

# Deterministic Decentralized Search in Random Graphs

Esteban Arcaute, Ning Chen, Ravi Kumar, David Liben-Nowell,  
Mohammad Mahdian, Hamid Nazerzadeh, and Ying Xu

**Abstract.** We study a general framework for decentralized search in random graphs. Our main focus is on deterministic memoryless search algorithms that use only local information to reach their destination in a bounded number of steps in expectation. This class includes (with small modifications) the search algorithms used in Kleinberg's pioneering work on long-range percolation graphs and hierarchical network models. We give a characterization of searchable graphs in this model, and use this characterization to prove a monotonicity property for searchability.

## 1. Introduction

Since Milgram's famous "small world" experiment [Milgram 67], it has generally been understood that social networks have the property that a typical node can reach any other node through a short path (the so-called "six degrees of separation"). An implication of this fact is that social networks have small diameter. Many random graph models have been proposed to explain this phenomenon, often by showing that adding a small number of random edges causes a highly structured graph to have a small diameter (see, for example, [Watts and Strogatz 98, Bollobás and Chung 88]). A stronger implication of Milgram's experiment, as Kleinberg observed [Kleinberg 00], is that for most social networks there are decentralized search algorithms that can *find* a short path from a source to a destination without a global knowledge of the graph. As Kleinberg proved, many

of the random graph models with small diameter do not have this property (i.e., any decentralized search algorithm in such graphs can take many steps to reach the destination), while in certain graph models, with a delicate balance of parameters, decentralized search is possible. Since Kleinberg's work, there have been many other models that provably exhibit the searchability property [Kumar et al. 06, Fraigniaud 05, Slivkins 05, Liben-Nowell et al. 05, Kleinberg 01, Duchon et al. 05]; however, we still lack a good understanding of what contributes to this property in graphs.

In this paper, we look at a general framework for searchability in random graphs. We consider a general random graph model in which the set of edges leaving a node  $u$  is independent of that of any other node  $v \neq u$ . This framework includes models such as the directed variant of the classical Erdős–Rényi graphs [Erdős and Rényi 59], random graphs with a given expected degree sequence (e.g., [Chung and Lu 03]), ACL graphs [Aiello et al. 01], long-range percolation graphs [Kleinberg 00], hierarchical network models [Kleinberg 01], and graphs based on Kronecker products [Leskovec et al. 05, Mahdian and Xu 07], but not models such as preferential attachment [Barabási and Albert 99] in which the distribution of edges leaving a node is dependent on the other edges of the graph. It is worth noting that in a random graph model where edges can have arbitrary dependencies, the search problem includes difficult learning problems as special cases, and therefore one cannot expect to have a complete characterization of searchable graphs in such a model.<sup>1</sup>

Throughout most of this paper, we restrict the class of decentralized search algorithms that we consider to deterministic memoryless algorithms that succeed in finding a path to the destination with probability one. This is an important class of search algorithms, and includes the decentralized search algorithms used in Kleinberg's work on long-range percolation graphs and hierarchical network models. For this class, we give a simple characterization of graphs that are searchable in terms of a node-ordering property. We will use this characterization to show a monotonicity property for searchability: if a graph is searchable in our model, it stays searchable if the probabilities of edges are increased.

The rest of this paper is organized as follows: Section 2 contains the description of the model. Section 3 presents a characterization of searchable random graphs. The monotonicity theorem is presented in Section 4.

---

<sup>1</sup>For example, the graph can include a sequence of layers, where each node in each layer is connected to the same set of nodes in the next layer, and this set encodes a labeled example  $(x, f(x))$  of an unknown function picked from a prior distribution. For the last layer, however, this set encodes only the input  $x$  and not the output  $f(x)$ . At this point, the algorithm faces a choice between a number of nodes as the next node to go to, each corresponding to one value of  $f(x)$ , and only the node that corresponds to the correct value of  $f(x)$  has a short path to  $t$ .

## 2. The Model

We define a random graph model parameterized by a positive integer  $n$  (the size of the graph) and  $n$  independent distributions  $\Omega_1, \Omega_2, \dots, \Omega_n$ . For each  $i \in \{1, \dots, n\}$ ,  $\Omega_i$  is a distribution over the collection of all subsets of  $\{1, \dots, n\}$ . The random digraph  $G(n, \Omega)$  is defined as follows: the node set of this graph is  $V = \{1, \dots, n\}$ , and for every  $i$ , the set of nodes that have an edge from  $i$  (i.e., the *out-neighbors* of  $i$ ) is picked (independently) from the distribution  $\Omega_i$ . For  $i \in V$ , let  $\Gamma(i)$  denote the set of out-neighbors of  $i$ . We denote by  $\omega_{i,S}$  the probability that  $\Gamma(i) = S$ .

This graph model is quite general and includes many random graph models such as those proposed in [Chung and Lu 03, Aiello et al. 01, Kleinberg 00, Kleinberg 01, Leskovec et al. 05, Mahdian and Xu 07]. For example, for the long-range percolation graphs [Kleinberg 00], the support of the distribution  $\Omega_i$  is all possible subsets consisting of all nodes with Hamming distance 1 to  $i$  in the lattice and one extra “long-range” node  $j$ , and the probability of any such set is a decreasing function of the distance between  $i$  and  $j$ .

