j \in S_{\text{min}} \) such that two random walkers in the original process starting from nodes \( i \) and \( j \) have not yet met.

Thus letting \( C \) be the time elapsed until \( V(t) \) shrinks by 1, and recalling our notation \( \tau^v(x,y) \) for the meeting time of two random walkers in the original process starting from \( x \) and \( y \), we have that:

\[
\mathbb{E}(C) = \int_0^\infty \mathbb{P}(C > t) dt \\
\leq \int_0^\infty \min \left\{ 1, \sum_{x,y} \mathbb{P}(\tau^v(x,y) > t) \right\} dt \\
\leq \int_0^\infty \min \left\{ 1, n^2 e^{(1 - \frac{1}{12c^2n^2}) t} \right\} dt \\
= O(n^2 \log n),
\]

where the second inequality is due to Eq. (10) and the last equality follows from

\[
\int_0^\infty \min \left\{ 1, A e^{-at} \right\} dt = \frac{1 + \log A}{a},
\]

which holds when \( A \geq 1 \), and is from [37], Ch. 5.3.2.

**IV. Expected Convergence Time over Time-Varying Networks**

In this section we analyze the expected convergence time of the quantized Metropolis dynamics over time-varying networks. Toward this aim, let us consider a sequence of connected undirected networks \( G(t) = (V,E(t)), t \geq 0 \), over the same set of vertices \( V \) which may change at discrete time instances, i.e., \( G(t) = G((t)), \forall t \geq 0 \). Thus our notation for the set of neighbors of \( x \) will now be \( N_x(t) \), which depends on \( t \). In contrast to the previous section, we will now assume that each node always possesses a self-loop, i.e., \( x \in N_x(t) \) for all \( x \in V, t \geq 0 \). We thus introduce the notation \( N'_x(t) \) to denote the neighbors of node \( x \) excluding itself: \( N'_x(t) = N_x(t) \setminus \{x\} \). The degree of a node \( x \), which we will denote by \( d_x(t) \), will refer to the cardinality of \( N'_x(t) \).
A. Quantized Metropolis Model over Time-Varying Networks

Given a network $G(t)$ at time $t \geq 0$, we associate a Poisson process with each edge \{i, j\} of $G(t)$ with a rate of $\lambda_{ij}(t) = 1/\max(d_i(t), d_j(t))$, i.e., the Metropolis weight corresponding to that edge at time $t$. When an edge \{i, j\} $\in E(t)$ registers an arrival, we let the incident nodes update their values based on Eq. \ref{eq:virtual Metropolis}. Moreover, for each node $x$ and time $t$, the self-loop $(x, x)$ registers arrivals according to a Poisson process with rate of

$$\lambda_{xx}(t) = 1 - \frac{1}{2} \sum_{x \in N_i(t)} \lambda_{xx_i}(t). \quad (12)$$

Note that $\lambda_{xx}(t) \geq 0$ for all $x \in V, t \geq 0$, and the sum of the rates of the edges of $G(t)$ is always equal to $|V| = n$. When a self-loop registers an arrival, no update is made.

B. Preliminary Definitions and Relevant Results

We next introduce the notions of the “original” and “virtual” processes over the time-varying graph sequence $\{G(t)\}$.

**Definition 4.1:** Consider two random walkers moving based on whether the activated edge in the quantized Metropolis location is incident to them or not. That is, if a random walker is at node $i$ at time $t$ when the edge \{i, j\} $\in G(t)$ registers an arrival, the walker moves to node $j$. As before, we will refer to this as the original process, but note that the graph sequence is now time-varying. The meeting time $M^O_{xy}(x, y)$ for $x \neq y$ is the expected time until two random walkers, starting at locations $x$ and $y$ at time $t$, are both incident on an edge which registers an arrival. By convention we set $M^O_{xx}(x, x) = 0$ for all $x \in V, t \geq 0$.

**Definition 4.2:** We define the virtual process to be identical to the original process except when the two walkers in the original process are each other’s neighbors at nodes $u$ and $w$ for some $t \geq 0$, i.e., $u \in N_w(t)$. At this time in the virtual process we let the edges \{r, s\} $\in E(t)$ register arrivals at rates $\mu_{rs}(t)$ defined as follows.

$$\mu_{rs}(t) = \begin{cases} 2\lambda_{uw}(t), & \text{if } \{r, s\} = \{u, w\} \\ \lambda_{uu}(t) - \frac{\lambda_{uw}(t)}{2}, & \text{if } \{r, s\} = \{u, u\} \\ \lambda_{uw}(t) - \frac{\lambda_{uw}(t)}{2}, & \text{if } \{r, s\} = \{w, w\} \\ \lambda_{rs}(t), & \text{else.} \end{cases} \quad (13)$$

Note that $\mu_{rs}(t) \geq 0$ for all $r, s \in V, t \geq 0$. We refer to the meeting time of the virtual process starting from $x$ and $y$ at time $t$ as $\tau^v_{xy}(x, y)$ and its expectation $M^v_{xy}(x, y)$. By convention, we set $M^v_{xx}(x, x) = 0$ for all $x \in V, t \geq 0$.

