Distance-dependent Kronecker Graphs for Modeling Social Networks

Elizabeth Bodine-Baron,* *Member, IEEE,* Babak Hassibi, *Member, IEEE,*
and Adam Wierman, *Member, IEEE*

***Abstract*—This paper focuses on a generalization of stochastic Kronecker graphs, introducing a Kronecker-like operator and defining a family of generator matrices $\mathcal{H}$ dependent on distances between nodes in a specified graph embedding. We prove that any lattice-based network model with sufficiently small distance-dependent connection probability will have a Poisson degree distribution and provide a general framework to prove searchability for such a network. Using this framework, we focus on a specific example of an expanding hypercube and discuss the similarities and differences of such a model with recently proposed network models based on a hidden metric space. We also prove that a greedy forwarding algorithm can find very short paths of length $O((\log\log n)^2)$ on the hypercube with n nodes, demonstrating that distance-dependent Kronecker graphs can generate searchable network models. EDICS: OTH-EMRG, SEN-APPL, SPC-APPL**

I. Introduction

Beginning with the simple Erdös-Rényi model of random networks [1], network science has attempted to capture the key characteristics of complex networks such as power networks, the Internet, protein interaction networks, and social networks with a simple, mathematically tractable model.[1] Social networks in particular have generated much interest due to the consistency of their characteristics. These networks tend to exhibit small diameter, high clustering, scale-free degree distributions, and perhaps most importantly, they are searchable by a local greedy algorithm; see [3], [4], and [5] for thorough surveys of this area.

The Erdös-Rényi random graph maintains a small diameter but fails to capture many of the other key properties [6], [1]. The combination of small diameter and high clustering is often called the "small-world effect," and Watts and Strogatz (see section III) generated much interest when they proposed a model that maintains these two characteristics simultaneously [7]. Several models were then proposed to explain the heavy-tailed degree distributions and densification of complex networks; these include the preferential attachment model [8], the forest-fire model [9], [10], Kronecker graphs [11], [12], and many others [3]. As demonstrated by Milgram's 1967 experiment using real people, individuals can discover and use short paths using only local information [13]. Kleinberg focuses on this searchability characteristic in his lattice model and proves searchability for a precise set of input parameters, but his model lacks any heavy-tailed distributions [14], [5], [15]. The Kronecker graphs described in [11], [12], and [16] are simple to generate, mathematically tractable, and have been shown to exhibit several important social network characteristics such as heavy-tailed degree and eigen-distributions, high clustering, small diameter, and network densification. However, Kronecker graphs are not searchable by a distributed greedy algorithm [16].

In this paper, we extend the model proposed in [2], a generalization of stochastic Kronecker graphs that can generate searchable networks. Instead of using the traditional Kronecker operation, we introduce a new "Kronecker-like" operation and a family of generator matrices, $\mathcal{H}$, both dependent upon the distance between two nodes. This new generation method yields networks that have both a local (lattice-based) and global (distance-dependent) structure. This dual structure is what allows a greedy algorithm to search the network using only local information. Additionally, the networks generated have a high clustering (due to the lattice structure) and a small diameter (due to the addition of long-range links).

As part of the analysis of this new model, we provide a general framework for analyzing degree distributions and the performance of greedy search algorithms on a general lattice-based network. We use this framework to study one example in detail: an expanding hypercube with distance-dependent long-range connections. We give an explicit description of its degree distribution, the circum-

E. Bodine-Baron is with the Electrical Engineering Department at the California Institute of Technology, MC 136-93, 1200 E. California Boulevard, Pasadena, CA 91125. Phone: (214) 797-0737 Fax: (626) 564-9307 Email: eabodine@caltech.edu

B. Hassibi is with the Electrical Engineering Department at the California Institute of Technology, MC 136-93, 1200 E. California Boulevard, Pasadena, CA 91125. Phone: (626) 395-4810 Fax: (626) 564-9307 Email: hassibi@caltech.edu

A. Wierman is with the Computer Science Department at the California Institute of Technology, MC 305-16, 1200 E. California Boulevard, Pasadena, CA 91125. Phone: (626) 395-6569 Fax: (626) 792-4257 Email: adamw@caltech.edu

[1]A preliminary version of many of the results of this paper first appeared in [2]

stances under which it will be searchable by a local greedy algorithm, and a lower bound on its diameter. We support our findings with simulations. This example is chosen because it mimics the defining feature of tree metrics and hyperbolic space – exponentially expanding neighborhoods – which are thought to be representative of both the Internet and social networks [17], [18], [19], [20]. Exponentially expanding neighborhoods lead to very small diameters ($O(\log\log n)$ as opposed to $O(\log n)$) and we can show that, as in [21], a local greedy algorithm on the hypercube will find ultra-short paths, $O((\log\log n)^2)$.

This paper is organized as follows. Section II briefly defines some key concepts frequently used in social network literature. Section III describes in detail our model and its relation to the original Kronecker graph model and other traditional models. Section IV explores the connection between a Kleinberg-like expanding hypercube example and the hidden metric space models proposed in [17]. Section V describes a general analysis of degree distributions for lattice-based networks and gives a theorem showing that all such networks will have a Poisson degree distribution provided that $P(d)$ is sufficiently small, and gives the relevant degree distribution for the expanding hypercube example. Section VI gives a general framework for proving searchability of a lattice-based distance-dependent network model and recovers the searchability result of [14] and finally proves that the expanding hypercube is in fact searchable. Section VII explores the diameter of the expanding hypercube example and Section VIII concludes with proposed future work. The appendices support the proof of searchability for the expanding hypercube example in Section VI.

II. Preliminaries

Before continuing further, it will be useful to define several terms commonly used in social network literature. A social network is represented by a graph $G = (V, E)$, where V and E are the sets of vertices and edges, respectively. There is one vertex for each agent, or person, in the network, and the edges represent the relationships between individuals. These relationships can be summarized in an adjacency matrix A where

$$A_{ij} = \begin{cases} 1 & \text{if nodes } i \text{ and } j \text{ are connected} \\ 0 & \text{otherwise.} \end{cases}$$

We note that while we will be working with undirected and unweighted graphs, in general, the edges in an adjacency matrix representing a social network can be both directed and weighted, showing the direction and the values of different relationships. The *neighborhood* $\mathcal{N}_i$ of a node i is defined as the set of its immediately connected neighbors. The *degree* k_i of a node is defined as the size of its neighborhood. We define the *geodesic* between two nodes u and v as the shortest path connecting them. The *diameter* of a network, for our purposes, is the length of the maximum geodesic for that network. Note that in some cases, what is meant by diameter is the average of all geodesics; however, for this paper we focus on the maximum. In social and most complex networks, the diameter of the network grows logarithmically with the number of nodes int the network [7], [22]. Another useful and commonly used term is clustering, which measures the amount of community structure present in a network. For an individual node, we define a *clustering coefficient* C_i where

$$C_i = \frac{2 \mid \{e_{jk}\} \mid}{k_i(k_i - 1)} : v_j, v_k \in \mathcal{N}_i, e_{jk} \in E$$

The clustering coefficient for the entire graph is then the average of the clustering coefficients over all n nodes [7].

$$\bar{C} = \frac{1}{n}\sum_{i=1}^{n} C_i$$

Finally, we call a network *searchable* if a distributed search algorithm can find paths through the network of length on the order of the diameter. For example, in Kleinberg's lattice model, a network has diameter $O(\log n)$, and is called searchable if a distributed algorithm can find paths of length $O((\log n)^2)$ [14]. For more details on the distributed search algorithm, see section VI.

III. Distance-dependent Kronecker graphs

In this section we describe the original formulation of stochastic Kronecker graphs as well as our new "distance"-dependent extension of the model. We then present a few examples illustrating how to generate existing network models using the "distance"-dependent Kronecker graph.

A. Stochastic Kronecker Graphs

Stochastic Kronecker graphs[2] are generated by recursively using a standard matrix operation, the Kronecker product [11]. Beginning with an initiator probability matrix P_1, with N_1 nodes, where the entries p_{ij} denote the probability that edge (i, j) is present, successively larger graphs $P_2, \ldots, P_n$ are generated such that the k^{th} graph P_k has $N_k = N_1^k$ nodes. The Kronecker product is used to generate each successive graph.

Definition 3.1: The k^{th} power of P_1 is defined as the matrix $P_1^{\otimes k}$, such that:

$$P_1^{\otimes k} = P_k = \underbrace{P_1 \otimes P_1 \otimes \ldots P_1}_{k\ times} = P_{k-1} \otimes P_1$$

[2]For a description of deterministic Kronecker graphs, see Leskovec et al, [11].

For each entry p_{uv} in P_k, include an edge in the graph G between nodes u and v with probability p_{uv}. The resulting binary random matrix is the adjacency matrix of the generated graph.

Kronecker graphs have many of the static properties of social networks, such as small diameter and a heavy-tailed degree distribution, a heavy-tailed eigenvalue distribution, and a heavy-tailed eigenvector distribution [11]. In addition, they exhibit several temporal properties such as densification and shrinking diameter. Using a simple 2x2 P_1, Leskovec demonstrated that he could generate graphs matching the patterns of the various properties mentioned above for several real-world datasets [11]. However, as shown by Mahdian and Xu, stochastic Kronecker graphs are not searchable by a distributed greedy algorithm [16] – they lack the necessary spatial structure that allows a local greedy agent to find a short path through the network. This is the motivation for the current paper.