A special case of this model that deserves special attention is that in which all edges of the graph are independent. In this case, given a positive integer  $n$  and an  $n \times n$  matrix  $\mathbf{P}$  with entries  $p_{i,j} \in [0, 1]$ , we define a directed random graph  $G(n, \mathbf{P})$  with the node set  $V = \{1, \dots, n\}$  and with a directed edge connecting node  $i$  to node  $j$  with probability  $p_{ij}$ , independently of all other edges. Note that this is a special case of the  $G(n, \Omega)$  random graph model, with  $\omega_{i,S} := \prod_{j \in S} p_{ij} \prod_{j \notin S} (1 - p_{ij})$ .

We fix two nodes  $s, t \in V$  of  $G(n, \Omega)$  as the *source* and the *destination*. We investigate the existence of a decentralized search algorithm that finds a path from  $s$  to  $t$  of at most a given length  $d$  in expectation.<sup>2</sup> We restrict our attention to *deterministic memoryless algorithms*. A deterministic memoryless algorithm can be defined as a partial function  $A : V \times 2^V \rightarrow V$ . Such an algorithm  $A$  defines a path  $v_0, v_1, v_2, \dots$  on a given graph  $G$  as follows:  $v_0 = s$ , and for every  $i \geq 0$ ,  $v_{i+1} = A(v_i, \Gamma(v_i))$ . The length of this path is defined as the smallest integer  $i$  such that  $v_i = t$ . If no such  $i$  exists, we define the length of the path as infinity.

We are now ready to define the notion of searchability. For a given  $(n, \Omega)$ , source and destination nodes  $s$  and  $t$ , and a number  $d$ , we say that  $G(n, \Omega)$  is *d-searchable* using a deterministic memoryless algorithm  $A$  if the expected length

<sup>2</sup>Alternatively, we could ask for which graphs a decentralized search algorithm can find a path between *every* pair of nodes  $s$  and  $t$ , or between a *random* pair of nodes  $s$  and  $t$ . Our techniques apply to these alternative formulations of the problem as well. The only point that requires some care is that the orderings in the characterization theorem can depend on  $s$  and  $t$ .

of the path defined by  $A$  on  $G(n, \Omega)$  is at most  $d$ . Note that this definition requires the algorithm to find a path from  $s$  to  $t$  with probability one.

### 3. A Characterization of Searchable Random Graphs

In this section, we provide a complete characterization of searchable random graphs. We begin by defining a class of deterministic memoryless search algorithms parameterized by two orderings of  $V$ , and then prove that if a graph is  $d$ -searchable, it is also  $d$ -searchable using an algorithm from this narrow class.

**Definition 3.1.** Let  $\sigma, \pi$  be two orderings (i.e., permutations) of the node set  $V$ . We define a deterministic memoryless algorithm  $A_{\sigma, \pi}$  corresponding to these orderings as follows: for every  $u \in V$ ,  $A_{\sigma, \pi}(u, \Gamma(u))$  is defined as the maximum element according to  $\pi$  of the set  $\{v \in \Gamma(u) : \sigma(v) > \sigma(u)\}$ .

In other words, algorithm  $A_{\sigma, \pi}$  never goes backward according to the ordering  $\sigma$ , and, subject to this restriction, makes the maximum possible progress according to  $\pi$ .

Before stating our main result, we comment on why the class of search algorithms we are considering is defined based on two permutations and not just one. Common intuition based on the known results (e.g., on the long-range percolation model [Kleinberg 00], or the hierarchical network models [Kleinberg 01]) might lead one to conjecture that it is enough to consider decentralized search algorithms that always try to get as close to the destination as possible according to a *single* ordering of the nodes. This, however, is not true, as the following simple example shows.

**Example 3.2.** Consider a graph with the node set  $\{s = u_1, u_2, \dots, u_n = t\}$ . For every  $i, j \in \{1, \dots, n/2\}, i \neq j$ , there is an edge from  $i$  to  $j$  with probability one. For  $i = n/2, \dots, n-1$ , there is an edge from  $u_i$  to  $u_{i+1}$  with probability one. Finally, for  $i = 2, \dots, n/2-1$ , there is an edge from  $u_i$  to  $t$  with probability  $\frac{1}{2}$ . One can find a path in this graph using a deterministic memoryless search algorithm as follows: The algorithm traverses the path  $su_2u_3 \dots u_{n/2-1}$  in this order until it finds the first node that has a direct edge to  $t$ . If it finds such a node, it goes to  $t$ ; otherwise, it takes the path  $u_{n/2}u_{n/2+1} \dots u_n$  to  $t$ . The expected length of this path is  $2 + \frac{1}{2} + \frac{1}{4} + \dots + \frac{1}{2^{n/2-1}} + \frac{n}{2^{n/2}} < 4$ . However, a memoryless algorithm that is based on a single ordering cannot visit more than one of the nodes in the set  $S = \{u_2, \dots, u_{n/2-1}\}$ . This is because if the first node in  $S$  that the algorithm visits is  $u_i$ , then  $u_i$  must be ahead of all other nodes of  $S$  in the

ordering. Hence, if the algorithm visits another  $u_j \in S$ , as its next step it must go back to  $u_i$  (since it has the exact same set of choices that it had at the step at which it first visited  $u_i$ ), and therefore it will fall into a loop. Therefore, such an algorithm cannot achieve an expected distance smaller than  $n/4$ .