In words, the rate of the edge \{u, w\} is doubled, while the rate of the self-loops is decreased to make sure that all rates still sum up to $n$.

C. Convergence Rate over Time-Varying Networks

In this part, we state our main results for the quantized Metropolis dynamics over time-varying networks. Define $\Lambda_{xy}(t)$ to be

$$\Lambda_{xy}(t) = \sum_{x_i \in N_x(t)} \lambda_{xx_i}(t) + \sum_{y_i \in N_y(t)} \lambda_{yy_i}(t) = 2 \left( 2 - \lambda_{xx}(t) - \lambda_{yy}(t) \right). \quad (14)$$

Fix $t$ and let $T_1$ be the first time after $t$ that an edge registers an arrival in the virtual process. We then have

$$\mathbb{E} [\tau^v_{xy}(x, y)|T_1] = (T_1 - t) + \left( 1 - \frac{\Lambda_{xy}(T_1)}{n} \right) M^v_{xy}(x, y)$$

$$+ \sum_{x_i \in N_x(t_1)} \frac{\lambda_{xx_i}(T_1)}{n} M^v_{xy}(x, x_i)$$

$$+ \sum_{y_i \in N_y(t_1)} \frac{\lambda_{yy_i}(T_1)}{n} M^v_{xy}(x, y_i). \quad (15)$$

Note that this equation holds regardless of whether $x$ and $y$ are neighbors, and this fact is why we introduced the virtual process in the first place.

Now since $T_1$ is exponential with parameter $n$, we further have that

$$M^v_{xy}(x, y) = \int_t^\infty n e^{-n(t_1 - t)} \mathbb{E} [\tau^v_{xy}(x, y)|T_1 = t_1] dt_1$$

$$= \frac{1}{n} \int_t^\infty n e^{-n(t_1 - t)} \left( 1 - \frac{\Lambda_{xy}(t_1)}{n} \right) M^v_{xy}(x, y) dt_1$$

$$+ \frac{\lambda_{xx}(t_1)}{n} M^v_{xy}(x, x_1)$$

$$+ \frac{\lambda_{yy}(t_1)}{n} M^v_{xy}(x, y_1). \quad (16)$$

Now let us define $v(t)$ to be a column vector of length $n(n-1)$ whose entries are the variables $M^v_{xy}(x, y), \forall x \neq y$ in any order. It follows that we can write the above recursion for $M^v_{xy}(x, y)$ in matrix form as

$$v(t) = \frac{1}{n} I + \int_t^\infty n e^{-n(t_1 - t)} D(t_1) v(t_1) dt_1, \quad (17)$$

where $D(t) \in \mathbb{R}^{n(n-1) \times n(n-1)}$ is a matrix whose rows and columns we will index by $(x, y), x \neq y$ as

$$D_{(x,y),(r,s)}(t) = \begin{cases} \frac{\lambda_{rs}(t)}{n} & \text{if } s = y, r \in N_x(t) \setminus \{x, y\} \\ \frac{\lambda_{uw}(t)}{n} & \text{if } r = x, s \in N_y(t) \setminus \{x, y\} \\ 1 - \frac{\lambda_{xx}(t)}{n} & \text{if } (r, s) = (x, y) \\ 0 & \text{else.} \end{cases} \quad (18)$$

We change variables in \ref{eq:v(t)} from $t_1$ to $z = -e^{-n(t_1 - t)}$ and obtain

$$v(t) = \frac{1}{n} I + \int_{-1}^0 D \left( t - \ln \left( \frac{z}{n} \right) \right) v \left( t - \ln \left( \frac{z}{n} \right) \right) dz. \quad (19)$$

We justify this change of variables by appealing to Theorem 2631 in \cite{40}. Indeed, the equivalence of Eq. \ref{eq:v(t)} and Eq. \ref{eq:v(t)} is an instance of the equality $\int_{-1}^0 g = \int_{-1}^0 g(\phi(z))\phi'(z)$.
Here \( \phi(z) = -e^{-n(z-t)} \) while \( g(z) = D(t-\ln(-z)/n)v(t-\ln(-z)/n) \). Theorem 2631 in [40] allows us to assert this equality subject to (i) \( g(\cdot) \) being Lebesgue measurable; (ii) and \( \phi(\cdot) \) being absolutely continuous on any closed bounded subinterval of \( I \). Item (ii) is clearly satisfied here. Item (i) follows because \( D(\cdot) \) is a symmetric matrix by definition, and \( v(\cdot) \) is a continuous function (indeed, taking \( x \to t \) and conditioning on no transitions in \( [t_n,t) \) as well as no meeting occurring in the same interval, we immediately obtain the continuity of \( v(t) \) as a function of \( t \).

Next, we note that it is immediate from Eq. (13) that \( D(t) \) is a symmetric matrix. Furthermore, it is easy to see that \( D(t) \) is sub-stochastic, implying that its eigenvalues are all real and less than 1 in modulus. Our next step is to upper bound both the largest and smallest eigenvalues of \( D(t) \).