B. Distance-Dependent Kronecker Graphs

In this section, we propose an extension to Kronecker graphs incorporating the spatial structure necessary to have searchability. We add to the framework of Kronecker graphs a notion of "distance", which comes from the embedding of the graph, and extend the generator from a single matrix to a family of matrices, one for each distance, defining the likelihood of a connection occurring between nodes at a particular "distance." We accomplish this with a new "Kronecker-like" operation. Specifically, whereas in the original formulation of Kronecker graphs one initiator matrix is iteratively Kronecker-multiplied with itself to produce a new adjacency or probability matrix, we define a "distance"-dependent Kronecker operator. Depending on the distance between two nodes u and v, $d(u,v) \in \mathbb{Z}$, a different matrix from a defined family will be selected to be multiplied by that entry, as shown below.

$$\mathbf{C} = \mathbf{A} \otimes_d \mathcal{H} = \begin{pmatrix} a_{11}H_{d(1,1)} & a_{12}H_{d(1,2)} & \dots & a_{1n}H_{d(1,n)} \\ a_{21}H_{d(2,1)} & a_{22}H_{d(2,2)} & \dots & a_{2n}H_{d(2,n)} \\ \vdots & \vdots & \ddots & \vdots \\ a_{n1}H_{d(n,1)} & a_{n2}H_{d(n,2)} & \dots & a_{nn}H_{d(n,n)} \end{pmatrix}$$

where

$$\mathcal{H} = \{H_i\}_{i \in \mathbb{Z}}$$

So, the k^{th} Kronecker power is now

$$G_k = \underbrace{G_1 \otimes_d \mathcal{H} \cdots \otimes_d \mathcal{H}}_{k\ times}$$

In the Kronecker-like multiplication, the choice of H_i from the family $\mathcal{H}$, multiplying entry (u,v), is dependent upon the distance $d(u,v)$. Note that our $d(u,v)$ is not a true distance measure — we can have negative distances. Further, $d(u,v)$ is not symmetric $(d(u,v) \neq d(v,u))$ since we need to maintain symmetry in the resulting matrix. Instead, $d(u,v) = -d(v,u)$ and $H_{d(u,v)} = H'_{d(v,u)}$.

This change to the Kronecker operation makes the model more complicated, and we do give up some of the beneficial properties of Kronecker multiplication. Potentially, we could have to define a large number of matrices for $\mathcal{H}$. However, for the models we want to generate, there are actually only a few parameters to define, as $d(i,j)$ and a simple function defines H_i for $i > 1$. The underlying reason for this simplicity is that the local lattice structure is usually specified by H_0 and H_1, while the global, distance-dependent probability of connection can usually be specified by an H_i with a simple form. So, while we lose the benefits of true Kronecker multiplication, we gain generality and the ability to create many different lattices and probability of long-range contacts. We note in passing that the generation of these lattice structures is not possible with the original formulation of the Kronecker graph model. For example, it is impossible to generate the Watts-Strogatz model with conventional Kronecker graphs. However, it can be done with the current generalization. This is illustrated in our examples below.

Example 1: Original Kronecker Graph. The simplest example is that of the original Kronecker graph formulation. For this case, the "distance" can be arbitrary, and the family of matrices, $\mathcal{H}$, is simply G_1, the same G_1 used in the original definition. Thus, we define

$$G_k = \underbrace{G_1 \otimes_d \mathcal{H} \cdots \otimes_d \mathcal{H}}_{k\ times} = \underbrace{G_1 \otimes G_1 \otimes \dots G_1}_{k\ times}$$

Example 2: Watts-Strogatz Small-World Model. The next example we consider, the Watts-Strogatz model, consists of a ring of n nodes, each connected to their neighbors within distance k on the ring. The probability of a connection to any other node on the ring is then $P(u,v) = p$ [7]. To generate the underlying ring structure with $k = 1$, start with an initiator matrix K_1, representing the graph in figure 1(a).

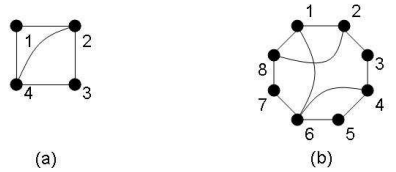

Fig. 1. Generating the Watts-Strogatz Model

In order to obtain the sequence of matrices repre-

senting the graphs in Figure 1, we define a "distance" measure as the number of hops from one node to another along the ring, where clockwise hops are positive, and counter-clockwise hops are negative. Recall that the definition of "negative distance" is required only to keep the matrix symmetric. The "negative" matrix is just the transpose of the matrix defined for the "positive" direction. After each operation, the distance between nodes is still the number of hops along the ring, though the number of nodes doubles each time. We then define the following family of matrices, $\mathcal{H}$:

$$H_0 = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, H_1 = \begin{pmatrix} p & p \\ 1 & p \end{pmatrix}, H_i = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix} \; \forall i > 1$$

Note that $H_{-i} = H_i'$. So, starting from the initiator matrix in Figure 1(a), we have the following progression of matrices:

$$G_1 = \begin{pmatrix} 1 & 1 & p & 1 \\ 1 & 1 & 1 & p \\ p & 1 & 1 & 1 \\ 1 & p & 1 & 1 \end{pmatrix},$$

$$\begin{aligned} G_2 =& G_1 \otimes_d \mathcal{H} \\ =& \begin{pmatrix} 1 \times H_0 & 1 \times H_1 & p \times H_2 & 1 \times H_{-1} \\ 1 \times H_{-1} & 1 \times H_0 & 1 \times H_1 & p \times H_2 \\ p \times H_2 & 1 \times H_{-1} & 1 \times H_0 & 1 \times H_1 \\ 1 \times H_1 & p \times H_2 & 1 \times H_{-1} & 1 \times H_0 \end{pmatrix} \\ =& \begin{pmatrix} 1 & 1 & p & p & p & p & p & 1 \\ 1 & 1 & 1 & p & p & p & p & p \\ p & 1 & 1 & 1 & p & p & p & p \\ p & p & 1 & 1 & 1 & p & p & p \\ p & p & p & 1 & 1 & 1 & p & p \\ p & p & p & p & 1 & 1 & 1 & p \\ p & p & p & p & p & 1 & 1 & 1 \\ 1 & p & p & p & p & p & 1 & 1 \end{pmatrix} \end{aligned}$$

Note that the W-S model is not searchable by a greedy agent; however, if $P(u,v) = \frac{1}{d(u,v)}$, it becomes searchable [14], [5]. It is possible to model this $P(u,v)$ by simply adjusting $H_i, i \geq 1$ as follows:

$$H_0 = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, H_i = i \begin{pmatrix} \frac{1}{2i} & \frac{1}{2i+1} \\ \frac{1}{2i-1} & \frac{1}{2i} \end{pmatrix}, \forall i \geq 1, i \neq \frac{n}{2},$$

$$H_i = i \begin{pmatrix} \frac{1}{2i} & \frac{1}{2i-1} \\ \frac{1}{2i-1} & \frac{1}{2i} \end{pmatrix}, \forall i \geq 1, i = \frac{n}{2}$$

As in the previous examples, $H_{-i} = H_i'$. The different definition for the middle node in the ring is due to the fact that we need the probability of a connection to reach a minimum at this point, and then start to rise again. With this new definition of $H_i, i \geq 1$, we have the following progression of matrices:

$$G_1 = \begin{pmatrix} 1 & 1 & 1/2 & 1 \\ 1 & 1 & 1 & 1/2 \\ 1/2 & 1 & 1 & 1 \\ 1 & 1/2 & 1 & 1 \end{pmatrix},$$

$$\begin{aligned} G_2 =& G_1 \otimes_d \mathcal{H} \\ =& \begin{pmatrix} 1 \times H_0 & 1 \times H_1 & 1/2 \times H_2 & 1 \times H_{-1} \\ 1 \times H_{-1} & 1 \times H_0 & 1 \times H_1 & 1/2 \times H_2 \\ 1/2 \times H_2 & 1 \times H_{-1} & 1 \times H_0 & 1 \times H_1 \\ 1 \times H_1 & 1/2 \times H_2 & 1 \times H_{-1} & 1 \times H_0 \end{pmatrix} \\ =& \begin{pmatrix} 1 & 1 & 1/2 & 1/3 & 1/4 & 1/3 & 1/2 & 1 \\ 1 & 1 & 1 & 1/2 & 1/3 & 1/4 & 1/3 & 1/2 \\ 1/2 & 1 & 1 & 1 & 1/2 & 1/3 & 1/4 & 1/3 \\ 1/3 & 1/2 & 1 & 1 & 1 & 1/2 & 1/3 & 1/4 \\ 1/4 & 1/3 & 1/2 & 1 & 1 & 1 & 1/2 & 1/3 \\ 1/3 & 1/4 & 1/3 & 1/2 & 1 & 1 & 1 & 1/2 \\ 1/2 & 1/3 & 1/4 & 1/3 & 1/2 & 1 & 1 & 1 \\ 1 & 1/2 & 1/3 & 1/4 & 1/3 & 1/2 & 1 & 1 \end{pmatrix} \end{aligned}$$

This example already illustrates that the generalized operator we have defined allows the generation of searchable networks, but we will provide another more realistic example in the next example.

Example 3: Kleinberg-like Model. The final example we consider, Kleinberg's lattice model, is particularly pertinent as it was shown to be searchable [14]. In the original formulation, local connections of nodes are defined on a k-dimensional lattice, and long-range links occur between two nodes at distance d with probability proportional to $d^{-\alpha}$. We focus on a "Kleinberg-like" model here, where instead of a k-dimensional lattice, we have an "expanding hypercube" as our underlying lattice. In this example, at any point, the graph is a

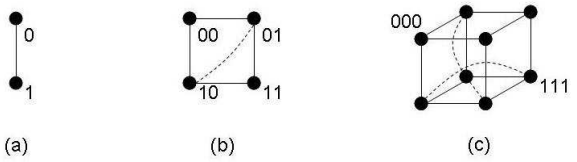

Fig. 2. Example: the growth of an expanding hypercube

hypercube with some extra long-range connections, and when it grows, it grows by doubling the number of nodes and adding a dimension to the hypercube. Note that we will have n nodes arranged on a $k = \log n$-dimensional hypercube. This example is of particular interest due to recent work suggesting that many networks

have an underlying hyperbolic or tree-metric structure [19], [18]. The expanding hypercube captures the key feature of these topologies, as the number of nodes at distance d grows exponentially in d. This example is also very naturally represented using our "distance"-dependent Kronecker operation and a Hamming distance as our "distance" measure.