We are now ready to state our main theorem.

**Theorem 3.3.** *For a given  $n$ , collection of out-neighbor distributions  $\Omega$ , source and destination nodes  $s$  and  $t$ , and number  $d$ , if  $G(n, \Omega)$  is  $d$ -searchable using a deterministic memoryless algorithm  $A$ , then there exist two orderings  $\sigma$  and  $\pi$  of  $V$  such that  $G(n, \Omega)$  is  $d$ -searchable using  $A_{\sigma, \pi}$ .*

To prove this theorem, we first construct the ordering  $\sigma$  using the structure of the search algorithm  $A$ . Next, we define an ordering  $\pi$  using  $\sigma$ . Finally, we use induction with respect to the ordering  $\sigma$  to show that the expected length of the path defined by  $A_{\sigma, \pi}$  on  $G(n, \Omega)$  is not more than the one defined by  $A$ .

We assume, without loss of generality, that for every set  $S \subseteq V$ ,  $A(t, S) = t$ . In other words, we assume that  $A$  never leaves  $t$  once it reaches this node.

Define a graph  $H$  with the node set  $V$  as follows: for every pair  $u, v \in V$ , the edge  $(u, v)$  is in  $H$  if and only if this edge is on the path from  $s$  to  $t$  defined by  $A$  on some realization of  $G(n, \Omega)$  (i.e., on some graph that has a nonzero probability in the distribution  $G(n, \Omega)$ ). We have the following important lemma.

**Lemma 3.4.** *The graph  $H$  is acyclic.*

**Proof.** Assume, for contradiction, that  $H$  contains a simple cycle  $C$ . Note that by the definition of  $H$ , if an edge  $(u, v)$  is in  $H$ , then  $u$  must be reachable from  $s$  in  $H$ . Therefore, every node of  $C$  must be reachable from  $s$  in  $H$ . Let  $v^*$  be a node in  $C$  that has the shortest distance from  $s$  in  $H$ , and let  $s = v_0, v_1, \dots, v_\ell = v^*$  be a shortest path from  $s$  to  $v^*$  in  $H$ . Also, let  $v^* = v_\ell, v_{\ell+1}, \dots, v_k, v_{k+1} = v^*$  denote the cycle  $C$ . Therefore,  $v_0, v_1, \dots, v_k$  are all distinct nodes, and for every  $i \in \{0, \dots, k\}$ , there is an edge from  $v_i$  to  $v_{i+1}$  in  $H$ .

By the definition of  $H$ , for every  $i \in \{0, \dots, k\}$ , there is a realization of  $G(n, \Omega)$  in which  $A$  traverses the edge  $(v_i, v_{i+1})$ . This means that there is a realization of  $G(n, \Omega)$  in which the set  $\Gamma(v_i)$  of out-neighbors of  $v_i$  is  $S_i^*$ , for some set  $S_i^*$  such that  $A(v_i, S_i^*) = v_{i+1}$ . Recall that by the definition of  $G(n, \Omega)$ , the random variables  $\Gamma(u)$  are all independent. Hence, since the  $v_i$ 's are all distinct and for each  $i$ , there is a realization satisfying  $\Gamma(v_i) = S_i^*$ , there must be a realization in which  $\Gamma(v_i) = S_i^*$  for all  $i$ . In this realization, the algorithm  $A$  falls in the cycle  $C$ , and therefore will never reach  $t$ . Thus the path found by  $A$  in this realization

is infinitely long, and therefore the expected length of the path found by  $A$  is infinite. This is a contradiction.  $\square$

By Lemma 3.4, we can find a topological ordering of the graph  $H$ . Furthermore, since by assumption  $t$  has no outgoing edge in  $H$ , we can find a topological ordering that places  $t$  last. Let  $\sigma$  be such an ordering; more precisely,  $\sigma$  is an ordering of  $V$  such that

- (i)  $t$  is the maximum element of  $V$  under  $\sigma$ ;
- (ii) for every edge  $(u, v)$  in  $H$ , we have  $\sigma(v) > \sigma(u)$ ; and
- (iii) all isolated nodes of  $H$  are placed at the beginning of  $\sigma$  in an arbitrary order, i.e.,  $\sigma(u) > \sigma(v)$  for any isolated node  $v$  and nonisolated node  $u$ .

By the definition of  $H$ , these conditions mean that the algorithm  $A$  (starting from the node  $s$ ) never traverses an edge  $(u, v)$  with  $\sigma(u) > \sigma(v)$ .