The first step is to extend \( D(t) \) to a stochastic matrix \( P \in \mathbb{R}^{n^2 \times n^2} \) as follows:

\[
P_{(x,y)(r,s)}(t) = \begin{cases} 
1, & \text{if } x = y = r = s \\
\frac{\lambda_{rs}(t)}{n}, & \text{if } x \neq y, s = r, r \in N_x(t) \setminus \{x\} \\
\frac{\lambda_{xy}(t)}{n}, & \text{if } x \neq y, r = x, s \in N_y(t) \setminus \{y\} \\
1 - \frac{\lambda_{xx}(t)}{n}, & \text{if } x \neq y, (r, s) = (x, y), \\
0, & \text{if else.}
\end{cases}
\]

(20)

Note that since \( P(t) \) is stochastic, it can be interpreted as the transition matrix of a Markov chain with absorbing states \( S = \{x, z: x \in V\} \). When we adopt the convention that states in \( S \) correspond to the first \( n \) rows of \( P(t) \), we have that the matrix \( P(t) \) can be represented as

\[
P(t) = \begin{pmatrix} I_{n \times n} & 0 \\ C(t) & D(t) \end{pmatrix},
\]

(21)

where \( C(t) \) and \( D(t) \) are, respectively, matrices of sizes \((n^2 - n) \times n \) and \((n^3 - n) \times (n^2 - n) \).

Example 4.3: Let us consider a line graph of three nodes, \( V = \{1, 2, 3\} \), and the set of edges \( E = \{(1, 2), (2, 3), (1, 1), (2, 2), (3, 3)\} \). In this case we have \( \lambda_{22} = \lambda_{12} = \lambda_{23} = \frac{1}{2} \), and \( \lambda_{11} = \lambda_{33} = \frac{3}{4} \). Then the hitting times are given by:

\[
\begin{pmatrix}
1 & 0 & 0 \\
0 & 1 & 0 \\
0 & 0 & 1 \\
\end{pmatrix}
\]

(1,1) (2,2) (3,3)

Comparing this to Eq. (9) and noting that \( H(t) \{i, i\} = 0 \) for all \( i \in V \), we obtain that \( H(t) \{x, y\} / n \) equals the

where the left-down and right-down blocks represent, respectively, matrices \( D \) and \( C \) in the block form representation given in [21]. As an example, to find \( P(t) \{1, 2, 1, 3\} \), one can either simply use (20) to obtain \( P(t) \{1, 2, 1, 3\} = \frac{4}{6} = \frac{2}{3} \), or to compute the probability of transition of the virtual process in the right-bottom of Figure 1 from state \((1, 2)\) to \((1, 3)\), which is equal to \( P(t) \{1, 2, 1, 3\} = \frac{4}{6} = \frac{2}{3} \). Similarly, to find \( P(t) \{1, 2, 1, 3\} \), we can write \( P(t) \{1, 2, 1, 3\} = 1 - 2\lambda_{13} + \lambda_{23} = \frac{1}{2} \), which is equal to the probability that the virtual process in the right-bottom of Figure 1 does not move, i.e., \( P(t) \{1, 2, 1, 3\} = \frac{1}{2} \).

Fig. 1. Rates in the original and virtual processes in Example 4.3.

**Lemma 4.4:** Consider a Markov chain with transition matrix \( Q = \left( \begin{array}{cc} I_{k \times k} & 0 \\ C & D \end{array} \right) \), which has the additional property that there is a path starting from any node and ending in the first \( k \) nodes. Furthermore, let us assume that \( D \) is symmetric and let us denote its largest eigenvalue by \( \lambda_{\text{max}}(D) \). Let \( H_i \) denote the expected time until the chain is absorbed in \( \{1, \ldots, k\} \) starting at node \( i \) and \( H = \max_i H_i \). Then

\[
\lambda_{\text{max}}(D) \leq 1 - 1/\bar{D}.
\]

**Proof:** Indeed, \( 1/\lambda_{\text{max}}(D) \) is a positive number which is an eigenvalue of \((I - D)^{-1}\) and is consequently upper bounded by \( ||(I - D)^{-1}||_{\infty} \). But by [41], Theorem 4.5, we have that the sum of the entries in the \( i \)th row of \((I - D)^{-1}\) is \( H_i \).

**Lemma 4.5:** For all \( t \geq 0 \),

\[
\lambda_{\text{max}}(D(t)) \leq 1 - \frac{1}{n^2}.
\]

**Proof:** Fix \( t \) and let \( H^P(t) \{x, y\} \) denote the hitting time to \( S = \{i, t\} | i \in V \} \) in \( P(t) \) starting from \( \{x, y\} \). It is immediate that when \( x \neq y \),

\[
H^P(t) \{x, y\} = \frac{n}{A_{xy}(t)} + \sum_{x_i \in N_x(t)} \frac{\lambda_{xx}(t)}{A_{xx}(t)} H^P(t) \{x_i, y\}.
\]

Comparing this to Eq. (9) and noting that \( H^P(t) \{i, i\} = 0 \) for all \( i \in V \), we obtain that \( H^P(t) \{x, y\} / n \) equals the
expected meeting time in the virtual process on the graph $G(t)$ starting from $x$ and $y$. Appealing to Lemma 3.8 we can conclude that expected hitting time in $P(t)$ to the first $n$ states is at most $6n^3$. Appealing to Lemma 4.4 then completes the proof.