To define the expanding hypercube, we define a graph G with n nodes, numbered $1...n$, where each node is labeled with its corresponding $\log n$-length bit vector. We define the "distance" between two nodes as the Hamming distance between their labels. The family of matrices $\mathcal{H}$ is as follows:

$$H_0 = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix}, H_i = \begin{pmatrix} 1 & \beta_i \\ \beta_i & 1 \end{pmatrix}, \text{for all } i \geq 1$$

where $\beta_1 =$ a normalizing constant, $\beta_i = \frac{P(i+1)}{P(i)}$. The graph may or may not be searchable depending on $P(i)$. To mimic Kleinberg's model, we let $P(i) = i^{-\alpha}$, so that $\beta_i = \left(\frac{i+1}{i}\right)^{-\alpha}$. Thus, for the sequence of graphs shown in the figure above, we have the following sequence of matrices:

$$G_1 = \begin{pmatrix} 1 & 1 \\ 1 & 1 \end{pmatrix},$$

$$G_2 = \begin{pmatrix} 1 & 1 & 1 & \beta_1 \\ 1 & 1 & \beta_1 & 1 \\ 1 & \beta_1 & 1 & 1 \\ \beta_1 & 1 & 1 & 1 \end{pmatrix},$$

$$G_3 = \begin{pmatrix} 1 & 1 & 1 & \beta_1 & 1 & \beta_1 & \beta_1 & \beta_1\beta_2 \\ 1 & 1 & \beta_1 & 1 & \beta_1 & 1 & \beta_1\beta_2 & \beta_1 \\ 1 & \beta_1 & 1 & 1 & \beta_1 & \beta_1\beta_2 & 1 & \beta_1 \\ \beta_1 & 1 & 1 & 1 & \beta_1\beta_2 & \beta_1 & \beta_1 & 1 \\ 1 & \beta_1 & \beta_1 & \beta_1\beta_2 & 1 & 1 & 1 & \beta_1 \\ \beta_1 & 1 & \beta_1\beta_2 & \beta_1 & 1 & 1 & \beta_1 & 1 \\ \beta_1 & \beta_1\beta_2 & 1 & \beta_1 & 1 & \beta_1 & 1 & 1 \\ \beta_1\beta_2 & \beta_1 & \beta_1 & 1 & \beta_1 & 1 & 1 & 1 \end{pmatrix}$$

From the matrix, we can tell that in each step,

$$P(u,v) = \begin{cases} 1 & \text{if } d(u,v) = 0, 1 \\ d(u,v)^{-\alpha} & \text{otherwise} \end{cases}$$

In the original k-dimensional lattice, a distributed algorithm (as defined in Section V), can find paths of length $O(\log n)$ only if $\alpha = k$ [14]; in the modified case presented above, we will see in section V that we need a different probability of connection to find short paths.

IV. Connection to hidden hyperbolic space model

As mentioned previously, the expanding hypercube model in Example 3 resembles models proposed in [17] and extended in [18], [21], and [23]. In [17], every node in the network has a hidden variable - their location in a hidden metric space. The probability of a connection between two nodes is based upon the distance between them in this hidden space. The resulting degree distribution depends on the curvature of this hidden space; if the space has negative curvature, the degree distribution will be scale-free with $P(k) = k^{-\gamma}$ [24].

In the distance-dependent Kronecker graph described in this paper and [2], the probability of a connection is based on the distance between two nodes in the given lattice, defined usually by H_0 and H_1 in the family of matrices $\mathcal{H}$. As a result, the lattice, or metric space, is not really hidden since neighbors are explicitly connected in the lattice. It is important to note that both models incorporate a distance-dependent probability of connection. As will be defined formally in Section V, a local greedy search algorithm can take advantage of this embedding into a hidden or physical space to forward a message to a destination. If a given node u has a message to forward to a destination t, it can use its knowledge of the embedding to forward the message to its neighbor closest to the destination in the embedding. It is not necessary that the embedding be physical, as shown in [18] and [21]; rather, what is necessary is that the the probability of a connection between two nodes is dependent on the distance between them. In most social networks the abstract distance is a measure of "social distance" - the likelihood of two individuals being connected depends on their memberships in various groups, among other factors.

In addition, in the models of [17], a hyperbolic space results in exponentially expanding neighborhoods around each node. In the distance-dependent hypercube example, there are $\binom{k}{d}$ nodes at each distance d, also resulting in exponentially expanding neighborhoods. However, the hidden metric space model necessarily includes the notion of a core and periphery of the network, where high-degree nodes form the core connecting many low-degree nodes at the periphery [21]. In the hypercube example, all nodes are homogeneous in expected degree - there is no notion of a core.

In [18], as nodes are located further from the origin in the hidden hyperbolic space their expected degree decreases exponentially ($\propto e^{-\beta r}$). When this is combined with the exponentially expanding neighborhoods ($\propto e^{\alpha r}$), the result is a scale-free distribution with $\gamma = 1 + \frac{\alpha}{\beta}$. It is important to note that an exponential decrease in expected degree is not strictly necessary; to see this, let the number of nodes at distance r from a reference origin in the hyperbolic space be

$$n(r) = e^{\alpha r}$$

Let the average degree of nodes at distance r be

$$k(r) = r^{-\delta}$$

so that

$$r(k) = k^{-\frac{1}{\delta}}$$

Using

$$n(k) \propto n[r(k)]\,|r'(k)|$$

we have

$$n(k) \propto e^{\alpha k^{-1/\delta}} k^{-1/\delta - 1}$$

which asymptotically behaves like a power law with $\gamma = 1 + 1/\delta$. In the hypercube example, despite the exponential expansion of neighborhoods, the resulting degree distribution will always be Poisson as long as the probability of connection is sufficiently small, as shown in the next section.

Nevertheless, the connection between this model and those based on tree-metrics and hidden metric spaces is important to note, as one key factor emerges: a distance-dependent relation is necessary for a greedy algorithm to succeed in finding shortest paths.

V. Degree distribution

In this section we describe a general characteristic function-based analysis of degree distributions for lattice-based networks, and apply it to the expanding hypercube example in Section III. In general, any lattice-based network with a distance-dependent probability of connection will have a Poisson degree distribution, as long as the probability of a connection at a distance d is sufficiently small. Formally,

Theorem 5.1: The degree distribution of a general lattice-based network with a distance-dependent probability of connection $P(d)$ and maximum distance d_{max} will have the following degree distribution:

$$P(\nu = i) = \frac{e^{-\alpha}\alpha^i}{i!}\left(1 + d_{max}O(P^2(d))\right)$$

where

$$\alpha = \sum_{d=1}^{d_{max}} P(d)\sigma(d) \tag{1}$$

and $\sigma(d)$ = number of nodes at distance d from a reference node in the lattice. We note that if $\lim_{n\to\infty} d_{max}P^2(d) = 0$, then the degree distribution is Poisson.

Proof: Let ν denote the degree of an arbitrary node u in a general lattice-based network with n nodes. Thus, $\nu = v_1 + v_2 + \cdots + v_n$ where

$$v_i = \begin{cases} 1 & \text{if link to node } i, \\ 0 & \text{otherwise.} \end{cases}$$

We define the characteristic function of the degree distribution as

$$\begin{aligned} E[e^{it\nu}] =& E[e^{it(v_1+v_2+\cdots+v_n)}] \\ =& E[e^{itv_1}]E[e^{itv_2}]\ldots E[e^{itv_n}] \end{aligned}$$

We can then group the expectations

$$\begin{aligned} E[e^{it\nu}] &= \prod_{d=1}^{d_{max}} (1 - P(d) + P(d)e^{it})^{\sigma(d)} \\ &= \prod_{d=1}^{d_{max}} (1 - P(d)(1 - e^{it}))^{\sigma(d)} \\ &= \prod_{d=1}^{d_{max}} \left(e^{-P(d)(1-e^{it})} + O(P^2(d))(1-e^{it})^2\right)^{\sigma(d)} \\ &\quad \text{as } e^{-x} = 1 - x + O(x^2) \end{aligned} \tag{2}$$

Thus, we can pull out the first term and using binomial approximation of $(1+x)^c = 1 + cx + O(x^2)$, we have

$$\begin{aligned} E[e^{it\nu}] &= \prod_{d=1}^{d_{max}} e^{-P(d)(1-e^{it})\sigma(d)} \left(1 + \frac{O(P^2(d))(1-e^{it})^2\sigma(d)}{e^{-P(d)(1-e^{it})}}\right) \\ &= e^{-(1-e^{it})\sum_{d=1}^{d_{max}} P(d)\sigma(d)} \times \\ &\quad \prod_{d=1}^{d_{max}} \left(1 + O(P^2(d))(1-e^{it})^2\sigma(d)e^{P(d)(1-e^{it})}\right) \\ &\approx e^{\alpha(e^{it}-1)}(1 + d_{max}O(P^2(d))) \end{aligned}$$

Expanding, we see that the characteristic function is

$$E[e^{it\nu}] = \left(1 + d_{max}O(P^2(d))\right)e^{-\alpha}\left(1 + \alpha e^{it} + \frac{(\alpha e^{it})^2}{2!} + \ldots\right)$$

From such a representation of the characteristic function, we can clearly see the degree distribution as

$$P(\nu = i) = \frac{e^{-\alpha}\alpha^i}{i!}\left(1 + d_{max}O(P^2(d))\right)$$

■

We now turn to a specific lattice-based network, the hypercube distance-dependent Kronecker graph described in Example 3 in Section III. In this example, $\sigma(d) = \binom{k}{d}$, and the maximum distance in the network is $k = \log n$. We use a particular $P(d) = \left[\binom{k-\frac{2d}{3}}{\frac{d}{3}} d \log k \ln 3\right]^{-1}$ optimized for searchability, as determined in Section VI.

