Low-cost search in scale-free networks

Jieun Jeong* and Piotr Berman†
Department of Computer Science and Engineering, The Pennsylvania State University, University Park, Pennsylvania 16802, USA
(Received 22 June 2006; revised manuscript received 19 October 2006; published 5 March 2007)

We study local search algorithms for networks with heterogeneous edge weights, testing them on scale-free and Erdös-Rényi networks. We assume that the location of the destination node is discovered when it is two edges away, and that the search cost is additive. It was previously shown that a search strategy preferring high-degree nodes reduces the average search cost over a simple random walk. In the prior work, for the case when the edge costs are randomly distributed, a different preference was investigated [high local betweenness centrality (LBC)], and was found to be superior to high-degree preference in scale-free networks, with the exception for the most sparse ones. We have found several preference criteria that are simpler and which, in all networks we tested, yield a lower cost than other criteria including high-degree, high-LBC, and low-edge cost.

DOI: 10.1103/PhysRevE.75.036104 PACS number(s): 89.75.Fb, 89.75.Hc, 02.10.Ox, 89.70.+c

I. INTRODUCTION

A. Motivation: Peer-to-peer networks

Spreading messages in random graphs has been investigated for several reasons. One is to model interactions in large social networks. Another is to actually optimize message delivery in existing communication networks that were formed without any "central planning."

Adamic *et al.* [1] cite peer-to-peer networks created by a set of participants that collectively store a set of files. As they describe, such a system either has a central server that contains information about every file in the system, like NAPSTER, or is decentralized, like GNUTELLA or FREENET. In the latter case each participant is a node in a graph, and communicates only with the participants that are neighbors in that graph. When a participant needs a file, it sends a request—without knowing which other participant actually possesses the file. Therefore to fulfill a request a *search* has to be executed.

Kleinberg [2] discussed search strategies in networks that had a known part—a grid—and unknown random connections (Watts-Strogatz graphs [3]). In our context a similar approach cannot work because the sender of the request has no information about the search target.

Without the knowledge of the target, it is hard to expect a message to reach its destination without visiting, on the average, half of the nodes (participants). This would make the network intolerably inefficient. We can do better if the location of each file is known to many nodes in the network. Adamic *et al.* [1] assumed that this information is given to the direct neighbors and their neighbors (second neighbors). This is reasonable because when the set of files is highly dynamic, updating information about each file itself creates considerable traffic; the above restriction limits the traffic of these updates and makes it easy to annotate each file name with the edge to follow on a shortest route to this file.

B. Graphs modeling the networks

In this paper, the *degree* of a graph node equals the number of its neighbors. It was observed [4] that many networks that are formed spontaneously by evolution or by social interactions follow a *power law*: the proportion of nodes of degree d is proportional to $d^{-\tau}$, where τ is the exponent coefficient of a particular power law. As in Ref. [4], we say that graphs generated according to this law are *scale-free*.

To avoid extremes of the variability in the graph structure, we do not generate nodes with degrees for which there is high chance that they do not appear at all. More precisely, we impose the following condition on degree d of a node: $d^{-\tau} \geqslant n^{-1}$ (this is *Aiello cutoff* introduced in Ref. [5]).

The measurement of GNUTELLA networks [6] showed that they exhibit power-law distribution of degrees, so it is appropriate to test search methods for these networks in scale-free graphs.

We will test our search methods on randomly generated scale-free graphs, and also in *Erdös-Rényi graphs* (a.k.a. Poisson random graphs). The latter allows us to check if the efficacy of search methods relies on peculiar properties of scale-free graphs.

Erdös-Rényi graphs are random graphs in which the degree of nodes has Poisson distribution. One can obtain such a graph by placing an edge connecting two nodes with some constant probability p using a separate independent random trial for each pair of nodes.

C. Local rules and cost models

The search process proposed by Adamic *et al.* [1] is a walk through the graph of the network traversed by the message that contains the original request. Because the request follows a single route it is easy to terminate the process when the destination is found.

One can fully describe such a search method as follows: each node orders the list of its neighbors according to some rule. More specifically, after this ordering values of some function applied to neighboring nodes should be nondecreasing (see the examples in Sec. II). A request message is sent and until it reaches its destination it is forwarded to a neighbor of the current node according to the first applicable rule:

*Electronic address: jijeong@cse.psu.edu
†Electronic address: berman@cse.psu.edu

 ©2007 The American Physical Society

(a) the current node is a second (or first) neighbor of the destination; according to our model, this node knows the destination, which means that it has a table of file identifiers and their locations with an entry for every file stored locally and at the first and second neighbors; thus the message can be forwarded to a first neighbor of the destination (or the destination itself);

(b) some neighbor of the current node has not been visited during the current search: the message is forwarded to the first such node on the neighbor list;

(c) the message is forwarded to a random neighbor, chosen with the exclusion of the neighbor that had this message immediately before.