Given the ordering  $\sigma$ , we define  $r_u$  for every  $u \in V$  to be the expected time to reach  $t$  from  $u$  following the best path that does not backtrack with respect to  $\sigma$ . The value of  $r_u$  can be computed recursively as follows:

$$r_u = \begin{cases} 0 & \text{if } u = t, \\ 1 + \sum_{\substack{S \subseteq T_u \\ S \neq \emptyset}} q_{u,S} \cdot \min_{v \in S} \{r_v\} & \text{if } u \neq t \text{ and } q_{u,\emptyset} = 0, \\ \infty & \text{if } u \neq t \text{ and } q_{u,\emptyset} > 0, \end{cases} \quad (3.1)$$

where  $T_u := \{v : \sigma(v) > \sigma(u)\}$  and, for a set  $S \subseteq T_u$ , we write

$$q_{u,S} := \sum_{S': S' \cap T_u = S} \omega_{u,S'}$$

to denote the probability that the subset of nodes of  $T_u$  that are out-neighbors of  $u$  is precisely  $S$ .<sup>3</sup> Note that the above formula defines  $r_u$  in terms of  $r_v$  for  $\sigma(v) > \sigma(u)$ , and therefore the definition is well founded.

We can now define the ordering  $\pi$  as follows: Let  $\pi(u) > \pi(v)$  if  $r_u < r_v$ . Pairs  $u, v$  with  $r_u = r_v$  are ordered arbitrarily by  $\pi$ .

The final step of the proof is the following lemma, which we will prove by induction using the ordering  $\sigma$ . To state the lemma, we need a few pieces of notation. For a search algorithm  $B$ , let  $d(B, u)$  denote the expected length of the path that the algorithm  $B$ , started at node  $u$ , finds to  $t$ . Also, let  $V_0$  denote

<sup>3</sup>In the special case of  $G(n, \mathbf{P})$ , we have  $q_{u,S} := (\prod_{v \in S} p_{uv})(\prod_{v \in T_u \setminus S} (1 - p_{uv}))$ .

the set of nonisolated nodes of  $H$ , i.e.,  $V_0$  is the set of nodes that the algorithm  $A$  (started from  $s$ ) has a nonzero chance of reaching.

**Lemma 3.5.** *Let  $\sigma$  and  $\pi$  be the orderings defined as above. Then for every node  $u \in V_0$ , we have that  $d(A, u) \geq d(A_{\sigma, \pi}, u) = r_u$ .*

**Proof.** We prove this statement by induction on  $u$ , according to the ordering  $\sigma$ . The statement is trivial for  $u = t$ . We now show that for  $u \in V_0 \setminus \{t\}$ , if the statement holds for every node  $v \in V_0$  with  $\sigma(v) > \sigma(u)$  (i.e., for every  $v \in T_u \cap V_0$ ), then it also holds for  $u$ . Observe that for any deterministic memoryless algorithm  $B$ ,

$$d(B, u) = 1 + \sum_{\substack{S \subset V \\ S \neq \emptyset}} \omega_{u, S} \cdot d(B, B(u, S)). \quad (3.2)$$

This statement follows from the fact that the algorithm  $B$  is memoryless, and that  $\omega_{u, \emptyset} = 0$  since  $u \in V_0$ . Applying (3.2) to  $A_{\sigma, \pi}$  implies

$$\begin{aligned} d(A_{\sigma, \pi}, u) &= 1 + \sum_{\substack{S \subset V \\ S \neq \emptyset}} \omega_{u, S} \cdot d(A_{\sigma, \pi}, A_{\sigma, \pi}(u, S)) \\ &= 1 + \sum_{\substack{S' \subset V \\ S' \neq \emptyset}} \omega_{u, S'} \cdot d(A_{\sigma, \pi}, A_{\sigma, \pi}(u, S' \cap T_u)) \\ &= 1 + \sum_{\substack{S \subset T_u \\ S \neq \emptyset}} \sum_{S': S' \cap T_u = S} \omega_{u, S'} \cdot d(A_{\sigma, \pi}, A_{\sigma, \pi}(u, S)) \\ &= 1 + \sum_{\substack{S \subset T_u \\ S \neq \emptyset}} q_{u, S} \cdot d(A_{\sigma, \pi}, A_{\sigma, \pi}(u, S)), \end{aligned} \quad (3.3)$$

$$= 1 + \sum_{\substack{S \subset T_u \\ S \neq \emptyset}} q_{u, S} \cdot d(A_{\sigma, \pi}, A_{\sigma, \pi}(u, S)), \quad (3.4)$$

where (3.3) follows from the fact that by definition of  $A_{\sigma, \pi}$ ,  $A_{\sigma, \pi}(u, S)$  depends only on  $u$  and  $S \cap T_u$ , and (3.4) follows from the definition of  $q_{u, S}$ . Since by the definition of  $A_{\sigma, \pi}$ ,  $\sigma(A_{\sigma, \pi}(u, S)) > \sigma(u)$ , the induction hypothesis implies that  $d(A_{\sigma, \pi}, A_{\sigma, \pi}(u, S)) = r_{A_{\sigma, \pi}(u, S)}$ . Furthermore, by the definition of  $A_{\sigma, \pi}$  and  $\pi$ , we have that  $r_{A_{\sigma, \pi}(u, S)} = \min_{v \in S} \{r_v\}$ . Combined with (3.4) and the definition of  $r_u$ , this shows that  $d(A_{\sigma, \pi}, u) = r_u$ , as desired.