Theorem 5.2: The degree distribution of the expanding hypercube is given by the following Poisson distribution,

$$P(\nu = i) = \frac{e^{-\alpha}\alpha^i}{i!} \text{ where } \alpha \approx \frac{3.6919\, n^{.4703}}{\log\log n\sqrt{\log n}} \tag{3}$$

Proof: We use the same framework as in the proof of Theorem 5.1, and let $e^{it} = x$ for simplicity. In this case, the characteristic function becomes

$$E[x^\nu] = e^{-(1-x)\sum_{d=1}^{k} P(d)\sigma(d)}$$

so that

$$\begin{aligned}\alpha &= \sum_{d=1}^{k} P(d)\sigma(d) \\ &= \sum_{d=1}^{k} \left[\binom{k - \frac{2d}{3}}{\frac{d}{3}} d \log k \ln 3 \right]^{-1} \binom{k}{d}\end{aligned}$$

To calculate α, we use the entropy approximation $\binom{k}{d} \approx 2^{kH(\frac{d}{k})}$, which holds as $\binom{n}{k} = 2^{n(H(p)+o(1))}$ when $k \propto pn$, so that

$$\alpha \approx \frac{1}{\log k \ln 3} \sum_{d=1}^{k} d^{-1} 2^{kH\left(\frac{d}{k}\right) - \left(k - \frac{2d}{3}\right) H\left(\frac{\frac{d}{3}}{k - \frac{2d}{3}}\right)}$$

We can approximate the sum by using saddle point integration.

$$\begin{aligned}&\int g(y) e^{kf(y)}\, dy \\ &= \sqrt{\frac{2\pi}{k\left|f''(y_0)\right|}} g(y_0) e^{kf(y_0)} \left(1 + O\left(\frac{1}{\sqrt{k}}\right)\right) \qquad (4)\end{aligned}$$

where y_0 is the saddle point of the function $f(y)$, i.e., the point at which $f'(y) = 0$.

We rewrite the sum $S(k)$ in nats, leaving out the constants in front,

$$S(k) = \frac{1}{k} \sum_{d=1}^{k} \frac{k}{d} e^{k\left[H\left(\frac{d}{k}\right) - \left(1 - \frac{2d}{3k}\right) H\left(\frac{\frac{d}{3k}}{1 - \frac{2d}{3k}}\right)\right]}$$

and then we let $y = \frac{d}{k}$,

$$S(y) = \int_{\frac{1}{k}}^{1} \frac{1}{y} e^{k\left[H(y) - \left(1 - \frac{2}{3y}\right) H\left(\frac{\frac{y}{3}}{1 - \frac{2}{3y}}\right)\right]} dy$$

so that, with the saddle point approximation of line (4), $g(y) = \frac{1}{y}$ and $f(y) = H(y) - (1 - \frac{2}{3y}) H(\frac{\frac{y}{3}}{1 - \frac{2}{3}y})$. Using Mathematica, we find

$$\begin{aligned} y_0 &= 0.417 \\ f(y_0) &= 0.326 \\ g(y_0) &= 2.4 \\ |f''(y_0)| &= 2.2 \end{aligned}$$

yielding,

$$S(k) \approx \sqrt{\frac{2\pi}{2.2k}} (2.4) e^{0.326k} \qquad (5)$$

So, our α is now

$$\begin{aligned}\alpha &\approx \frac{1}{\log k \ln 3} \sqrt{\frac{2\pi}{2.2k}} (2.4) e^{0.326k} \\ &\approx \frac{3.6919\, n^{0.4703}}{\log \log n \sqrt{\log n}}\end{aligned}$$

With the results of Theorem 5.1, we have a Poisson degree distribution with parameter α. ■

A. Expected Degree

From the characteristic function, we can also determine the expected degree.

$$\begin{aligned} E[\nu] &= \left. \frac{\partial}{\partial x} E[x^{\nu}] \right|_{x=1} \\ &= \left. \frac{\partial}{\partial x} [e^{-(1-x)\alpha}] \right|_{x=1} \\ &= \alpha \end{aligned}$$

Thus, the expected degree of the expanding hypercube example is a growing function of n.

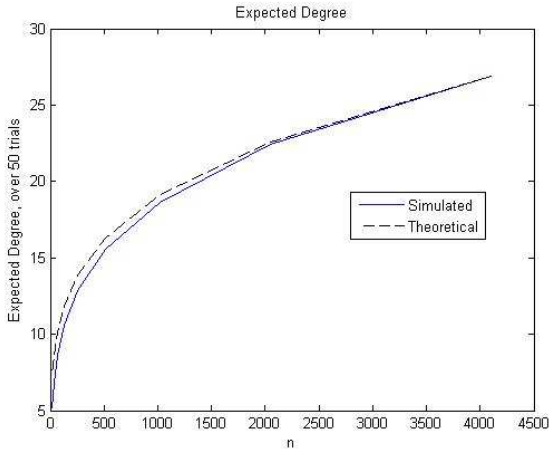

Fig. 3. Expected degree of expanding hypercube

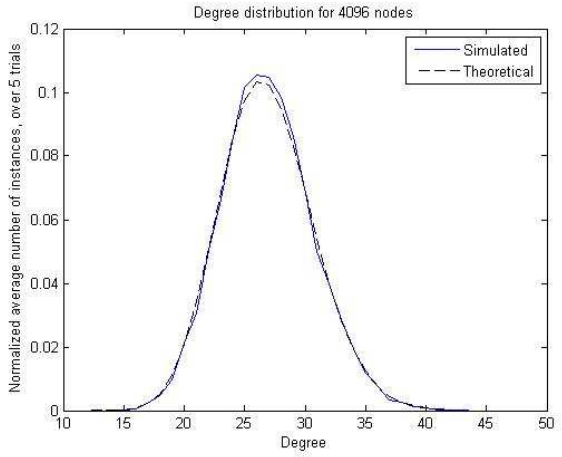

Fig. 4. Example histogram with $n = 4096$

B. Simulation of expanding hypercube example

Simulating the expanding hypercube with the $P(d)$ determined in Section VI yields results that match well, within a constant, the analysis above. Figure 3 shows

the comparison of the theoretical and simulated expected degrees, while figure 4 shows an example histogram of the degree distribution, both theoretical and simulated, with $n = 4096$. The Poisson nature of the distribution is clearly visible, as is the growth of the expected degree as a function of n.

VI. Proving searchability

While the distance-dependent Kronecker graph model is more complicated than the original Kronecker graph model, it can capture several existing network models, and it incorporates "distance" into the probability of connection, allowing for several cases in which searchability can be proven. In this section, we first give a general framework within which a lattice-based network can be proven searchable and then proceed to the specific cases of the Kleinberg model [14] and the expanding hypercube model of Example 3 in Section III.

A. General Searchability Theorem

We define a decentralized algorithm $\mathcal{A}$ similar to [14]. In each step, the current message-holder u passes the message to a neighbor that is closest to the destination, t. Each node only has knowledge of its address on the lattice (given by its bit vector label in the case of the expanding hypercube), the address of the destination, and the nodes that have previously come into contact with the message. For the graph to be searchable, we need to have that the distributed algorithm $\mathcal{A}$ is able to find short paths through the network, which are usually $O(D)$ where D is the diameter of the network.

Let the current message holder be node u and the destination node t. We will say that the execution of a decentralized search algorithm $\mathcal{A}$ is in phase j when $2^j < d(u,t) \leq 2^{j+1}$, where $d(u,t)$ is the distance between node u and node t. Thus, the largest value of j in a general lattice-based network is $j_{max} = \log d_{max}$ where d_{max} denotes the maximum geodesic in the network. For example, in a hypercube, the maximum geodesic is $d_{max} = \log n = k$, so $j_{max} = \log\log n = \log k$. We define $N_{u,t}(d) = \{v : d(v,t) \leq 2^j, d(u,v) = d\}$ and $\min|N(d)| = \min_{u,t,d(u,t)=d}|N_{u,t}(d)|$.

Theorem 6.1: A decentralized algorithm $\mathcal{A}$ will find short paths of length $O(\log^2(d_{max}))$, when the probability of a connection is

$$P(u,v) = [c\, d \min|N(d)|]^{-1} \tag{6}$$

where $c \propto \log d_{max}$.

Proof: Suppose we are in phase j with current message holder node u; we want to determine the probability that the phase ends at this step. This is equivalent to the probability that the message enters a set of nodes B_j where $B_j = \{v : d(v,t) \leq 2^j\}$.

$$\begin{aligned}\Pr(\{\text{message enters } B_j\}) =& 1 - \prod_{v\in B_j}(1 - P(u,v : v \in B_j)) \\ =& 1 - \prod_{d=d(u,t)-2^j}^{d(u,t)+2^j}(1-P(d))^{|N_{u,t}(d)|} \\ \geq& 1 - \prod_{d=d(u,t)-2^j}^{d(u,t)+2^j}(1-P(d))^{\min|N(d)|}\end{aligned}$$

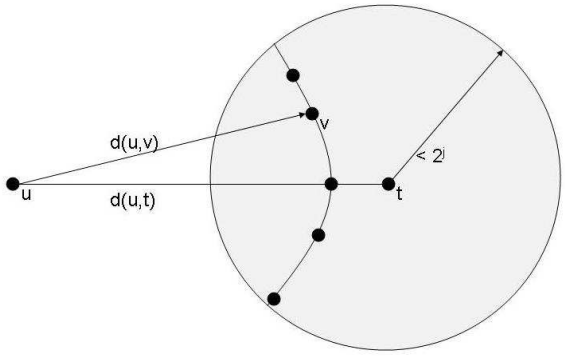

Fig. 5. Relative positions of nodes u,v, and t in phase j

In any network model, enforcing searchability boils down to determining this $\min|N(d)|$, the minimum number of nodes at a distance d from a given node u within a ball of nodes centered around the destination, t, as illustrated in Figure 5. Once this $\min|N(d)|$ is found, if we set the probability of a connection between two nodes distance d apart as in Theorem 6.1, with an appropriate constant, we will find that each phase described above will end in approximately j_{max} steps, and, as there are only j_{max} such phases, our greedy forwarding algorithm will be able to find very short paths of length $O(j_{max}^2)$.