Observe that the lower bound

$$\lambda_{\text{min}}(D(t)) \geq 1 - \frac{4}{n}$$  \hspace{1cm} (22)

follows immediately by Gershgorin circles due to the observation that $A_{xy}(t) \leq 2$ for all $x, y \in V, t \geq 0$.

We are now in a position to state and prove the main result of this section.

**Theorem 4.6:** The expected time until the quantized Metropolis dynamics over time-varying connected networks reaches the consensus set is $O(n^2 \log^2 n)$.

**Proof:** Note that we may iterate the recursion of Eq. (19) to obtain

$$v(t) = \frac{1}{n} + \int_{-1}^{0} D \left( t - \frac{\ln(-z)}{n} \right) \left( \frac{1}{n} \right)$$

$$+ \int_{-1}^{0} D \left( t - \frac{\ln(-z)}{n} - \frac{\ln(-z\prime)}{n} \right)$$

$$\cdot v \left( t - \frac{\ln(-z)}{n} - \frac{\ln(-z\prime)}{n} \right) dz d\tau$$

$$\leq \frac{2}{n} + \int_{-1}^{0} \int_{-1}^{0} D \left( t - \frac{\ln(-z)}{n} - \frac{\ln(-z\prime)}{n} \right)$$

$$D \left( t - \frac{\ln(-z)}{n} - \frac{\ln(-z\prime)}{n} \right)$$

$$\cdot v \left( t - \frac{\ln(-z)}{n} - \frac{\ln(-z\prime)}{n} \right) d\tau dz,$$

where the last step used the sub-stochasticity of $D(\cdot)$. Iterating this $k$ times, we obtain

$$v(t) \leq \frac{k}{n} + \int_{-1}^{0} \left[ \prod_{j=1}^{k} D \left( t - \frac{\sum_{i=1}^{j} \ln(-z_i)}{n} \right) \right]$$

$$v \left( t - \frac{\sum_{i=1}^{k} \ln(-z_i)}{n} \right) dz_k \ldots dz_1.$$

(23)

Now let us use introduce the notation $M = \sup_{t > 0} \|v(t)\|$. Elementary arguments suffice to establish that $M < \infty$; recall that $v(t)$ stacks up expected meeting times in the virtual process starting at time $t$, and the finiteness of $M$ can follow from the observation that the probability of intersection in any positive length interval $[t, t+\alpha]$ is positive and bounded away from zero independently of $t$. Thus for every $\epsilon > 0$ there exists $t^*$ such that $M - \epsilon \leq \|v(t^*)\|$. Since the entries of $D(\cdot)$ and $v(\cdot)$ are always nonnegative, we can take the infinity-norm from both sides of (23) and use triangle inequality to obtain

$$M - \epsilon \leq \|v(t^*)\| \leq \frac{k}{n} + \int_{-1}^{0} \int_{-1}^{0} \left[ \prod_{j=1}^{k} D \left( t - \frac{j \ln(-z_i)}{n} \right) \right]$$

$$v \left( t - \frac{j \ln(-z_i)}{n} \right) dz_k \ldots dz_1,$$

where in the fourth inequality we have used the fact that the infinity-norm of a matrix is always upper bounded by its induced 2-norm times the square root of its dimension, i.e., $\|D(\cdot)\|_\infty \leq \sqrt{n(n-1)}\|D(\cdot)\|_2$. Moreover, the last inequality is valid due to Lemma 4.5 and Eq. (22).

Let us choose $k^*$ so that

$$\sqrt{n(n-1)} \left( 1 - \frac{1}{6n^3} \right)^k \leq \frac{1}{2}.$$

Appealing to the inequality $(1 - 1/x)^x \leq e^{-1}$, we obtain that we may choose $k^* = O(n^3 \log n)$ to accomplish this. Plugging this into the last line of Eq. (24), we immediately obtain

$$M - \epsilon \leq \frac{k^*}{n} + M.$$

Since this holds for any $\epsilon > 0$, it implies that $M \leq 2k^*/n = O(n^2 \log n)$.