We now prove that  $d(A, u) \geq r_u$ . By Lemma 3.4 and the definition of  $V_0$ , we have  $A(u, S) \in S \cap T_u \cap V_0$ . Therefore,

$$d(A, A(u, S)) \geq \min_{v \in S \cap T_u \cap V_0} \{d(A, v)\}. \quad (3.5)$$

By the induction hypothesis, we have that  $d(A, v) \geq r_v$  for every  $v \in T_u \cap V_0$ . This, together with (3.2) and (3.5), implies

$$\begin{aligned}
 d(A, u) &\geq 1 + \sum_{\substack{S \subseteq V \\ S \neq \emptyset}} \omega_{u,S} \cdot d(A, A(u, S)) \\
 &\geq 1 + \sum_{\substack{S \subseteq V \\ S \neq \emptyset}} \omega_{u,S} \cdot \min_{v \in S \cap T_u \cap V_0} \{d(A, v)\} \\
 &\geq 1 + \sum_{\substack{S \subseteq V \\ S \neq \emptyset}} \omega_{u,S} \cdot \min_{v \in S \cap T_u \cap V_0} \{r_v\} \\
 &= 1 + \sum_{\substack{S \subseteq T_u \\ S \neq \emptyset}} \sum_{S': S' \cap T_u = S} \omega_{u,S'} \cdot \min_{v \in S \cap V_0} \{r_v\} \\
 &= 1 + \sum_{\substack{S \subseteq T_u \\ S \neq \emptyset}} q_{u,S} \cdot \min_{v \in S} \{r_v\} \\
 &= r_u,
 \end{aligned} \tag{3.6}$$

where (3.6) follows from the definition of  $q_{u,S}$  and the fact that by property (iii) of  $\sigma$ ,  $S \cap V_0 = S$  for every  $S \subseteq T_u$ . This completes the proof of the induction step.  $\square$

**Proof of Theorem 3.3.** Define the graph  $H$ , the ordering  $\sigma$ , the values  $r_u$ , and the ordering  $\pi$  as above. By Lemma 3.5, we have that  $d(A_{\sigma,\pi}, s) \leq d(A, s)$ . Since  $G(n, \Omega)$  is  $d$ -searchable using  $A$  by assumption, we have that  $d(A, s) \leq d$ . Hence we have  $d(A_{\sigma,\pi}, s) \leq d$ , as desired.  $\square$

Note that in the above proof, the second ordering  $\pi$  was defined in terms of the first ordering  $\sigma$ . Therefore, the condition for the searchability of  $G(n, \Omega)$  can be stated in terms of only one ordering  $\sigma$  as follows:

**Corollary 3.6.**  *$G(n, \Omega)$  is  $d$ -searchable if and only if there is an ordering  $\sigma$  on the nodes for which  $r_s \leq d$ , where  $r$  is defined as in (3.1).*

A second corollary of the above characterization is that the following problem is in NP.

**SEARCHABILITY:** Given a positive integer  $n$ , an  $n \times n$  matrix  $\mathbf{P}$ , two nodes  $s$  and  $t$ , and a positive number  $d$ , decide whether  $G(n, \mathbf{P})$  is  $d$ -searchable.

**Corollary 3.7.** SEARCHABILITY is in NP.

**Proof.** We use the ordering  $\sigma$  as a polynomial-size certificate for membership in SEARCHABILITY. By Corollary 3.6, it is enough to show that  $r_u$  can be computed in polynomial time. We prove this by rewriting (3.1) for nodes  $u$  with  $u \neq t$  and  $q_{u,\emptyset} = 0$ . To do this, fix any such  $u$  and let  $v_1, \dots, v_t$  denote the nodes of  $T_u$  ordered in increasing order of their  $r_v$ 's, i.e.,  $r_{v_1} \leq r_{v_2} \leq \dots \leq r_{v_t}$ . We have

$$\begin{aligned}
 r_u &= 1 + \sum_{\substack{S \subseteq T_u \\ S \neq \emptyset}} \prod_{v \in S} p_{uv} \prod_{v \in T_u \setminus S} (1 - p_{uv}) \cdot \min_{v \in S} \{r_v\} \\
 &= 1 + \sum_{i=1}^t \sum_{\substack{S \subseteq \{1, \dots, t\} \\ \min\{S\} = i}} \prod_{j \in S} p_{uv_j} \prod_{j \in \{1, \dots, t\} \setminus S} (1 - p_{uv_j}) \cdot r_{v_i} \\
 &= 1 + \sum_{i=1}^t r_{v_i} p_{uv_i} \prod_{j=1}^{i-1} (1 - p_{uv_j}) \sum_{S \subseteq \{i+1, \dots, t\}} \prod_{j \in S} p_{uv_j} \prod_{j \in \{i+1, \dots, t\} \setminus S} (1 - p_{uv_j}) \\
 &= 1 + \sum_{i=1}^t r_{v_i} p_{uv_i} \prod_{j=1}^{i-1} (1 - p_{uv_j}).
 \end{aligned}$$