Thus, we have

$$\Pr(\{\text{message enters } B_j\}) \geq 1 - \prod_{d=d(u,t)-2^j}^{d(u,t)+2^j}(1-P(d))^{\min|N(d)|}$$

$$\begin{aligned}\approx& 1 - e^{-\sum_{d=d(u,t)-2^j}^{d(u,t)+2^j}\min|N(d)|P(d)} \qquad (7)\\ =& 1 - e^{-\frac{1}{c}\sum_{d=d(u,t)-2^j}^{d(u,t)+2^j} d^{-1}} \\ \geq& 1 - e^{-\frac{1}{c}\ln\frac{d(u,t)+2^j}{d(u,t)-2^j}} \\ \geq& 1 - e^{-\frac{1}{c}\ln\frac{3\,2^j}{2^j}} \\ =& 1 - e^{-\frac{1}{c'}} \\ \geq& \frac{1}{c'} \qquad (8)\end{aligned}$$

where the approximation in line (7) requires that $\lim_{n\to\infty} d_{max}P^2(d) = 0$, which holds with the $P(d)$ as specified in line (6) (see proof of Theorem 5.1 for extra order terms), and line (8) comes from the power series expansion of e^{-x}. Let X_j denote the total number of steps spent in phase j. Then,

$$\begin{aligned} EX_j =& \sum_{i=1}^{\infty} \Pr[X_j \geq i] \\ \leq& \sum_{i=1}^{\infty} \left(1 - \frac{1}{c'}\right)^{i-1} \\ =& c' \end{aligned}$$

Let X denote the total number of steps taken by the algorithm $\mathcal{A}$.

$$X = \sum_{j=0}^{j_{max}} X_j$$

and

$$\begin{aligned} EX =& \sum_{j=0}^{j_{max}} EX_j \\ \leq& (1 + j_{max})(c') \\ =& (1 + \log d_{max}) \log d_{max} \\ \leq& \delta(\log d_{max})^2 \end{aligned} \tag{9}$$

where line (9) holds $\forall\, \delta \geq 2$, $\log d_{max} \geq 2$. ■

With this framework, we can explore the searchability of any lattice-based network model with distance-dependent connection probability.

B. Searchability in original Kleinberg model

In the original Kleinberg two-dimensional lattice [14], the number of nodes at a distance d from a reference node is approximately $4d$, ignoring edge effects. The maximum distance between any two nodes is $O(n)$, so $j_{max} \approx \log n$. Additionally, the diameter of the graph is on the order of $\log n$. In general, $\min|N(d)| \propto d$ for a fixed j, resulting in the probability of connection optimized for searchability, $P(d) = [\alpha \log(n) d^2]^{-1}$. Using this $P(d)$,

$$\begin{aligned} \Pr(\{\text{message enters } B_j\}) \geq& 1 - \\ & \prod_{d=d(u,t)-2^j}^{d(u,t)+2^j} (1 - P(d))^{\min|N(d)|} \\ \approx& 1 - e^{-\frac{1}{\alpha \log n} \sum_{d=d(u,t)-2^j}^{d(u,t)+2^j} d^{-1}} \qquad (10) \\ \geq& 1 - e^{-\frac{1}{\alpha' \log n}} \\ \geq& \frac{1}{\alpha' \log n} \qquad (11) \end{aligned}$$

where line (10) holds for the $P(d)$ specified, and line (11) comes from the power series expansion of e^{-x}. Therefore,

$$EX_j \leq \alpha' \log n$$

and

$$\begin{aligned} EX =& \sum_{j=0}^{\log n} EX_j \\ \leq& \delta(\log n)^2. \end{aligned} \tag{12}$$

where line (12) holds $\forall\, \delta \geq 2$, $\log n \geq 2$.

C. Searchability in expanding hypercube example

In the expanding hypercube example of Section III, each node has $\log n$ neighbors from the lattice itself. With the addition of long-range links, we expect the diameter to be $O(\log \log n)$, similar to [18]. Note that with this example, $j_{max} = \log \log n = \log k$ and the number of nodes at distance d equals $\binom{n}{d}$. Using Theorem 6.1, we can prove the following result:

Theorem 6.2: A decentralized algorithm $\mathcal{A}$ will find paths of length $O((\log \log n)^2)$ in the expanding hypercube example when

$$\begin{aligned} &\beta_0 = 1,\ \beta_1 = [2 \log k \ln 3]^{-1}, \\ &\beta_i = \left[\binom{k - \frac{2i}{3}}{\frac{i}{3}} i\right] \left[\binom{k - \frac{2(i+1)}{3}}{\frac{i+1}{3}} (i+1)\right]^{-1} \quad \forall i \geq 2 \end{aligned} \tag{13}$$

such that the probability of a connection is

$$P(u,v) = \begin{cases} 1 & \text{if } d(u,v) = 0, 1 \\ \left[\binom{k - \frac{2d}{3}}{\frac{d}{3}} d \log k \ln 3\right]^{-1} & \text{if } d(u,v) = d \end{cases} \tag{14}$$

Proof: Using Theorem 6.1, all that remains is to find $\min|N(d)|$ and to determine the appropriate constants to use. Without loss of generality, we assume that the destination node t is the all-zero node (i.e., its label is the zero vector) so that we can write $d(u,t) = \|u\|$. To determine $\min|N(d)|$ in our case, since the distance measure is a Hamming distance, we must count the number of possible bit vectors that are at a specific distance d from a node u while still being within a certain distance of the destination. We prove that $\min|N(d)| = \binom{k - \frac{2d}{3}}{\frac{d}{3}}$ in Appendix A. We then let $c = \log k \ln 3$ for reasons that will be clear below. Using the same framework as

in Theorem 6.1 we have that

$$\begin{aligned}\Pr(\{\text{msg enters } B_j\}) \geq & 1 - \prod_{d=\|u\|-2^j}^{\|u\|+2^j} (1-P(d))^{\min|N(d)|} \\ \approx & 1 - e^{-\frac{1}{\log k \ln 3}\sum_{d=\|u\|-2^j}^{\|u\|+2^j} d^{-1}} \qquad (15) \\ \geq & 1 - e^{-\frac{1}{\log k}} \\ \geq & \frac{1}{\log k} \qquad (16)\end{aligned}$$

where line (15) holds for the $P(d)$ specified, and line (16) comes from the power series expansion of e^{-x}. Therefore, we have

$$EX_j \leq \log k$$

and

$$\begin{aligned}EX = & \sum_{j=0}^{\log k} EX_j \\ \leq & \delta(\log k)^2,\ \forall\ \delta \geq 2,\ \log k \geq 2\end{aligned}$$

Since the expected number of steps in phase j is $\log k$, and there are at most $\log k$ phases, the expected amount of steps taken by the algorithm $\mathcal{A}$ is at most $\delta \log^2 k$. So, with this definition of $P(d)$, the distributed algorithm provides searchabilty. ■

D. Simulation of distributed search algorithm

We simulated the local greedy algorithm described above in MATLAB for $16 \leq n \leq 4096$ with the probability distribution as in Theorem 5.2 and appropriate floor functions. We found that the greedy algorithm finds a path between two nodes with an average length of a constant factor away from the diameter of the simulated network, where diameter is defined as the maximum geodesic in the network. Note that the two nodes selected for the simulation are actually the "worst-case" nodes - the distance between them in the network is exactly the diameter. Figure 6 illustrates the results of the greedy algorithm simulations.

E. Path length with sub-optimal P(d)

In this section we analyze the performance of the local greedy search algorithm on the expanding hypercube when $P(d)$ is not optimal, as in Theorem 6.2. For this example, let $P(d) = [\log k \binom{k}{d}]^{-1}$, which is clearly not $\min|N(d)|$ from Lemma 9.1. We will show that this suboptimal $P(d)$ also allows for searchability.

