Generalizing Kronecker Graphs in order to Model Searchable Networks

Elizabeth Bodine, Babak Hassibi, Adam Wierman
California Institute of Technology
Pasadena, CA 91125
Email: {eabodine, hassibi, adamw}@caltech.edu

***Abstract*—This paper describes an extension to stochastic Kronecker graphs that provides the special structure required for searchability, by defining a "distance"-dependent Kronecker operator. We show how this extension of Kronecker graphs can generate several existing social network models, such as the Watts-Strogatz small-world model and Kleinberg's lattice-based model. We focus on a specific example of an expanding hypercube, reminiscent of recently proposed social network models based on a hidden hyperbolic metric space, and prove that a greedy forwarding algorithm can find very short paths of length $O((\log\log n)^2)$ for graphs with n nodes.**

I. Introduction

There exists a large body of work exploring the various structural properties of social networks – small diameter, high clustering, heavy-tailed degree distributions, and searchability; see [1], [2], and [3] for surveys of this area. Many generative models have been proposed that capture some of these properties with varying levels of complexity, but the challenge remains to develop a simple, quantitative model that can exhibit all of these properties. For example, the simple Erdös-Rényi random graph maintains a small diameter, but fails to capture many of the other properties [4], [5]. The combination of small diameter and high clustering is often called the "small-world effect," and Watts and Strogatz (see section 2.2) generated much interest on the topic when they proposed a model that maintains these two characteristics simultaneously [6]. Several models were then proposed to explain the heavy-tailed degree distributions and densification of complex networks; these include the preferential attachment model [7], the forest-fire model [8], [9], Kronecker graphs [10] [11], and many others [1]. As demonstrated by Milgram's 1967 experiment using real people, individuals can discover and use short paths using only local information [12]. This is termed "searchability." Kleinberg focuses on this characteristic in his lattice model and proves searchability for a precise set of input parameters, but his model lacks any heavy-tailed distributions [13], [3], [14].

One promising model proposed recently is Kronecker graphs [10], [11], [15]. These graphs are simple to generate, are mathematically tractable, and have been shown to exhibit several important social network characteristics. In particular, these graphs can have heavy-tailed degree and eigenvalue distributions, a high-clustering coefficient, small diameter, and densify over time. Additionally, Leskovec developed an algorithm that could find an appropriate 2x2 or 3x3 initiator matrix to fit real-world data. However, Kronecker graphs are not searchable by a distributed greedy algorithm [15].

The contribution of this work is to provide an extension of Kronecker graphs that maintains searchability. We provide a general modeling framework that should allow the exploration of the interaction of several of the properties mentioned above. Instead of using the traditional Kronecker product operation to grow the network, we introduce a "distance"-dependent Kronecker operation that iteratively grows the network from an initiator matrix. This extension allows for the generation of traditional models such as the Watts-Strogatz ring, the Kleinberg model, as well as the original Kronecker graphs. The key feature of this new model is that the likelihood of long-range contacts is now based on a "distance" measure, which preserves the natural distinction between local and long-range contacts. This additional structure allows the model to generate graphs that are searchable.

To illustrate this new model, we present a few examples and study one particular example in detail: an expanding hypercube. This example is chosen to mimic the defining features of tree metrics and hyperbolic space, which are thought to be representative of the Internet and social networks [16], [17], [18]. In these models and in ours, the key is to have exponentially expanding neighborhoods around each node, which leads to $O(\log n)$ diameter, even without long-range links. Adding long range links, the diameter can shrink to $O(\log\log n)$ [19]. In our expanding hypercube model, we use a $\log n$-dimensional hypercube as the underlying lattice, with "distance"-dependent long-range contacts added to each node. We "grow" our model by adding a dimension in each iteration, doubling the number of nodes. We then prove that local, greedy agents can find paths of length $O((\log\log n)^2)$.