Given this equation, we argue that it is possible to compute  $r_u$  given  $r_v$  for all  $v \in T_u$  in polynomial time. This is obvious in the real-computation model. To see that this is also possible in the standard bit operation model, note that the above equation gives the value of  $r_u$  as a linear combination of  $r_{v_i}$ 's, where each coefficient in this linear combination is a product of at most  $n$  probabilities of the form  $p_{uv_j}$  or  $1 - p_{uv_j}$ . Therefore, if  $\ell_{\max}$  denotes the length of the longest binary representation of any element in the probability matrix  $\mathbf{P}$ , each coefficient in the linear expression of  $r_u$  in terms of the  $r_{v_i}$ 's is of bit-length at most  $n(\ell_{\max} + 1)$ . Using a simple induction on  $u$ , this implies that the size of the binary representation of any  $r_u$  is at most  $n(1 + n(\ell_{\max} + 1))$ , which is polynomial in the size of the input. Therefore, using the above expression, all the  $r_u$ 's can be computed exactly in polynomial time.

Therefore, by Corollary 3.6, membership in SEARCHABILITY can be tested in polynomial time given the certificate  $\sigma$ .  $\square$

#### 4. The Monotonicity Property

Armed with the characterization theorem of the previous section, we can now prove the following natural monotonicity property for searchability.

**Theorem 4.1.** *Let  $\mathbf{P}, \mathbf{P}'$  be two  $n \times n$  probability matrices such that for every  $i$  and  $j$ , we have  $p_{ij} \leq p'_{ij}$ . Fix the source and destination nodes  $s$  and  $t$ . Then, if  $G(n, \mathbf{P})$  is  $d$ -searchable for some  $d$ , so is  $G(n, \mathbf{P}')$ .*

**Proof.** By Corollary 3.6, since  $G(n, \mathbf{P})$  is  $d$ -searchable, there is an ordering  $\sigma$  such that the value  $r_s$  defined using (3.1) is at most  $d$ . To show  $d$ -searchability of  $G(n, \mathbf{P}')$ , we apply the same ordering  $\sigma$ . Let  $\{r'_u\}$  denote the values computed using (3.1), but with  $\mathbf{P}$  replaced by  $\mathbf{P}'$ . Similarly, we define  $q'_{u,S}$ 's. By Corollary 3.6, it suffices to show that  $r'_s \leq d$ . To do this, we prove by induction that for every  $u \in V$ , we have  $r'_u \leq r_u$ . This statement is trivial for  $u = t$ . We assume that it has been proved for every  $v \in V$  with  $\sigma(v) > \sigma(u)$ , and prove it for  $u$ . First, note that if  $q'_{u,\emptyset} > 0$ , we have

$$q_{u,\emptyset} = \prod_{v \in T_u} (1 - p_{uv}) \geq \prod_{v \in T_u} (1 - p'_{uv}) = q'_{u,\emptyset} > 0.$$

Hence,  $r_u = \infty$  and the inequality  $r'_u \leq r_u$  holds. Therefore, we may assume  $q'_{u,\emptyset} = 0$ . Thus, we have

$$\begin{aligned} r'_u &= 1 + \sum_{\substack{S \subseteq T_u \\ S \neq \emptyset}} \prod_{v \in S} p'_{uv} \prod_{v \in T_u \setminus S} (1 - p'_{uv}) \cdot \min_{v \in S} \{r'_v\} \\ &\leq 1 + \sum_{\substack{S \subseteq T_u \\ S \neq \emptyset}} \prod_{v \in S} p'_{uv} \prod_{v \in T_u \setminus S} (1 - p'_{uv}) \cdot \min_{v \in S} \{r_v\}. \end{aligned}$$

Let  $1, 2, \dots, k$  denote the nodes of  $T_u$ , ordered in such a way that  $r_1 \leq r_2 \leq \dots \leq r_k$ . Recall that  $\prod_{v \in S} p'_{uv} \prod_{v \in T_u \setminus S} (1 - p'_{uv})$  is the probability that in  $G(n, \mathbf{P}')$ ,  $\Gamma(u) \cap T_u = S$ . Therefore, we have

$$\begin{aligned} r'_u &\leq 1 + \sum_{\substack{S \subseteq T_u \\ S \neq \emptyset}} \Pr_{G(n, \mathbf{P}')}[\Gamma(u) \cap T_u = S] \cdot \min_{v \in S} \{r_v\} \\ &= 1 + \sum_{i=1}^k r_i \cdot \Pr_{G(n, \mathbf{P}')}[\min\{\Gamma(u) \cap T_u\} = i] \\ &= 1 + \sum_{i=1}^k r_i (\Pr_{G(n, \mathbf{P}')}[\min\{\Gamma(u) \cap T_u\} \leq i] - \Pr_{G(n, \mathbf{P}')}[\min\{\Gamma(u) \cap T_u\} \leq i - 1]) \\ &= 1 + r_k - \sum_{i=1}^{k-1} \Pr_{G(n, \mathbf{P}')}[\min\{\Gamma(u) \cap T_u\} \leq i] (r_{i+1} - r_i). \end{aligned}$$