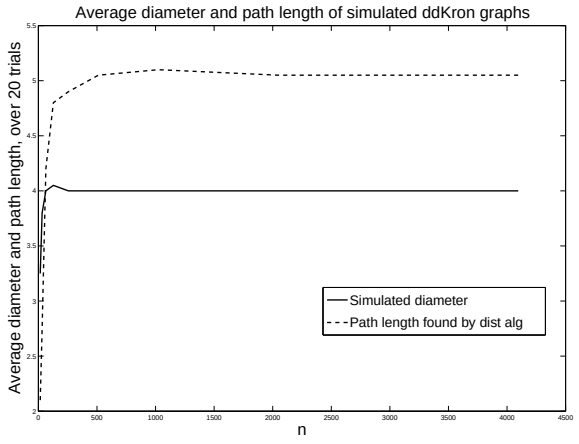

Fig. 6. Average path length found by greedy algorithm using local information

Using the same framework as in Theorem 6.1,

$$\begin{aligned}\Pr(\{\text{msg enters } B_j\}) \geq & 1 - \prod_{d=d(u,t)-2^j}^{d(u,t)+2^j} (1-P(d))^{\min|N(d)|} \\ \approx & 1 - e^{\sum_{d=d(u,t)-2^j}^{d(u,t)+2^j} P(d)\min|N(d)|} \qquad (17) \\ = & 1 - e^{-\sum_{d=d(u,t)-2^j}^{d(u,t)+2^j} P(d)\binom{k-\frac{2d}{3}}{\frac{d}{3}}} \\ = & 1 - e^{-\frac{1}{\log k}S(k,d)} \\ \geq & 1 - e^{-\frac{1}{\log k}\min S(k,d)}\end{aligned}$$

where line (17) holds for the specified $P(d)$ and where

$$\begin{aligned}S(k,d) = & \sum_{d=2^j}^{3*2^j} \binom{k}{d}^{-1}\binom{k-\frac{2d}{3}}{\frac{d}{3}} \\ \approx & \sum_{d=2^j}^{3*2^j} 2^{(k-\frac{2d}{3})H(\frac{\frac{d}{3}}{k-\frac{2d}{3}})-kH(\frac{d}{k})} \qquad (18) \\ \geq & \min_d \sum_{d=2^j}^{3*2^j} 2^{(k-\frac{2d}{3})H(\frac{\frac{d}{3}}{k-\frac{2d}{3}})-kH(\frac{d}{k})} \\ \geq & 2^{\max_d (k-\frac{2d}{3})H(\frac{\frac{d}{3}}{k-\frac{2d}{3}})-kH(\frac{d}{k})}\end{aligned}$$

where we have used the approximation $\binom{k}{d} \approx 2^{kH(\frac{d}{k})}$, which holds as $\binom{n}{k} = 2^{n(H(p)+o(1))}$ when $k \propto pn$, in line (18). Since the exponent is convex in d, the maximum will be at either the upper or lower bound of the sum. For $0 \leq j \leq \log k$ the lower bound ($d = 2^j$) yields the maximal exponent. So, we have

$$\begin{aligned}\Pr(\{\text{msg enters } B_j\}) \geq & 1 - e^{-\frac{1}{\log k}2^{f(k,j)}} \\ \geq & \frac{2^{f(k,j)}}{\log k}\end{aligned}$$

where we have used the power series expansion of e^{-x} and where

$$f(k,j) = (k - \frac{2^{j+1}}{3})H(\frac{\frac{2^j}{3}}{k - \frac{2^{j+1}}{3}}) - kH(\frac{2^j}{k}). \quad (19)$$

Continuing with the proof of searchability, we have

$$\begin{aligned} EX_j &= \sum_{i=1}^{\infty} \Pr[X_j \geq i] \\ &\leq \log k \; 2^{-f(k,j)} \end{aligned}$$

and

$$\begin{aligned} EX &= \sum_{j=0}^{\log k} EX_j \\ &\leq (1 + \log k) \log k \; 2^{-\min_j f(k,j)} \\ &\leq \delta (\log k)^2, \; \forall \, \delta \geq 2, \; \log k \geq 2 \end{aligned}$$

since $f(k,j)$ is convex but its minimum occurs close to $\log k$. As a result, even for suboptimal $P(d)$, a local greedy algorithm can find short paths. However, the bounds used in the analysis above are looser than those in previous sections, so the final expected number of steps taken by $\mathcal{A}$ is not as tight. This analysis is supported by simulation results as shown in the figure below.

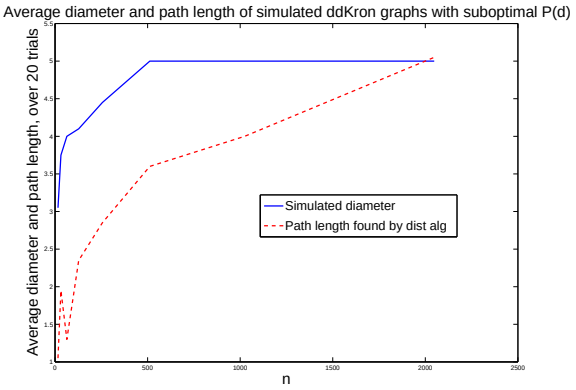

Fig. 7. Performance of greedy algorithm when $P(d) = [\log k \binom{k}{d}]^{-1}$

Finally, if $P(d) = [d \log k \binom{k}{d}]^{-1}$, using the same sort of techniques as above we can show that $EX \leq \delta k (\log k)^2$ for a large enough δ. Note that in this case, the paths found will be $O(\log n \log \log n)$, which are longer than before. Simulation results with this $P(d)$ are shown in figure 8.

VII. Brief diameter analysis of hypercube

In this section, we briefly discuss the diameter of a general random graph. Finding the actual diameter, defined as either the maximum or the average geodesic in the network, can be very complicated. We discuss

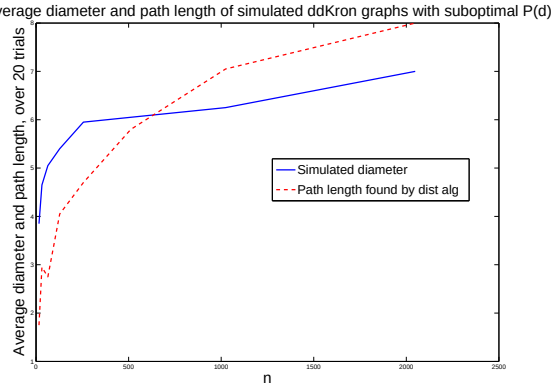

Fig. 8. Performance of greedy algorithm when $P(d) = [d \log k \binom{k}{d}]^{-1}$

a simple lower bound of the hypercube example here, which can be applied to any random graph.

If we assume that the expected degree of the hypercube example in Section III is polynomial in n, say n^β, similar to what was found in Section IV for the expanding hypercube, we can lower bound the diameter as follows. We assume that at each step, every node has d neighbors and that it takes α steps to reach all n nodes. Therefore, to reach all n nodes in the network, we have

$$\begin{aligned} & d^\alpha = n \\ & \Rightarrow (n^\beta)^\alpha = n \\ & \Rightarrow \alpha = \frac{1}{\beta} \\ & \Rightarrow \text{Constant diameter} \end{aligned}$$

Thus, a simple lower bound for the diameter of a graph with polynomial expected degree is some constant, $\frac{1}{\beta}$. We can also work backwards, assuming a $\log \log n$ diameter. In this case, we have

$$\begin{aligned} & d^\alpha = n \\ & \Rightarrow d^{\log \log n} = n \\ & \Rightarrow d = n^{\frac{1}{\log \log n}} = e^{\frac{\log n}{\log \log n}} \end{aligned}$$

which is less than a polynomial in n, but still grows with n. Figure 9 compares the simulated diameter of the expanding hypercube example with the two lower bounds discussed above. For $16 \leq n \leq 4096$, both lower bounds appear to be a good match.

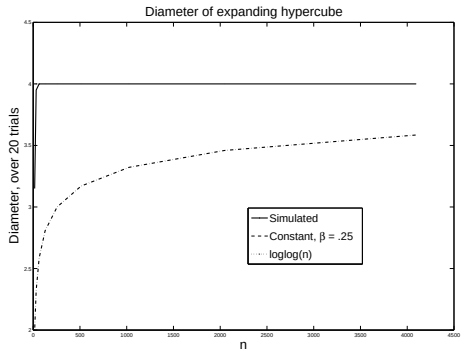

Fig. 9. Simulated and theoretical diameter of expanding hypercube

VIII. Conclusion

We have presented a generalization of Kronecker graphs by defining a family of "distance"-dependent matrices and a new Kronecker-like operation. As a result, the network model defines both local regular structures and global distance-dependent connections. Though the model is more complicated than the original Kronecker graph model, it is more general, as it can generate existing social network models, and more importantly, networks that are searchable. These properties emerge naturally from the definition of the embedding of the nodes and the probability of connection within the family of matrices $\mathcal{H}$. Any lattice-based network model with distance-dependent connection probabilities can be analyzed using the framework described in Sections V, VI, and VII for exploring degree distribution, diameter, and searchability. Most importantly, the searchability analysis shows how to make any network model searchable by defining the appropriate probability of connection based upon $|N(d)|$. The particular expanding hypercube example explicitly described here shares characteristics with those based upon hidden hyperbolic spaces [17], [18], though it has one major difference - degree homogeneity across nodes. Nevertheless, its exponentially expanding neighborhoods and distance-dependent probability of connection make it a good model for social networks as people tend to exhibit strong homophily, i.e., associating with other people most like themselves. In addition, in contrast to Kleinberg's lattice-based model [14], the searchability of the expanding hypercube is not too sensitive to the choice of $P(d)$.

Though this paper gives a near complete description of the characteristics of "distance"-dependent Kronecker graphs, there are many interesting questions that remain. These include how to parameterize the model from real-world data sets, and how to incorporate network dynamics. Ideally, given any data set, we would like to be able to find an appropriate family of distance-dependent matrices to match any desired characteristic of the data set. Additionally, while the current model incorporates some measure of growth, growing from a small initiator matrix to a final nxn adjacency matrix, we would like to better incorporate mobility into the model so that it is not just a static description of the network at one point in time.

IX. Appendix A - Calculating the size of $N_{u,t}(d)$

In this appendix, we show a lower bound for $|N_{u,t}(d)|$, the number of nodes at distance d from a given node u, still within distance 2^j of the destination, t.