This paper is organized as follows. Section II describes in detail our model and its relation to the original Kronecker graph model and other traditional models. Section III gives a detailed proof of searchability for the example of an expanding hypercube. Section IV concludes with proposed future work.

II. 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 are generated by recursively using a standard matrix operation, the Kronecker product[1] [10]. 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 2.1: The k^{th} power of P_1 is defined as the matrix $P_1^{\otimes k}$, such that:

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

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 [10]. 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 [10]. However, as shown by Mahdian and Xu, stochastic Kronecker graphs are not searchable by a distributed greedy algorithm [15] – 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.

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

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

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

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 $P(u, v) = p$ [6]. To generate the underlying ring structure with $k = 1$,

[1]For a description of deterministic Kronecker graphs, see Leskovec et al, [10].

start with an initiator matrix K_1, representing the graph in figure 1(a).

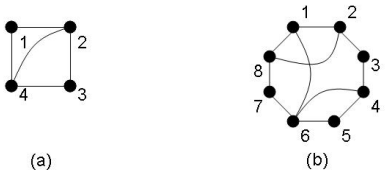

Fig. 1. Generating the Watts-Strogatz Model

In order to obtain the sequence of matrices representing 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:

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

$$G_2 = G_1 \otimes_d \mathcal{H}$$

$$G_2 = \begin{pmatrix} 1 \times H_0 & 1 \times H_1 & p \times H_2 & 1 \times H_{-1} \\ 1 \times H_{-1} & 1 \times H_0 & 1 \times H_1 & p \times H_2 \\ p \times H_2 & 1 \times H_{-1} & 1 \times H_0 & 1 \times H_1 \\ 1 \times H_1 & p \times H_2 & 1 \times H_{-1} & 1 \times H_0 \end{pmatrix}$$

$$G_2 = \begin{pmatrix} 1 & 1 & p & p & p & p & p & 1 \\ 1 & 1 & 1 & p & p & p & p & p \\ p & 1 & 1 & 1 & p & p & p & p \\ p & p & 1 & 1 & 1 & p & p & p \\ p & p & p & 1 & 1 & 1 & p & p \\ p & p & p & p & 1 & 1 & 1 & p \\ p & p & p & p & p & 1 & 1 & 1 \\ 1 & p & p & p & p & p & 1 & 1 \end{pmatrix}$$

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 [13], [3]. 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. We omit showing the resulting matrices for brevity. 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 section.

Example 3: Kleinberg-like Model. The final example we consider, Kleinberg's lattice model, is particularly pertinent as it was shown to be searchable [13]. 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 an "expanding hypercube" as our underlying lattice. In this example, at any point, the graph is a 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 [17], [16]. The expanding hypercube captures the core of these topologies, as the number of nodes at distance d grows exponentially in d. Interestingly, this example is also very naturally represented using our "distance"-dependent Kronecker operation and a Hamming distance as our "distance" measure.

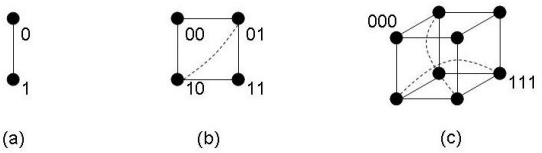

Fig. 2. Example: the growth of an expanding hypercube

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 β_1 = a normalizing constant, $\beta_i = \frac{P(i)}{P(i+1)}$. 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 $\beta_i = \left(\frac{i}{i+1}\right)^{-\alpha}$. 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 III), can find paths of length $O(\log n)$ only if $\alpha = k$ [13]; in the modified case presented above, we will see in the next section that we need a different probability of connection to find short paths.

III. Proving searchability for an expanding hypercube

While our model is more complicated than the original Kronecker graphs, it can capture several existing network models, and it incorporates "distance" into the probability of connection, allowing for several parameters in which searchability can be proven. Given the model of Example 3 in the previous section, we will show that is searchable by a simple greedy, distributed algorithm if the probability of a connection is inversely proportional to the number of nodes in an exponentially expanding torus.