We have thus obtained an upper bound on $O(n^2 \log n)$ on the expected meeting time in the virtual process starting from any pair of nonidentical nodes and any time $t$. The rest of the proof directly parallels the arguments we have made in the fixed graph case, and we only sketch it rather than repeat the relevant sections verbatim. The first step is to argue that $2M$ is an upper bound on the meeting time of the original process. The proof proceeds exactly as in the case of Lemma 3.9 by coupling the two processes and conditioning on the transition at the last step in the meeting time of the virtual process. The next (and final) step is to argue that the time it takes $V(t) = \max_{x} x(t) - \min_{x} x(t)$ to shrink by 1 is upper bounded by $O(n^2 \log^2 n)$. This follows exactly as in the proof of Theorem 3.11. Indeed, we can argue that if $V(t') = V(t)$ at some time $t' > t$, this means that a pair of nodes performing a random walk according to the original process have not met between times $t$ and $t'$. Upper bounding the latter using the union bound, we once again obtain that the expected time it takes for $V(t)$ to shrink by 1 is the expected meeting time in the virtual process times a multiplicative factor of $O(\log n)$. 

$\blacksquare$
V. Expected Convergence Time Over Specific Networks

In this section, we provide some improved bounds for the expected convergence time of the quantized Metropolis protocol on some concrete networks. In several cases we obtain bounds which are an order-of-magnitude better than the essentially quadratic convergence times derived in the previous sections.

Our main technical tool will be to rely on the analogy between hitting times and expected resistances in the network. Indeed, recall that the quadratic bounds obtained in the earlier section were all consequences of the result from [39] that hitting times in the Metropolis chain are $O(n^2)$ on any graph. To obtain improved bounds, we only need to get improved upper bounds on hitting times. For example, if we show that the hitting times in a certain class of graphs are $O(n)$, then repeating our proof of Theorem [3.11] verbatim, we obtain that the convergence time of quantized Metropolis consensus is

$$\left(\max_{a,b\in V} R(a \leftrightarrow b)\right) O(\log n)$$

where $R(a \rightarrow b)$ is the resistance between $a$ and $b$ in the graph where the edge $(i,j)$ has resistance $n \max(d(i),d(j))$.

We next apply this fact to a variety of graphs.

- **Line Graph**
  For the case of line graph, we always have $\max\{d(x),d(y)\} = 2, \forall \{x,y\} \in \mathcal{E}$. Therefore, in Corollary 5.2, the resistance of every edge of the electrical circuit associated with the Metropolis chain for the line graph is $2n$. Hence $\max_{a,b\in V} R(a \leftrightarrow b) = 2n(n-1)$. In particular, $\max_{a,b\in V} H^m(a, b) = O(n(n-1))$. The bound we obtain in this case is essentially identical to the bound of Theorem 3.11 by a multiplicative factor of $n/\log n$.

- **2-Dimensional Grids**
  A well-known result for the 2-dimensional grids (Proposition 9.16, [36]) states that the largest resistance between any pair of nodes of an $n \times n$ grid where each edge has a unit size resistance is bounded from above by $2\log(n)$. For the case of Metropolis chain over the 2D grid, $\max\{d(x),d(y)\} \leq 4, \forall \{x,y\} \in \mathcal{G}$. By Corollary 5.2, this allows us to bound hitting times in the 2D grid as $O(n \log n)$. We thus obtain a convergence time bound of $O(n \log^2(n))$ for the quantized Metropolis protocol, which is better than the bound of Theorem 3.11 by a multiplicative factor of $n/\log n$.

- **3-Dimensional Grids**
  For 3-dimensional grids it is known that the largest resistance of an $n \times n \times n$ grid with unit resistances on each edge is bounded from above by a universal constant independent of $n$. Using the same line of argument as 2D grids, one can see that $\max\{d(x),d(y)\} \leq 6$, which in turn means that the maximum expected hitting time of the Metropolis chain over 3D grids is at most $O(n)$. We thus obtain a convergence time bound of $O(n \log n)$ for the quantized Metropolis protocol, which is better than the bound of Theorem 3.11 by a multiplicative factor of $n$.

- **Star Graph**
  For the star graph (a tree with a root and $n-1$ leaves), we have $\max\{d(x),d(y)\} = n-1, \forall \{x,y\} \in \mathcal{G}$. This shows that $\max_{a,b\in V} R(a \leftrightarrow b) = n(n-1)$. This leads to bounds which are essentially the same as the bounds of Theorem 3.11.

This may appear somewhat counter-intuitive at first glance, since the simple random walk on the star graph is easily seen to have a linear hitting time. However, the Metropolis chain attains symmetry (which is crucial...
for consensus) by putting large self-loops at the leaves, which leads to a much slower hitting time.

**Lollipop Graph**

For the lollipop graph (two clusters of size \( \lfloor \frac{n}{2} \rfloor \) which are connected by a line graph of \( n - 2 \lfloor \frac{n}{2} \rfloor \) nodes), one can argue that \( \max \{ d(x), d(y) \} \leq \lfloor \frac{n}{2} \rfloor \) for the edges belonging to the clusters and \( \max \{ d(x), d(y) \} \leq 2 \) for the edges belonging to the path. This implies that \( \max_{a,b \in V} R(a \leftrightarrow b) = O(n^2) \), once again leading to bounds which are identical to those of Theorem 3.11.