Lemma 9.1: $\min |N_{u,t}(d)| = \binom{k-\frac{2d}{3}}{\frac{d}{3}}$

Proof: We first count exactly the number of nodes in $N_{u,t}(d)$, the number of nodes at a distance d from a given node u within a ball of nodes centered around the destination, t, as illustrated in Figure 5. Without loss of generality, define t as the all zero node, $t = (00...0)$. Arrange the label of u such that $u = (1...10...0)$. Define $v = (v_{11}\, v_{10}\, v_{01}\, v_{00})$ according to this partition of u, so that v_{11} and v_{01} have "1" entries and v_{10} and v_{00} have "0" entries. Let $\|x\|$ denote the weight, or number of ones, of the label of node x. We know the following:

$$\begin{aligned} v_{11} + v_{10} + v_{01} + v_{00} &= k \\ v_{11} + v_{10} &= \|u\| \\ v_{01} + v_{10} &= d \\ v_{11} + v_{01} &= \|v\| \end{aligned}$$

We can solve in terms of v_{11}, yielding

$$\begin{aligned} v_{00} &= k - d - v_{11} \\ v_{10} &= \|u\| - v_{11} \\ v_{01} &= d - \|u\| + v_{11} \end{aligned}$$

We also know that we must satisfy the following:

$$\begin{aligned} v_{11}, v_{10}, v_{01}, v_{00} &\geq 0 \\ 2^j < \|u\| &\leq 2^{j+1} \\ \|u\| - 2^j \leq d &\leq \|u\| + 2^j \\ \|v\| &\leq 2^j \end{aligned}$$

From these bounds we have

$$\max(0, \|u\| - d) \leq v_{11} \leq \min(\|u\|, k - d, \frac{1}{2}(2^j + \|u\| - d))$$

Note that the second and third bounds do not affect v_{11}. Counting the number of nodes in the ball, we have

$$|N_{u,t}(d)| = \sum_{v_{11}=v_l}^{v_u} \binom{\|u\|}{v_{11}} \binom{k - \|u\|}{d - \|u\| + v_{11}}$$

where we have substituted v_u and v_l, for the upper and lower bounds above, respectively. We can now approximate the number of nodes in $N_{u,t}(d)$, using the

entropy approximation for combinations. Let $\|u\| = ak, d = bk, 2^j = ck, x = v_{11}$. Using this notation, we have

$$\begin{aligned}|N_{u,t}(d)| &= \sum_{x=v_l}^{v_u} \binom{ak}{x}\binom{k(1-a)}{k(1-b)+x} \\ &\approx \sum_{x=v_l}^{v_u} 2^{k(aH\left(\frac{x}{ak}\right)+(1-a)H\left(\frac{b-a+\frac{x}{k}}{1-a}\right))} \qquad (20)\\ &\geq 2^{k\mathcal{X}}\end{aligned}$$

where

$$\mathcal{X} = \max_x aH\left(\frac{x}{ak}\right) + (1-a)H\left(\frac{b-a+\frac{x}{k}}{1-a}\right) \qquad (21)$$

subject to

$$k\max(0, a-b) \leq x \leq k\min(a, 1-b, \frac{1}{2}(a-b+c))$$

Note that line (20) is true as $\binom{n}{k} = 2^{n(H(p)+o(1))}$ when $k \propto pn$.

Note that the function $\mathcal{X}$ is concave in x, so unconstrained optimization yields the two solutions below, each giving different values of $\min|N_{u,t}(d)|$:

$$x_1^* = ak - abk \text{ when } c \geq a + b(1-2a)$$

yielding

$$\min|N_{u,t}(d)| = \binom{k}{d}$$

$$x_2^* = \frac{1}{2}k(a-b+c) \text{ when } c < a + b(1-2a)$$

yielding

$$\min|N_{u,t}(d)| = \binom{k-\frac{2d}{3}}{\frac{d}{3}}$$

The resulting $\min|N_{u,t}(d)|$ are derived in Sections A and B below. As the second solution yields a smaller $\min|N_{u,t}(d)|$, we have an overall $\min|N_{u,t}(d)| = \binom{k-\frac{2d}{3}}{\frac{d}{3}}$.

A. Solution 1: $c \geq a + b(1-2a)$

In this region, the solution to the unconstrained problem, $x_1^* = ak - abk$ gives us the maximal $\mathcal{X}$. Substituting in for the size of $N_{u,t}(d)$ and using the same entropy approximation as before, we have

$$\begin{aligned}|N_{u,t}(d)| &= 2^{k(aH\left(\frac{ak-abk}{ak}\right)+(1-a)H\left(\frac{b-a+\frac{ak-abk}{k}}{1-a}\right))} \\ &= 2^{k(aH(1-b)+(1-a)H(b))} \\ &= 2^{kH(b)} \\ &\approx \binom{k}{bk} \\ &= \binom{k}{d}.\end{aligned}$$

B. Solution 2: $c < a + b(1-2a)$

In this region, we choose one of the boundary points, $x_2^* = \frac{1}{2}k(a-b+c)$, as the solution to the maximization problem. Substituting this solution for x in $|N_{u,t}(d)|$, we obtain

$$|N_{u,t}(d)| = 2^{k(aH\left(\frac{a-b+c}{2a}\right)+(1-a)H\left(\frac{-a+b+c}{2(1-a)}\right))}$$

This gives us a function of a, b, c, so we want to find the worst case a, c that minimizes $|N_{u,t}(d)|$. The new optimization problem is thus

$$\begin{aligned}f(b) &= \min|N_{u,t}(d)| \\ &= \min_{a,c} aH\left(\frac{a-b+c}{2a}\right) + (1-a)H\left(\frac{-a+b+c}{2(1-a)}\right) \qquad (22)\end{aligned}$$

Note that the bounds for this region are:

1) $a - b - c \leq 0$
2) $a - b + c \geq 0$
3) $c < a \leq 2c$
4) $0 \leq c \leq \frac{1}{2}$
5) $0 \leq a, b \leq 1$
6) $0 \leq 2 - a - b - c$
7) $0 \leq a + b - c$
8) $0 \leq a + b - c - 2ab$

where 1) and 2) come from the bounds on $d(u, v)$, 3) comes from the bounds on $\|u\|$, and 4) and 5) come from the ranges for j and the size of the network. Note that 1-5 are always true, not just in this region. 6),7), and 8) come from the fact that our solution x_2^* is minimal in this region. Note that 8) implies 7).

Computing the Hessian of the function in line (22) shows that it is concave in both a and b; the derivation is in Appendix B. Since our function is concave, the $\min|N_{u,t}(d)|$ is found from the boundary points of Region 2. Rearranging the bounds from before in terms of a we have:

1) $a \leq b + c$
2) $a \geq b - c$
3) $a > c, a \leq 2c$
4) $c > 0, c \leq \frac{1}{2}$
5) $0 \leq a, a \leq 1$
6) $a \leq 2 - b - c$
7) $a \geq -b + c$
8) $a \geq \frac{c}{1-2b} - \frac{b}{1-2b}$ when $b \leq \frac{1}{2}$
9) $a \leq \frac{c}{1-2b} - \frac{b}{1-2b}$ when $b > \frac{1}{2}$

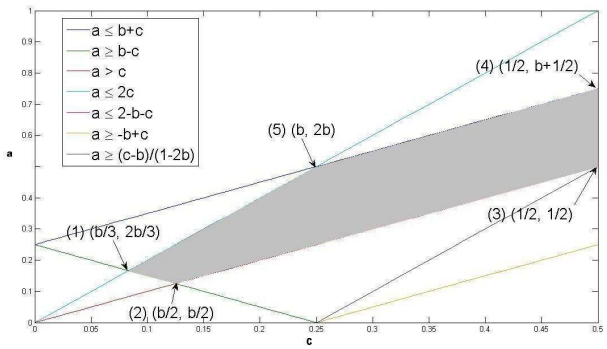

Fig. 10. Boundaries of $f(b)$ when $b \leq \frac{1}{2}$

When $b \leq \frac{1}{2}$, only bounds (1,2,3,4) apply to $f(b)$, yielding 5 points that we need to examine, as shown in Figure 10. If $b \geq .115$, then $f(b)$ is minimal at point (1), $(\frac{b}{3}, \frac{2b}{3})$, yielding

$$\begin{aligned}\min|N_{u,t}(d)| &= 2^{k(1-\frac{2b}{3})H\left(\frac{\frac{b}{3}}{1-\frac{2b}{3}}\right)} \\ &\approx \binom{k-\frac{2bk}{3}}{\frac{bk}{3}} \\ &= \binom{k-\frac{2d}{3}}{\frac{d}{3}}\end{aligned} \tag{23}$$

where line (23) holds for large k, using the entropy approximation $\binom{n}{k} = 2^{n(H(p)+o(1))}$. If $b < 0.115$, then $f(b)$ is minimal at point (5), $(b, 2b)$, yielding

$$\min|N_{u,t}(d)| = 2^{k2b} = 4^d$$

When $b > \frac{1}{2}$, only bounds (2,3,4,and 8) apply to $f(b)$, yielding 4 points that we need to examine, as shown in Figure 11.