We define a decentralized algorithm $\mathcal{A}$ similar to [13]. 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), 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 this distributed algorithm $\mathcal{A}$ be able to find short paths through the network of $O((\log\log n)^2) = O(\log^2 k)$. Recall that the hypercube itself provides paths of length $\log n$; the addition of random long-range links reduces the diameter, similar to [19]. We prove that for a particular definition of β_i, the algorithm $\mathcal{A}$ can find these paths. Specifically:

Theorem 3.1: If

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

such that the probability of a connection is

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

then the decentralized algorithm $\mathcal{A}$ will find paths of length $O((\log\log n)^2)$.

Proof: We will say that the execution of $\mathcal{A}$ is in phase j when

$$2^j < d(u,t) \leq 2^{j+1} \tag{1}$$

Thus, the largest value of j is $\log k$.

Suppose we are in phase j; we want to find out 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 $B_j = \{v : d(v,t) \leq 2^j\}$.

$$\Pr(\text{msg enters } B_j) = 1 - \prod_{v \in B_j} (1 - P(u, v : v \in B_j)) \tag{2}$$
$$= 1 - \prod_{d=\|u\|-2^j}^{\|u\|+2^j} (1 - P(d))^{|N_{u,t}(d)|} \tag{3}$$
$$\geq 1 - \prod_{d=\|u\|-2^j}^{\|u\|+2^j} (1 - P(d))^{\min|N(d)|} \tag{4}$$

where $N_{u,t}(d) = \{v : d(v,t) \leq 2^j, d(u,v) = d\}$

and $\min|N(d)| = \min_{u,t,d(u,t)=d} |N_{u,t}(d)|$

In any network model, enforcing searchibility 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 3. Once this $\min|N(d)|$ is found, if we set the

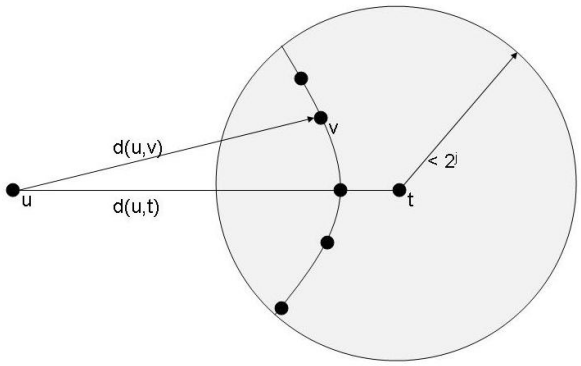

Fig. 3. Relative positions of nodes u,v, and t in phase j

probability of a connection between two nodes distance d apart to be inversely proportional to $\min|N(d)|$, with an appropriate constant $\frac{1}{d\log k\ln 3}$, we will find that each phase described above will end in approximately $\log k$ steps, and, as there are only $\log k$ such phases, our greedy forwarding algorithm will find be able to find very short paths of length $O(\log^2 k)$. 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 show in Lemma 5.1 in Appendix A that $\min|N(d)| = \binom{k-\frac{2d}{3}}{\frac{d}{3}}$. Continuing, we have

$$\begin{aligned}
\Pr(\text{msg enters } B_j) \geq & 1 - \prod_{d=\|u\|-2^j}^{\|u\|+2^j} (1-P(d))^{\min|N(d)|} && (5)\\
\approx & 1 - e^{-\sum_{d=\|u\|-2^j}^{\|u\|+2^j} \min|N(d)|P(d)} && (6)\\
= & 1 - e^{-\frac{1}{\log k\ln 3}\sum_{d=\|u\|-2^j}^{\|u\|+2^j} d^{-1}} && (7)\\
\geq & 1 - e^{-\frac{1}{\log k\ln 3}\ln\frac{\|u\|+2^j}{\|u\|-2^j}} && (8)\\
\geq & 1 - e^{-\frac{1}{\log k\ln 3}\ln\frac{3}{2}\frac{2^j}{2^j}} && (9)\\
= & 1 - e^{-\frac{1}{\log k}} && (10)\\
\geq & \frac{1}{\log k} && (11)
\end{aligned}$$