**Random Geometric Graph**

A random geometric graph \( G(n, r) \) is a graph which is obtained by distributing \( n \) nodes uniformly on the unit square and connecting two nodes if and only if the Euclidean distance between these two nodes is at most \( r \). Such graphs are popular in modeling wireless ad-hoc and sensor networks [42].

It was shown in [42] (Lemma 2.13 and Sections 2.3-2.5.2) that for \( c > 1 \), \( r^2 = c8 \log n \) is sufficient to guarantee that with high probability \( G(n, r) \) is “geodense,” roughly meaning that the geometric graph has almost uniform node density across the unit square. In particular, for \( r^2 \geq c8 \log n \) all nodes have degree of \( O(nr^2) \) with high probability. Furthermore, for this range of \( r \) and by Theorem 2.20 in [42], the maximum resistance between two nodes of a geometric graph when each node has a unit resistance is \( O(\frac{1}{nr^2}) \) with high probability. Putting these two results together and in view of Corollary 5.2 we can see that for a random geometric graph with \( r^2 \geq c8 \log n \), \( c > 1 \) we have \( \max_{a,b \in V} H(a,b) = O(n \times nr^2 \times \frac{1}{nr^2}) = O(n) \). Therefore, the expected convergence time of the quantized Metropolis protocol for this range of \( r \) is at most \( O(n \log n) \) with high probability.

VI. SIMULATIONS

In this section, we report on the results of some simulations on the performance of the quantized Metropolis method. Since the key bottleneck in performance is the expected time until a nontrivial update takes place, here we compare this time to the quadratic upper bounds we have derived for a variety of graphs.

More precisely, we consider the ratio of \( \frac{T(G)}{n^2} \) for four simple classes of graphs \( G \), namely, line, star, lollipop and 3-regular graphs, as well as Erdos-Renyi graphs. We have depicted our simulation results in Figure 2 while we let the number of nodes vary from 1 to 70 for each of these graphs.

For the case of the Erdos-Renyi graph, we assume that the probability that each edge appears in the graph is \( \frac{1}{2} \) and independent of the other edges. As it can be seen in Figure 2, the ratio of \( \frac{T}{n^2} \) is bounded from above by 0.5 and vanishes very fast, corresponding to a sub-quadratic convergence time. For the case of the lollipop graph, we consider two cliques, each of size \( \lfloor \frac{n}{4} \rfloor \) which are connected by a path of the remaining nodes. As can be seen again in Figure 2, the ratio of \( \frac{T}{n^2} \) for each of the line graph, lollipop and star graph is bounded from above by 0.5 and asymptotically converges to a constant, meaning that our quadratic upper bounds are essentially sharp for these graphs.

Finally, in the case of 3-regular graphs, we consider a cycle of \( n \) nodes where each node is connected to its diagonally opposite node on the cycle (this is possible when \( n \) is an even number and when \( n \) is odd, we leave the final node \( n \) to be of degree 2). Again, it can be seen from Figure 2 that the quantized Metropolis dynamics display good performance on such graphs.

Next we consider the expected time for the occurrence of a nontrivial update over various sequences of time-varying graphs. For this purpose, we consider five different sequences of time-varying networks, all having the property that graph changes can only occur at discrete time instances \( k = 1, 2, \ldots \). In particular, we consider:

- (I) An alternating sequence of line and star networks, i.e.,
  \[
  G(k) = \begin{cases} 
  \text{line graph}, & \text{if } k \text{ is even} \\
  \text{star graph}, & \text{if } k \text{ is odd}.
  \end{cases}
  \]

- (II) An alternating sequence of line, star and lollipop networks, i.e.,
  \[
  G(k) = \begin{cases} 
  \text{lollipop graph}, & \text{if } \text{mod}(k, 3) = 0 \\
  \text{line graph}, & \text{if } \text{mod}(k, 3) = 1 \\
  \text{star graph}, & \text{if } \text{mod}(k, 3) = 2.
  \end{cases}
  \]

- (III) A random sequence of line and star networks generated by tossing an unbiased coin, i.e.,
  \[
  G(k) = \begin{cases} 
  \text{line graph}, & \text{w.p. } \frac{1}{2} \\
  \text{star graph}, & \text{w.p. } \frac{1}{2}.
  \end{cases}
  \]

- (IV) A sequence of Erdos-Renyi graphs with \( \frac{1}{2} \) probability of emerging an edge in each graph.
- (V) A uniformly at random generated sequence of line, star, lollipop and Erdos-Renyi graphs, i.e.,

\[
G(k) = \begin{cases} 
\text{line graph}, & \text{w.p.}\ \frac{1}{4} \\
\text{lollipop graph}, & \text{w.p.}\ \frac{1}{4} \\
\text{star graph}, & \text{w.p.}\ \frac{1}{4} \\
\text{Erdos-Renyi graph}(p = \frac{1}{2}), & \text{w.p.}\ \frac{1}{4} 
\end{cases}
\]

In each of the above cases we let the number of nodes vary from 1 to 50. We let the length of the sequence of the time-varying graphs to be large enough such that in all the sample paths we generate, at least one nontrivial update takes place (this ended up resulting in a length of 5,000). We ran the quantized Metropolis protocol over these sequences of time-varying networks 20,000 times and take an average of the waiting times for the occurrence of a nontrivial update. We initialized the process with an initial condition of two starting nodes for the values 0 and 2, while all the rest have value 1; furthermore, we simulated this for every possible pair of starting nodes for the values 0 and 2 and take the maximum. The results are depicted in the Figure 3.