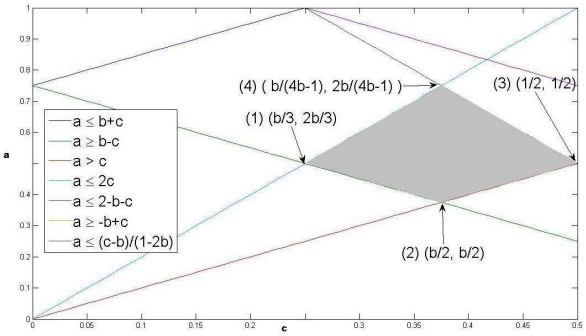

Fig. 11. Boundaries of $f(b)$ when $b \geq \frac{1}{2}$

For this region, $f(b)$ is minimal at point (1), matching point (5) in the previous region, yielding

$$\begin{aligned}\min|N_{u,t}(d)| &= 2^{k(1-\frac{2b}{3})H\left(\frac{\frac{b}{3}}{1-\frac{2b}{3}}\right)} \\ &\approx \binom{k-\frac{2d}{3}}{\frac{d}{3}}\end{aligned} \tag{24}$$

where line (24) holds for large k, using the entropy approximation $\binom{n}{k} = 2^{n(H(p)+o(1))}$. Thus, when $b < 0.115$, we have $\min|N_{u,t}(d)| = 4^d$, and when $b \geq 0.115$, we have $\min|N_{u,t}(d)| = \binom{k-\frac{2d}{3}}{\frac{d}{3}}$. Finally, we have that when $c < a + b(1-2a)$, we apply Solution 2, and we have $\min|N_{u,t}(d)| = \binom{k-\frac{2d}{3}}{\frac{d}{3}}$ when Solution 2 is valid. Comparing the Solution 1 with Solution 2, we have again that $\min|N_{u,t}(d)| = \binom{k-\frac{2d}{3}}{\frac{d}{3}}$. ■

X. Appendix B - Concavity of $f(a,b,c)$ for Solution 2

Lemma 10.1: The function $f(a,b,c) = aH\left(\frac{a-b+c}{2a}\right) + (1-a)H\left(\frac{-a+b+c}{2(1-a)}\right)$ is concave in both a and b.

Proof: To prove that the function is concave in both a and b, we need to see if the Hessian is negative definite. Let

$$f(a,b,c) = aH\left(\frac{a-b+c}{2a}\right) + (1-a)H\left(\frac{-a+b+c}{2(1-a)}\right)$$

Taking derivatives with respect to a, we find

$$\begin{aligned}\frac{\partial f}{\partial a} =& \frac{1}{2}\left(-\log\frac{c+a-b}{2a} - \log\frac{b+a-c}{2a}\right. \\ &\left. + \log\frac{c-a+b}{2(1-a)} + \log\frac{2-a-b-c}{2(1-a)}\right)\end{aligned}$$

and

$$\begin{aligned}\frac{\partial^2 f}{\partial a^2} =& \frac{-(c-b)^2}{a(c+a-b)(b+a-c)} \\ &+ \frac{-(1-c-b)^2}{(1-a)(c-a+b)(2-a-b-c)} \\ =& \frac{1}{2}\left(\frac{-1}{a-b+c} + \frac{-1}{a+b-c} + \frac{2}{a}\right. \\ &\left. + \frac{-1}{-a+b+c} + \frac{-1}{2-a-b-c} + \frac{2}{1-a}\right)\end{aligned}$$

From the bounds for this region, we can see that the function is concave in a.

Taking derivatives with respect to c, we find

$$\begin{aligned}\frac{\partial f}{\partial c} =& \frac{1}{2}\left(-\log(c+a-b) + \log(a+b-c)\right. \\ &\left. - \log(c-a+b) + \log(2-a-b-c)\right)\end{aligned}$$

and

$$\begin{aligned}\frac{\partial^2 f}{\partial c^2} =& \frac{1}{2}\left(\frac{-1}{c+a-b} + \frac{-1}{a+b-c}\right. \\ &\left. + \frac{-1}{c-a+b} + \frac{-1}{2-a-b-c}\right)\end{aligned}$$

From the bounds in this region, we can see that the function is concave in c.

Taking derivatives with respect to both a and c, we find

$$\frac{\partial^2 f}{\partial c \partial a} = \frac{1}{2}\left(\frac{-1}{a-b+c} + \frac{1}{a+b-c} + \frac{1}{-a+b+c} + \frac{-1}{2-a-b-c}\right)$$

The Hessian H is

$$\begin{pmatrix} \frac{\partial^2}{\partial a^2} & \frac{\partial^2}{\partial a \partial c} \\ \frac{\partial^2}{\partial a \partial c} & \frac{\partial^2}{\partial c^2} \end{pmatrix}$$

We want to show that the Hessian is negative definite, i.e, that $H < 0$. We have already shown that $\frac{\partial^2}{\partial a^2} < 0$, so it remains to show that the second leading principal minor of H is positive definite. This is just the determinant of H

$$det[H] = \frac{\partial^2}{\partial a^2}\frac{\partial^2}{\partial c^2} - \left(\frac{\partial^2}{\partial a \partial c}\right)^2 > 0$$

We rewrite the second derivatives as

$$\frac{\partial^2}{\partial a^2} = \frac{1}{2}\left(f_1 + f_2 + \frac{2}{a} + \frac{2}{1-a}\right)$$
$$\frac{\partial^2}{\partial c^2} = \frac{1}{2}(f_1 + f_2)$$
$$\frac{\partial^2}{\partial a \partial c} = \frac{1}{2}(f_1 - f_2)$$

where, from above,

$$f_1 = \frac{-1}{a-b+c} + \frac{-1}{2-a-b-c} < 0$$
$$f_2 = \frac{-1}{a+b-c} + \frac{-1}{-a+b+c} < 0$$

So, our determinant is now

$$\begin{aligned} det[H] &= \left(f_1 + f_2 + \frac{2}{a} + \frac{2}{1-a}\right)(f_1 + f_2) - (f_1 - f_2)^2 \\ &= \frac{1}{4}(f_1 + f_2)^2 + \frac{1}{4}\left(\frac{2}{a} + \frac{2}{1-a}\right)(f_1 + f_2) \\ &\quad - \frac{1}{4}(f_1 - f_2)^2 \\ &= f_1 f_2 + \frac{(f_1 + f_2)}{2a(1-a)} \end{aligned}$$

Simplifying, this is just

$$\begin{aligned} det[H] = &- (-a-b+c+2ab)^2 \\ &[(a-b+c)(-2+a+b+c)(a+b-c) \\ &(a-b-c)a(a-1)]^{-1} \end{aligned}$$

which, from our bounds, is positive. Since the determinant of H is positive, and since $\frac{\partial^2}{\partial a^2}$ is negative, we can say that H is negative definite, and the function is concave in both a and c. ■

REFERENCES

[1] P. Erdős and A. Rényi, "On random graphs," *Publicationes Mathematicae 6*, p. 290297, 1959.

[2] E.Bodine, B. Hassibi, and A. Weirman, "Generalizing kronecker graphs in order to model searchable networks," in *Proc. Forty-Seventh Annual Allerton Conference*, 2009.

[3] M.E.J. Newman, "The structure and function of complex networks," *SIAM Review*, 2003.

[4] R. Albert and A. Barabási, "Statistical mechanics of complex networks," *Reviews of Modern Physics*, vol. 74, 2002.

[5] J. Kleinberg, "Complex networks and decentralized search algorithms," in *Proc. of International Conference of Mathematicians*, 2006.

[6] B. Bollobás, *Random Graphs*, Academic Press, Inc., 1985.

[7] D.J. Watts and S.H. Strogatz, "Collective dynamics of small-world networks," *Nature*, vol. 393, pp. 440, 1998.

[8] A. Barabási and R. Albert, "Emergence of scaling in random networks," *Science*, vol. 286, pp. 509–512, 1999.

[9] J. Leskovec, J. Kleinberg, and C. Faloutsos, "Graphs over time: densification laws, shrinking diameters and possible explanations," in *Proc. of ACM SIGKDD conf. on knowledge discovery in data mining*, 2005, pp. 177–187.

[10] P. Bak, K. Chen, and C. Tang, "A forest-fire model and some thoughts on turbulence," *Phys. Lett. A*, vol. 147, pp. 297300, 1990.

[11] J. Leskovec, D. Chakrabarti, J. Kleinberg, and C. Faloutsos, "Realistic, mathematically tractable graph generation and evolution, using kronecker multiplication," in *Conf. on Principles and Practice of Knowledge Discovery in Databases*, 2005.

[12] J. Leskovec, *Dynamics of Large Networks*, PhD in Computer Science, Carnegie Mellon University, 2008.

[13] S. Migram, "The small-world problem," *Psychology Today*, vol. 2, pp. 60, 1967.

[14] J. Kleinberg, "The small-world phenomenon: An algorithmic perspective," in *Proc. of the 32nd ACM Symposium on Theory of Computing*, 2000, pp. 163–170.

[15] C. Martel and V. Nguyen, "Analyzing Kleinberg's and other small-world models," in *In PODC '04: Proc. of ACM symposium on Principles of distributed computing*, 2004, pp. 179–188.

[16] M. Mahdian and Y. Xu, "Stochastic kronecker graphs," in *In WAW07: Proc. of Workshop On Algorithms And Models For The Web-Graph*, 2007, pp. 179–186.

[17] M. Ángeles Serrano, D. Krioukov, and M. Boguñá, "Self-similarity of complex networks and hidden metric spaces," *Physics Review Letters*, vol. 100, 2008.

[18] D. Krioukov, F. Papadopoulos, M. Boguñá, and A. Vahdat, "Greedy forwarding in scale-free networks embedded in hyperbolic metric spaces," in *Proc. of MAMA Workshop at Sigmetrics*, 2009.

[19] A. Cvetkovski and M. Crovella, "Hyperbolic embedding and routing for dynamic graphs," in *Proc. of Infocom*, 2009.

[20] V. Ramasubramanian and D. Malkhi, "On the treeness of internet latency and bandwidth," in *Proc. of ACM Sigmetrics*, 2009, pp. 61–72.

[21] M. Boguñá and D. Krioukov, "Navigating ultrasmall worlds in ultrashort time," *Physical Review Letters*, vol. 102, pp. 058701, 2009.

[22] M. Jackson, *Social and Economic Networks*, Princeton University Press, 2008.

[23] M. Boguñá, D. Krioukov, and kc claffy, "Navigability of complex networks," *Nature Physics*, vol. 5, pp. 74–80, 2009.

[24] D. Krioukov, F. Papadopoulos, A. Vahdat, and M. Boguñá, "Curvature and temperature of complex networks," *Physical Review E*, vol. 80, pp. 635101, 2009.