where line (6) is true in the limit of large n ($\lim_{n\to\infty}(1-x/n)^n = e^{-x}$), and line (11) comes from the power series expansion of e^{-x}. Let X_j denote the total number of steps spent in phase j. So,

$$\begin{aligned}
EX_j = & \sum_{i=1}^{\infty}\Pr[X_j \geq i] && (12)\\
\leq & \sum_{i=1}^{\infty}\left(1-\frac{1}{\log k}\right)^{i-1} && (13)\\
= & \log k && (14)
\end{aligned}$$

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

$$X = \sum_{j=0}^{\log k} X_j \quad (15)$$

and

$$\begin{aligned}
EX = & \sum_{j=0}^{\log k} EX_j && (16)\\
\leq & (1+\log k)(\log k) && (17)\\
\leq & \delta(\log k)^2, \text{ for a large enough } \delta && (18)
\end{aligned}$$

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. ■

IV. Conclusion

We have presented a generalization of Kronecker Graphs that maintains several of their advantageous properties while additionally allowing the generation of graphs that are searchable. In general, by defining a family of "distance"-dependent matrices, used with a "Kronecker-like" operation to grow a network, we can define both local regular structures and global distance-dependent long-range connections. Even though this model is more complicated than the original Kronecker graph formulation, we gain generality (as we can generate other social network models) and searchability (as shown by our example). This general modeling framework should allow us to explore the interaction of properties such as searchability and heavy-tailed degree distributions. Other recent models have attempted to explore this interaction by imposing a power-law degree distribution onto existing models like the Kleinberg lattice [20]. Distance-dependent Kronecker graphs, in contrast, allow for these properties to emerge naturally from the definition of the family of distance-dependent matrices. The expanding hypercube example in Section III bears a resemblance to recent models proposed based on hidden hyperbolic metric spaces, which have been shown to be representative of real-world complex networks [16]. In [16], the observed network topology is based upon a hidden metric space, assumed to be a non-Euclidean hyperbolic space. As in our model, this leads to scale-free topologies and very efficient greedy forwarding, though searchability has only been shown via simulations [19].

This paper should be viewed as a first step towards the analysis of "distance"-dependent Kronecker graphs. There are many interesting questions that remain, including how to parameterize our model from real-world datasets. Ideally, given any dataset, we would like to be able to find an appropriate family of "distance"-dependent matrices to match any desired characteristic of the dataset. We would also like to use our model to examine the interaction of searchability and other properties, particularly degree distributions. As our model was not designed to demonstrate one characteristic in particular, but rather a range of characteristics, it is ideally suited to this sort of analysis. Finally, we would also like to investigate the dynamics of complex networks within this model – determining how we could incorporate growth and mobility into our model.

References

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

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

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

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

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

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

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

[8] 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.
[9] P. Bak, K. Chen, and C. Tang, "A forest-fire model and some thoughts on turbulence," *Phys. Lett. A*, vol. 147, pp. 297300, 1990.
[10] 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.
[11] J. Leskovec, *Dynamics of Large Networks*, PhD in Computer Science, Carnegie Mellon University, 2008.
[12] S. Migram, "The small-world problem," *Psychology Today*, vol. 2, pp. 60, 1967.
[13] J. Kleinberg, "The small-world phenomenon: An algorithmic perspective," in *Proc. of the 32nd ACM Symposium on Theory of Computing*, 2000, pp. 163–170.
[14] 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.
[15] 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.
[16] D. Krioukov, F. Papadopoulos, M. Boguna, and A. Vahdat, "Greedy forwarding in scale-free networks embedded in hyperbolic metric spaces," in *Proc. of MAMA Workshop at Sigmetrics*, 2009.
[17] A. Cvetkovski and M. Crovella, "Hyperbolic embedding and routing for dynamic graphs," in *Proc. of Infocom*, 2009.
[18] V. Ramasubramanian and D. Malkhi, "On the treeness of internet latency and bandwidth," in *Proc. of ACM Sigmetrics*, 2009, pp. 61–72.
[19] M. Boguna and D. Krioukov, "Navigating ultrasmall worlds in ultrashort time," *Physical Review Letters*, vol. 102, pp. 058701, 2009.
[20] P. Fraigniaud and G. Giakkoupis, "The effect of power-law degrees on the navigability of small worlds," in *28th ACM Symposium on Principles of Distributed Computing (PODC)*, 2009.