In the first three cases (I, II and III), the curve shows a quadratic behavior in the number of nodes, matching the bounds of Theorem 4.6. For the two other cases (IV, V), the curve appears to be linear, once again confirming the fast performance of Metropolis consensus over Erdos-Renyi graphs.

![Graph showing the maximum expected time that a nontrivial update takes place and the upper bound of $n^2$ for time-varying networks.](image)

Fig. 3. Comparison between the maximum expected time that a nontrivial update takes place and the upper bound of $n^2$ for time-varying networks.

VII. CONCLUSION

We have studied the quantized consensus problem on undirected connected networks in both static and time-varying settings. Using Metropolis chains within the structure of quantized consensus protocols, we were able to improve the expected convergence time relative to the previous work in the literature. In particular, we have proved an upper bound of $O(n^2 \log n)$ for the convergence time of quantized Metropolis dynamics over static networks and an upper bound of $O((n^2 \log^2 n)$ over dynamic networks.

A future direction of research would be to improve convergence times further. For example, [43] attained a linear convergence time for consensus on any fixed graph, and it is an open question to obtain a quantized consensus protocol which replicates this. A related problem is to extend the quadratic convergence times obtained in this paper to dynamic networks which are not necessarily connected at every time, but rather only connected in a long-term sense. Furthermore, working out the implications of these and related results for network design is of considerable interest. For example, if a network designer had the ability to add a fixed number of edges, it is unclear at the present what the most efficient procedure to determine the best locations is.

REFERENCES

[1] T. Başar, S. R. Etesami, and A. Olshevsky, “Fast convergence of quantized consensus using metropolis weights,” proc. 53rd IEEE Conference on Decision and Control (CDC’14, Dec 15-17, 2014; Los Angeles, CA).

[2] J. N. Tsitsiklis, D. P. Bertsekas, and M. Athans, “Distributed asynchronous deterministic and stochastic gradient optimization algorithms,” IEEE Transactions on Automatic Control, vol. 31, no. 9, pp. 803–812, 1986.

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

[4] C. Gao, J. Cortes, and F. Bullo, “Notes on averaging over acyclic digraphs and discrete coverage control,” Automatica, vol. 44, no. 8, pp. 2120–2127, 2008.

[5] W. Ren and R. W. Beard, Distributed Consensus in Multi-Vehicle Cooperative Control. London: Springer, 2008.

[6] A. Olshevsky, “Efficient information aggregation strategies for distributed control and signal processing,” Ph. D. thesis, arXiv:1009.6036, 2010.

[7] M. Rabbat and R. D. Nowak, “Distributed optimization in sensor networks,” in Information Processing and Sensor Networks. IEEE, 2004, pp. 29–27.

[8] L. Xiao, S. Boyd, and S.-J. Kim, “Distributed average consensus with least-mean-square deviation,” Journal of Parallel and Distributed Computing, vol. 67, no. 1, pp. 33–46, 2007.

[9] S. Ram, V. Veeravalli, and A. Nedic, “Distributed non-autonomous power control through distributed convex optimization,” in INFOCOM. IEEE, 2008, pp. 3001–3005.

[10] B. Ghosh and S. Muthukrishnan, “Dynamic load balancing in parallel and distributed networks by random matchings,” in Proceedings of the sixth annual ACM symposium on Parallel algorithms and architectures. ACM, 1994, pp. 226–235.

[11] G. Neglia, G. Reina, and S. Alouf, “Distributed gradient optimization for epidemic routing: a preliminary evaluation,” in Wireless Days, 2nd IFIP. IEEE, 2009, pp. 1–6.

[12] D. P. Bertsekas and J. N. Tsitsiklis, Parallel and Distributed Computation. Prentice Hall Inc, 1989.

[13] L. Xiao and S. Boyd, “Fast linear iterations for distributed averaging,” Systems and Control Letters, p. 6578, 2004.

[14] S. Boyd, A. Ghosh, B. Prabhakar, and D. Shah, “Gossip algorithms: Design, analysis and applications,” in INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE, vol. 3, IEEE, 2005, pp. 1653–1664.

[15] M. Draief and M. Vojnovic, “Convergence speed of binary interval consensus,” SIAM Journal on Control and Optimization, vol. 50, no. 3, pp. 1087–1109, 2012.

[16] A. Kashyap, T. Başar, and R. Srikant, “Quantized consensus,” Automatica, pp. 1192–1203, 2007.

[17] S. R. Etesami and T. Başar, “Convergence time for unbiased quantized consensus,” Proc. 52nd IEEE Conference on Decision and Control (CDC’13, Dec 10-13, 2013; Florence, Italy), pp. 6190–6195, 2013.

