Deterministic Decentralized Search in Random Graphs

Esteban Arcaute[1] Ning Chen[2] Ravi Kumar[3]
David Liben-Nowell[4] Mohammad Mahdian[3]
Hamid Nazerzadeh[1] Ying Xu[1]

[1]Stanford University [2]University of Washington

[3]Yahoo! Research [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
  $\hookrightarrow$ six degrees of separation in Social Networks

Kleinberg's observation: short paths exist *and* can be found
  $\hookrightarrow$ algorithmic problem: decentralized search

Implicit observation: decentralized search using simple algorithms:

- Milgram: professional and geographical information
- Kleinberg: graph distance in some structure

Ordering Algorithms

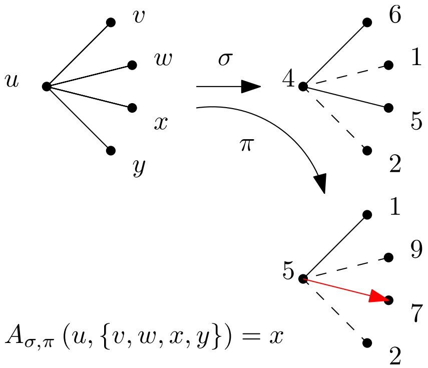
We define a class of simple algorithms:

Definition (Ordering Algorithm)

Let σ, π 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 π.

Illustration



$$A_{\sigma,\pi}\left(u,\{v,w,x,y\}\right)=x$$

In words: $A_{\sigma,\pi}$ never goes backwards according to the ordering σ, and, subject to this restriction, makes the maximum possible progress according to π.

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

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

$\hookrightarrow$ 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 $G(n, \mathcal{P})$ 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)

$G(n, \mathcal{P})$ 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 σ

Let A be such that $G(n, \mathcal{P})$ 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 σ from H, we

- prove that H is a DAG;
- prove that t is the only node with out-degree zero;
- σ is a topological ordering of H with t its largest element.

Proof Sketch
Finding π

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

- initialize all $r_u = \infty$; then
- start from t, set $r_t = 0$;
- go to the next highest vertex u with respect to σ,
  - given its neighborhood $\Gamma(u)$, select v with minimal r_v.
  - given this choices, let r_u be the expected distance from u to t
- continue until $u = s$.

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

Thus, out of all possible local search algorithms that route according to σ, we select the best one.

Completing the Proof
Important Remark

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.

$\hookrightarrow$ our results also hold for long-range percolation graphs and other useful models.

Important Corollary

Theorem (Functional Characterization)

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

For a given σ and $\mathcal{P}$, we can efficiently calculate the numbers r_u.

$\hookrightarrow$ we reduced searchability to a functional *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 $\mathcal{P}_1$, d and A such that $G(n, \mathcal{P}_1)$ is d-searchable using A. Further, we have $\mathcal{P}_2 \geq \mathcal{P}_1$.

We know that $G(n, \mathcal{P}_2)$ 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

Comments? Questions?

Thank you

More Details on Finding π

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

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

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