The coefficient of  $(r_{i+1} - r_i)$  in the above expression is the probability of the event that the set of nodes that have an edge from  $u$  in  $G(n, \mathbf{P}')$  contains at least one of the nodes  $1, \dots, i$ . This event is monotone; therefore the probability of this event under  $G(n, \mathbf{P})$  is less than or equal to the probability under  $G(n, \mathbf{P}')$ . Therefore,

$$\begin{aligned} r'_u &\leq 1 + r_k - \sum_{i=1}^{k-1} \Pr_{G(n, \mathbf{P})}[\min\{\Gamma(u) \cap T_u\} \leq i](r_{i+1} - r_i) \\ &= 1 + \sum_{i=1}^k r_i \cdot \Pr_{G(n, \mathbf{P})}[\min\{\Gamma(u) \cap T_u\} = i] \\ &\leq r_u, \end{aligned}$$

where the last inequality follows from the definition of  $r_u$  (and holds with equality unless  $q_{u, \emptyset} > 0$ ). This completes the proof of the induction step.  $\square$

We want to point out that the monotonicity result may seem surprising especially to those who are familiar with Kleinberg’s result on the searchability of long-range percolation graphs [Kleinberg 00]: in that graph model, there is a critical value for a parameter  $r$  such that the graph is searchable in polylogarithmically many steps only if  $r$  takes on this value; either larger or smaller values for  $r$  would render graphs with polynomial-time decentralized routing time. Our result is not contradictory to Kleinberg’s result: in his model, changing the parameter  $r$  affects the probabilities of many edges simultaneously; in particular, those probabilities sum up to 1, so either increasing or decreasing the value of  $r$  will cause the probability of certain edges to decrease. Therefore the condition for our monotonicity result does not hold in his model.

Note that simple as the statement of Theorem 4.1 sounds, we do not know whether a similar statement holds for randomized memoryless algorithms. On the other hand, we have proved the monotonicity property for randomized algorithms with memory; the proof can be found in [Mahdian and Xu 07].

## 5. Conclusions and Open Problems

In this paper, we defined a general class of random graphs, and gave a simple characterization of random graphs in this class that are searchable using decentralized deterministic memoryless algorithms. Our framework includes many of the previously studied small-world networks. Two important corollaries of our characterization are the monotonicity of the searchability property, and mem-

bership of the problem of testing searchability of a given graph in the complexity class NP.

Our framework and results lead to many interesting open questions. First, it would be interesting to settle the complexity of SEARCHABILITY. We proved that this problem belongs to NP. However, we do not know whether this problem can be solved in polynomial time, or whether it is NP-complete.

Characterizing searchability with respect to more general classes of decentralized search algorithms is another important open question. The class of search algorithms we considered in this paper can be generalized in three ways: allowing the algorithm to use randomization, allowing the algorithm to have memory, and allowing the algorithm to fail (i.e., not find a path to  $t$ ) with a small probability  $\epsilon$ . The following simple example shows that any of these generalizations (or in fact, even allowing the algorithm to store one bit of memory) can drastically increase the power of the algorithm.

**Example 5.1.** Consider a graph, consisting of nodes  $s = u_0, u_1, \dots, u_{n-2} = t$ , and  $v$ . For every  $i = 0, \dots, n - 1$ , there is an edge from  $u_i$  to  $u_{i+1}$  with probability one. Also, there is an edge from  $s$  to  $v$  and one from  $v$  to  $s$  with probability one. Finally, there is an edge from  $v$  to  $t$  with probability  $1 - 1/n$ . A deterministic memoryless algorithm on this graph cannot ever visit  $v$ , since if it does and the direct edge from  $v$  to  $t$  is not present, it has to go back to  $s$  and then it will fall into a loop, since it has to go back to  $v$ . Therefore, any such algorithm reaches  $t$  in expected  $n - 2$  steps. However, if the algorithm is allowed to have one bit of memory, it can first go to  $v$ , and if the edge to  $t$  is not present, go back to  $s$ , remembering that it has already visited  $v$ . This achieves an expected path length of  $(1 - 1/n)2 + \frac{1}{n} \cdot n < 3$ . Similarly, if the algorithm is allowed to use randomization, it can flip a coin at  $s$ , and choose to go to either  $v$  or  $u_1$  with probabilities  $1 - 1/n$  and  $1/n$ , respectively. If it goes to  $v$  and the edge to  $t$  is not present, it returns to  $s$  and flips the coin again. It is not hard to prove that the expected number of steps that this algorithm takes to reach  $t$  is at most 6.

The above example suggests that a characterization as simple as the one in Theorem 3.3 is probably impossible for algorithms that have memory, use randomization, or are allowed to fail. But at least it would be interesting to determine whether these problems belong to NP.

Finally, we note that despite the fact that it seems intuitive that the searchability property is monotone, we know how to prove this property only for deterministic memoryless algorithms and for algorithms with memory. Most importantly, proving this property for randomized memoryless algorithms is an intriguing open question.

**Acknowledgments.** The first author’s work was performed in part during a visit to Yahoo! Research. The authors thank Amin Saberi for many useful discussions.