[18] P. F. R. Carli, F. Fagnani, and S. Zampieri, “Gossip consensus algorithms via quantized communication,” Automatica, vol. 46, pp. 70–80, 2010.

[19] M. Zhu and S. Martínez, “On the convergence time of asynchronous distributed quantized averaging algorithms,” Automatic Control, IEEE Transactions on, vol. 56, no. 2, pp. 386–390, 2011.
[20] R. Carli, F. Fagnani, P. Frasca, and S. Zampieri, “Gossip consensus algorithms via quantized communication,” *Automatica*, vol. 46, no. 1, pp. 70–80, 2010.

[21] K. Cai and H. Ishii, “Quantized consensus and averaging on gossip digraphs,” *Automatic Control, IEEE Transactions on*, vol. 56, no. 9, pp. 2087–2100, 2011.

[22] F. Bénetzì, P. Thiran, and M. Vetterli, “Interval consensus: from quantized gossip to voting,” in *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*. IEEE, 2009, pp. 3661–3664.

[23] R. Olfati-Saber and R. M. Murray, “Consensus problems in networks of agents with switching topology and time-delays,” *Automatic Control, IEEE Transactions on*, vol. 49, no. 9, pp. 1520–1533, 2004.

[24] F. Paqualetti, D. Borra, and F. Bullo, “Consensus networks over finite fields,” *Automatica*, vol. 50, no. 2, pp. 349–358, 2014.

[25] S. Shang, P. Cuff, S. Kulkarni, and P. Hui, “An upper bound on the convergence time for distributed binary consensus,” in *Information Fusion, 15th International Conference on*. IEEE, 2012, pp. 369–375.

[26] K. Cai and H. Ishii, “Convergence time analysis of quantized gossip consensus on digraphs,” *Automatica*, vol. 48, pp. 2344–2351, 2012.

[27] R. Carli, F. Bullo, and S. Zampieri, “Quantized average consensus via dynamic coding/decoding schemes,” *International Journal of Robust and Nonlinear Control*, vol. 20, no. 2, pp. 156–175, 2010.

[28] T. Li and L. Xie, “Distributed coordination of multi-agent systems with quantized-observer based encoding-decoding,” *IEEE Transactions on Automatic Control*, vol. 57, no. 12, pp. 3023–3027, 2012.

[29] M. Guo and D. Dimarogonas, “Consensus with quantized relative state measurements,” *Automatica*, vol. 49, no. 8, pp. 2531–2537, 2013.

[30] J. Lavaei and R. M. Murray, “Quantized consensus by means of gossip algorithm,” *Automatic Control, IEEE Transactions on*, vol. 57, no. 1, pp. 19–32, 2012.

[31] A. Olshevsky, “Consensus with ternary messages,” *SIAM Journal on Control and Optimization*, vol. 52, no. =2, pp. 987–1009, 2014.

[32] M. Zhu and S. Martinez, “On the convergence time of asynchronous distributed quantized averaging algorithms,” *IEEE Transactions on Automatic Control*, vol. 56, no. 2, pp. 386–390, 2011.

[33] S. R. Etesami and T. Başar, “Convergence time for unbiased quantized consensus over static and dynamic networks,” *arXiv:1403.4109v1*, 2014.

[34] S. Shang, P. Cuff, P. Hui, and S. Kulkarni, “An upper bound on the convergence time for quantized consensus,” in *INFOCOM, 2013 Proceedings IEEE*. IEEE, 2013, pp. 600–604.

[35] J. M. Hendrickx, A. Olshevsky, and J. N. Tsitsiklis, “Distributed anonymous discrete function computation,” *Automatic Control, IEEE Transactions on*, vol. 56, no. 10, pp. 2276–2289, 2011.

[36] D. A. Levin, Y. Peres, and E. L. Wilmer, *Markov Chains and Mixing Times*. American Mathematical Society, 2009.

[37] D. Aldous and J. Fill, “Reversible markov chains and random walks on graphs,” 2002.

[38] D. Coppersmith, P. Tetali, and P. Winkler, “Collisions among random walks on a graph,” *SIAM J. on Discrete Mathematics*, vol. 6, pp. 363–374, 1993.

[39] Y. Nonaka, H. Ono, K. Sadakane, and M. Yamashita, “The hitting and cover times of metropolis walks,” *Theoretical Computer Science*, vol. 411, no. 16, pp. 1889–1894, 2010.

[40] D. Fremlin, “Measure Theory,” vol. 2, ISBN 978-0-9538129-7-4.

[41] E. Seneta, *Non-negative matrices and Markov chains*. Springer, 2006.

[42] C. Avis, “Random geometric graphs: an algorithmic perspective,” Ph.D. dissertation, University of California Los Angeles, 2006.

[43] A. Olshevsky. (2014) Linear time average consensus on fixed graphs and implications for decentralized optimization and multi-agent control. [Online]. Available: http://arxiv.org/abs/1411.4186