V. Appendix - 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 5.1: $\min |N_{u,t}(d)| = \binom{k-\frac{2d}{3}}{\frac{d}{3}}$

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 3. Without loss of generality, define t as the all zero node, $t = (00...0)$. Arrange the label of u such that $u = (1 \ldots 1\ 0 \ldots 0)$. 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:

$$v_{11} + v_{10} + v_{01} + v_{00} = k \tag{19}$$
$$v_{11} + v_{10} = \|u\| \tag{20}$$
$$v_{01} + v_{10} = d \tag{21}$$
$$v_{11} + v_{01} = \|v\| \tag{22}$$

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

$$v_{00} = k - d - v_{11} \tag{23}$$
$$v_{10} = \|u\| - v_{11} \tag{24}$$
$$v_{01} = d - \|u\| + v_{11} \tag{25}$$

We also know that we must satisfy the following:

$$v_{11}, v_{10}, v_{01}, v_{00} \geq 0 \tag{26}$$
$$2^j < \|u\| \leq 2^{j+1} \tag{27}$$
$$\|u\| - 2^j \leq d \leq \|u\| + 2^j \tag{28}$$
$$\|v\| \leq 2^j \tag{29}$$

From these bounds we have

$$\max(0, \|u\| - d) \leq v_{11} \leq \min(\|u\|, k-d, \frac{1}{2}(2^j + \|u\| - d)) \tag{30}$$

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

$$|N_{u,t}(d)| = \sum_{v_{11}=v_l}^{v_u} \binom{\|u\|}{v_{11}} \binom{k - \|u\|}{d - \|u\| + v_{11}} \tag{31}$$

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

$$|N_{u,t}(d)| = \sum_{x=v_l}^{v_u} \binom{ak}{x} \binom{k(1-a)}{k(1-b)+x} \tag{32}$$
$$\approx \sum_{x=v_l}^{v_u} 2^{k\left(aH\left(\frac{x}{ak}\right) + (1-a)H\left(\frac{b-a+\frac{x}{k}}{1-a}\right)\right)} \tag{33}$$
$$\geq 2^{k\mathcal{X}} \tag{34}$$

where

$$\mathcal{X} = \max_x aH\left(\frac{x}{ak}\right) + (1-a)H\left(\frac{b-a+\frac{x}{k}}{1-a}\right) \tag{35}$$

subject to

$$k * \max(0, a-b) \leq x \leq k * \min(a, 1-b, \frac{1}{2}(a-b+c)) \tag{36}$$

Note that line (34) is true as $\binom{n}{k} = 2^{n(H(p)+o(1))}$ when $k \propto pn$.

There are two solutions to the optimization problem stated above, yielding two 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 function is concave in x, so the two possible solutions can be seen from the boundary points and the bounds for the region. 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 $N_{u,t}(d)$, we have

$$|N_{u,t}(d)| = 2^{k(aH\left(\frac{ak-abk}{ak}\right)+(1-a)H\left(\frac{b-a+\frac{ak-abk}{k}}{1-a}\right))} \quad (37)$$
$$= 2^{k(aH(1-b)+(1-a)H(b))} \quad (38)$$
$$= 2^{kH(b)} \quad (39)$$
$$\approx \binom{k}{bk} \quad (40)$$
$$= \binom{k}{d} \quad (41)$$

where line (41) is true in the limit of large k.