## References

- [Aiello et al. 01] W. Aiello, F. Chung, and L. Lu. “A Random Graph Model for Power Law Graphs.” *Experimental Mathematics* 10:1 (2001), 53–66.
- [Barabási and Albert 99] A.-L. Barabási and R. Albert. “Emergence of Scaling in Random Networks.” *Science* 286:15 (1999), 509–512.
- [Bollobás and Chung 88] B. Bollobás and F. R. K. Chung. “The Diameter of a Cycle Plus a Random Matching.” *SIAM Journal on Discrete Mathematics* 1:3 (1988), 328–333.
- [Chung and Lu 03] F. Chung and L. Lu. “The Average Distance in a Random Graph with Given Expected Degrees.” *Internet Mathematics* 1:1 (2003), 91–114.
- [Duchon et al. 05] P. Duchon, N. Hanusse, E. Lebhar, and N. Schabanel. “Could Any Graph Be Turned into a Small World?” *Theoretical Computer Science* 355:1 (2006), 96–103.
- [Erdős and Rényi 59] P. Erdős and A. Rényi. “On Random Graphs I.” *Publications Mathematics Debrecen* 6 (1959), 290–297.
- [Fraigniaud 05] P. Fraigniaud. “Greedy Routing in Tree-Decomposed Graphs.” In *Algorithms—ESA 2005: 13th Annual European Symposium, Palma de Mallorca, Spain, October 3–6, 2005, Proceedings*, Lecture Notes in Computer Science 3669, pp. 791–802. Berlin: Springer, 2005.
- [Kleinberg 00] J. Kleinberg. “The Small-World Phenomenon: An Algorithmic Perspective.” In *Proceedings of the Thirty-Second Annual ACM Symposium on Theory of Computing*, pp. 163–170. New York: ACM Press, 2000.
- [Kleinberg 01] J. Kleinberg. “Small-World Phenomena and the Dynamics of Information.” *Advances in Neural Information Processing Systems* 14 (2001), 431–438.
- [Kumar et al. 06] R. Kumar, D. Liben-Nowell, and A. Tomkins. “Navigating Low-Dimensional and Hierarchical Population Networks.” In *Algorithms—ESA 2006: 14th Annual European Symposium, Zurich, Switzerland, September 11–13, 2006, Proceedings*, Lecture Notes in Computer Science 4168, pp. 480–491. Berlin: Springer, 2006.
- [Leskovec et al. 05] J. Leskovec, D. Chakrabarti, J. Kleinberg, and C. Faloutsos. “Realistic, Mathematically Tractable Graph Generation and Evolution, Using Kronecker Multiplication.” In *Knowledge Discovery in Databases: PKDD 2005 9th European Conference on Principles and Practice of Knowledge Discovery in Databases, Porto, Portugal, October 3–7, 2005, Proceedings*, Lecture Notes in Computer Science 3721, pp. 133–145. Berlin: Springer, 2005.
- [Liben-Nowell et al. 05] D. Liben-Nowell, J. Novak, R. Kumar, P. Raghavan, and A. Tomkins. “Geographic Routing in Social Networks.” *Proceedings of the National Academy of Sciences* 102:33 (2005), 11623–11628.

- [Mahdian and Xu 07] M. Mahdian and Y. Xu. “Stochastic Kronecker Graphs.” In *Algorithms and Models for the Web-Graph: 5th International Workshop, WAW 2007, San Diego, CA, USA, December 11–12, 2007, Proceedings*, Lecture Notes in Computer Science 4863, pp. 179–186. Berlin: Springer, 2007.
- [Milgram 67] S. Milgram. “The Small World Problem.” *Psychology Today* 1 (1967), 61–67.
- [Slivkins 05] A. Slivkins. “Distance Estimation and Object Location via Rings of Neighbors.” In *Proceedings of the Twenty-Fourth Annual ACM Symposium on Principles of Distributed Computing*, pp. 41–50. New York: ACM Press, 2005.
- [Watts and Strogatz 98] D. J. Watts and S. H. Strogatz. “Collective Dynamics of ‘Small-World’ Networks.” *Nature* 393 (1998), 440–442.

---

Esteban Arcaute, Institute for Computational and Mathematical Engineering, Stanford University, Durand Building 496 Lomita Mall, Stanford, CA 94305-4042  
(arcaute@stanfordalumni.org)

Ning Chen, Division of Mathematical Sciences, Nanyang Technological University, 21 Nanyang Link, Singapore 637371 (ningc@ntu.edu.sg)

Ravi Kumar, Yahoo! Research, 701 First Ave., Sunnyvale, CA 94089  
(ravikumar@yahoo-inc.com)

David Liben-Nowell, Department of Computer Science, Carleton College, 1 North College Street, Northfield, MN 55057 (dlibenno@carleton.edu)

Mohammad Mahdian, Yahoo! Research, 701 First Ave., Sunnyvale, CA 94089  
(mahdian@yahoo-inc.com)

Hamid Nazerzadeh, Department of Management Science and Engineering, Stanford University, Terman Engineering Center 380 Panama Way, Stanford, CA 94305-4026  
(hamidnz@stanfordalumni.org)

Ying Xu, Department of Computer Science, Stanford University, 353 Serra Mall, Stanford, CA 94305-9025 (xuying@stanford.edu)

Received February 14, 2008; accepted May 6, 2008.