Deterministic Decentralized Search in Random Graphs

Esteban Arcaute\textsuperscript{1} \quad Ning Chen\textsuperscript{2} \quad Ravi Kumar\textsuperscript{3} \\
David Liben-Nowell\textsuperscript{4} \quad Mohammad Mahdian\textsuperscript{3} \\
Hamid Nazerzadeh\textsuperscript{1} \quad Ying Xu\textsuperscript{1}

\textsuperscript{1}Stanford University \quad \textsuperscript{2}University of Washington \\
\textsuperscript{3}Yahoo! Research \quad \textsuperscript{4}Carleton College

The 5th International Workshop on Algorithms and Models for the Web-Graph WAW’07
Outline

Motivation

Model

Results

Open Problems
Small-World
Algorithmic Small-World

Milgram experiment: from Omaha to Boston
  six degrees of separation in Social Networks

Kleinberg’s observation: short paths exist and can be found
  algorithmic problem: decentralized search

Implicit observation: decentralized search using simple algorithms:
  - Milgram: professional and geographical information
  - Kleinberg: graph distance in some structure
Ordering Algorithms

We define a class of simple algorithms:

Definition (Ordering Algorithm)
Let $\sigma, \pi$ be two orderings of $V$. Let $u \in V$, and $\Gamma(u)$ its neighborhood. $A_{\sigma,\pi}(u, \Gamma(u))$ is such that:

1. out of the nodes $v \in \{u\} \cup \Gamma(u)$ such that $\sigma(v) \geq \sigma(u)$,
2. select the maximum element with respect to $\pi$. 
In words: \( A_{\sigma,\pi} \) never goes backwards according to the ordering \( \sigma \), and, subject to this restriction, makes the maximum possible progress according to \( \pi \).
Goals

- Provide a flexible definition of searchability.
- Prove that if a graph is searchable, then users can search with an ordering algorithm.
- Explore how one can find such an algorithm.
- Finally, prove a monotonicity result: if more edges are added, searchability is preserved.
Random Graph Model

Algorithms

Let $P$ be a $n$ by $n$ matrix with entries in $[0, 1]$. Independently at random, we have

$$\text{Prob}[(i, j) \in E] = P_{ij}$$

generalization of the directed variant of Erdős–Rényi graphs.

Local search algorithms: deterministic and memoryless: given

- source destination pair $(s, t)$,
- current vertex and its out-neighborhood
gives next vertex to visit deterministically.
Searchability

Definition

Let $d > 0$ be given.

Definition (Searchability)

A random graph $G(n, P)$ is $d$-searchable if, for all source-destination pairs $(s, t)$, there exists a local search algorithm $A$ such that,

1. $A$ is deterministic and memoryless;
2. $A$ finds a $s-t$ path with probability one; and
3. the expected length of the path is at most $d$.

The probability space is over the set of possible outcomes for the graph.
Characterization of $d$-Searchable Random Graphs

Theorem (Algorithmic Characterization)

$G(n, P)$ is $d$-searchable if and only if it is $d$-searchable using an ordering algorithm $A_{\sigma, \pi}$.

Interpretation: in a $d$-searchable social network, users can navigate using an ordering algorithm.

$\hookrightarrow$ searchability definition compatible with “simple algorithm” observation.
Proof Sketch
Finding $\sigma$

Let $A$ be such that $G(n, P)$ is $d$-searchable using $A$.

Let $H$ be the union over all $s - t$ paths found by $A$ with positive probability.

$H$ can be interpreted as a directed graph over $V$.

To define $\sigma$ from $H$, we

- prove that $H$ is a DAG;
- prove that $t$ is the only node with out-degree zero;
- $\sigma$ is a topological ordering of $H$ with $t$ its largest element.
Proof Sketch

Finding $\pi$

Given $\sigma$ and $\mathcal{P}$ we define numbers $r_u$ for every $u \in V$ recursively as follows:

1. Initialize all $r_u = \infty$; then
2. Start from $t$, set $r_t = 0$;
3. Go to the next highest vertex $u$ with respect to $\sigma$,
   - Given its neighborhood $\Gamma(u)$, select $v$ with minimal $r_v$.
   - Given this choice, let $r_u$ be the expected distance from $u$ to $t$
4. Continue until $u = s$.

We define $\pi$ as follows: let $\pi(u) > \pi(v)$ if $r_u < r_v$.

Thus, out of all possible local search algorithms that route according to $\sigma$, we select the best one.
To complete the proof, we prove that

- The expected distance from any \( u \) such that \( \sigma(u) \geq \sigma(s) \) using algorithm \( A_{\sigma,\pi} \) is exactly \( r_u \); and
- \( r_u \) is a lower bound on the expected distance using algorithm \( A \).

The previous proof holds in a more general random graph model. We only need the out neighborhoods of distinct nodes to be independent.

\( \rightarrow \) our results also hold for long-range percolation graphs and other useful models.
Important Corollary

Theorem (Functional Characterization)

\[ G(n, P) \text{ is } d\text{-searchable if and only if there is an ordering } \sigma \text{ on the nodes for which } r_s \leq d, \text{ where } r \text{ is defined as in Equation (1)}. \]

For a given \( \sigma \) and \( P \), we can efficiently calculate the numbers \( r_u \).

\( \leadsto \) we reduced searchability to a functional \textit{tractable} property of node orderings.
Monotonicity

Assume that $\mathcal{P}_1 \leq \mathcal{P}_2$, then we have that

**Theorem**

If $G(n, \mathcal{P}_1)$ is $d$-searchable, then $G(n, \mathcal{P}_2)$ is $d$-searchable. Further, the ordering algorithm defined for $G(n, \mathcal{P}_1)$ can also be used for $G(n, \mathcal{P}_2)$.

The proof follows from a mild modification of that of the previous theorem.

**Interpretation:** adding more links cannot “confuse” members of a social networks
Note on Monotonicity

Assume that we are given $P_1$, $d$ and $A$ such that $G(n, P_1)$ is $d$-searchable using $A$. Further, we have $P_2 \geq P_1$.

We know that $G(n, P_2)$ is $d$-searchable using $A$.

Remember: $d$ is given, and thus is not to be confused with a function of the random graph model (such as the expected diameter).
Open Problems
Research Directions

- Characterization when positive probability of failure allowed
- Results for other types of algorithms:
  - Monotonicity result available for randomized algorithms with memory
  - Differences between memoryless and algorithms with memory - tradeoffs
- Find minimal searchable graphs
Thank you
More Details on Finding $\pi$

Given $\sigma$ and $\mathcal{P}$ we define numbers $r_u$ for every $u \in V$ recursively as follows: $r_t = 0$, and for every $u \neq t$,

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

(1)

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

$$q_{u,S} := \left( \prod_{v \in S} p_{uv} \right) \left( \prod_{v \in T_u \setminus S} (1 - p_{uv}) \right)$$

to denote the probability that the subset of nodes of $T_u$ that are out-neighbors of $u$ is precisely $S$.

We define $\pi$ as follows: let $\pi(u) > \pi(v)$ if $r_u < r_v$. 