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

$$|N_{u,t}(d)| = 2^{k(aH\left(\frac{a-b+c}{2a}\right)+(1-a)H\left(\frac{-a+b+c}{2(1-a)}\right))} \quad (42)$$

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

$$f(b) = \min |N_{u,t}(d)| \quad (43)$$
$$= \min_{a,c} aH\left(\frac{a-b+c}{2a}\right) + (1-a)H\left(\frac{-a+b+c}{2(1-a)}\right) \quad (44)$$

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 (45) shows that it is concave in both a and b; however, we omit the cumbersome calculation for brevity. 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 $b > \frac{1}{2}$

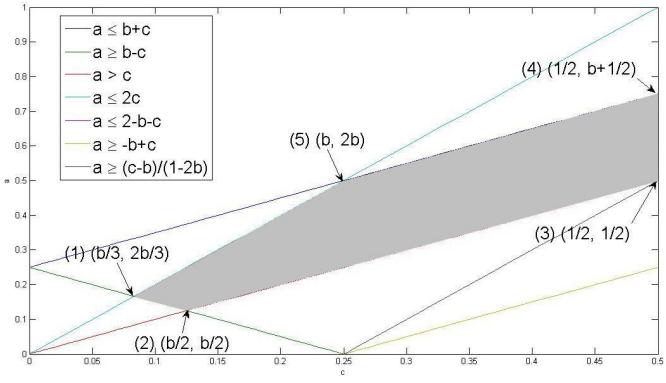

Fig. 4. Boundaries of $f(b)$ when $b \leq \frac{1}{2}$

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 4. If $b \geq .115$, then $f(b)$ is minimal at point (1), $(\frac{b}{3}, \frac{2b}{3})$, yielding

$$\min |N_{u,t}(d)| = 2^{k(1-\frac{2b}{3})H\left(\frac{\frac{b}{3}}{1-\frac{2b}{3}}\right)} \quad (45)$$
$$\approx \binom{k - \frac{2bk}{3}}{\frac{bk}{3}}, \text{for large } k \quad (46)$$
$$= \binom{k - \frac{2d}{3}}{\frac{d}{3}} \quad (47)$$

If $b < 0.115$, then $f(b)$ is minimal at point (5), $(b, 2b)$, yielding

$$\min |N_{u,t}(d)| = 2^{k2b} = 4^d \quad (48)$$

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 5. For this region, $f(b)$ is minimal at point (1), matching point

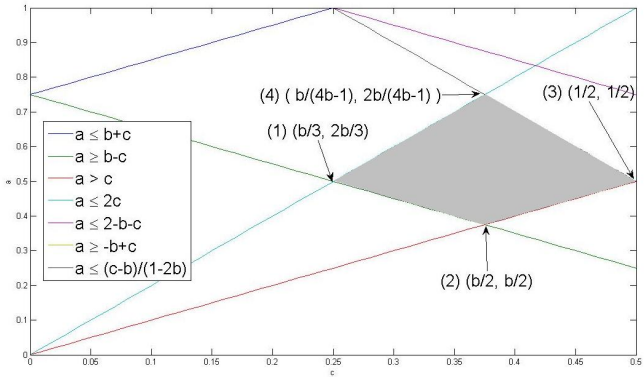

Fig. 5. Boundaries of $f(b)$ when $b \geq \frac{1}{2}$

(5) in the previous region, yielding

$$\min |N_{u,t}(d)| = 2^{k(1-\frac{2b}{3})H\left(\frac{\frac{b}{3}}{1-\frac{2b}{3}}\right)} \tag{49}$$

$$\approx \binom{k-\frac{2d}{3}}{\frac{d}{3}}, \text{for large } k \tag{50}$$

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 from Lemma 5.1, 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 $\min |N_{u,t}(d)| = \binom{k-\frac{2d}{3}}{\frac{d}{3}}$. ■