Adamic *et al.* [1] assumed that the cost of the search process equals the number of edges that it traverses. While this model may be adequate, there may also be important reasons to charge different costs for the use of different edges. For example, different nodes may experience different congestion, and they may have different capacity for handling traffic.

For this reason, we can give each node a cost coefficient and the cost of a search will be the sum of the costs of the nodes that were visited. If the node costs are unpredictable, they can be modeled with a random distribution. Thadamakalla *et al.* [7] made the assumption that edge costs, rather than node costs, are random. We used this model so we could directly compare our search heuristic, but the methodology is essentially the same whether cost is assigned to edges or to nodes.

D. Previous work and the present contribution

The case when each edge has the same cost was investigated by Adamic *et al.* [1]. They considered local search rules: random (random ordering of the list of neighbors) and high degree (after the ordering, degrees of the neighbors on the list are nondecreasing). Rather unsurprisingly, the high degree rule was better; the nontrivial contribution of Adamic *et al.* [1] was to provide the probabilistic analysis that predicts the cost of both delivery methods.

The case when the edge costs are randomly distributed was investigated by Thadamakalla *et al.* [7]. They introduced a new rule, high *local betweenness centrality* or LBC, and verified that in scale-free graphs, especially those that have low τ, it is superior to both the high-degree criterion and the low edge-cost criterion. In Erdös-Rényigraphs low edge cost was superior.

The LBC rule is somewhat complex and we have compared it with variants of a very simple rule: high degree or cost, which means that when we order the neighbors of some node u, the value of our function applied to v is d_v/c_{uv}, where d_v is the degree of v and c_{uv} is the cost of the edge $\{u,v\}$.

Simulations have shown that this rule performs better than high LBC, even in graphs in which the LBC rule is inferior to the simple high-degree rule. In Sec. III we mention other rules that we tested in our simulations; they were somewhat more complicated and, dependent on the class of graphs, they were either inferior to high degree or cost or only marginally better.

E. The rest of this paper

This paper is organized as follows. In Sec. II we describe the criteria we investigate in this paper. We explain the reasons for selecting these criteria in Sec. III. In the same section we present results of our tests in the form of four figures. We finish with our conclusions in Sec. IV.

II. LOCAL SEARCH CRITERIA

We will show results for six functions that may serve as local search criteria. When the search process gives us a choice of more than one edge to follow, we select one that maximizes such a function, breaking draws arbitrarily: (i) random (equal chance of moving along any edge); (ii) degree; (iii) LBC (defined below); (iv) 1/edge cost (the preference for low values of the edge cost); (v) degree/edge cost; (vi) degree^2/edge cost.

The first two functions used for local search criteria were investigated by Adamic *et al.* [1]; the next two were tested by Thadamakalla *et al.* [7], and the last two are tested for the first time in this paper.

LBC function $L(v)$ (local betweenness centrality) is defined in Ref. [7] as follows. When we order neighbors of node u, we consider $G_2(u)$, the subgraph of the network induced by the set that consists of u, its neighbors, and second neighbors. When v, w, x are nodes of $G_2(u)$, we define $\psi(v,w,x)$ as the number of shortest paths from w to x [within $G_2(u)$] that contain node v, and

$$L(v) = \sum_{w \neq v \neq x} \psi(v,w,x).$$

This definition was inspired in part by the notion of betweenness centrality introduced by Freeman [8] in the context of social networks. The definition in Ref. [7] is a bit more complicated, but we assumed that in $G_2(u)$ there exists only one shortest path from w to x—this assumption is valid when edges have a continuous cost distribution, so the costs of any two paths are never equal.

As we will see in the next section, local betweenness centrality is a rather poor criterion when compared with simpler ratio-based criteria.

III. RESULTS

One can model the process of the search through a random graph as the process of generating the edges that we consider as possible steps of the process. We decide with an appropriate random distribution the degree of every node. When we select a neighbor at random, we select it with probability proportional to its "remaining" degree (the number of neighbors not selected yet), the "random edge" distribution (in a scale-free graph, in "random node" distribution the degree d has probability proportional to $d^{-\tau}$, and in the random edge distribution the degree d has probability proportional to $d^{-\tau+1}$).

The target node t at random builds its set of neighbors—first, it selects the number of neighbors according to the random node distribution, and then it selects the neighbors

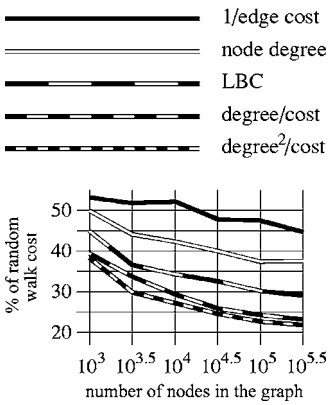

FIG. 1. Performance of various search methods as a function of graph size. The edges were generated according to power law with $\tau=2.2$ and their costs were generated using a uniform distribution in the interval (0,2). Costs are the average costs divided by the average cost of a random walk (which was within 4% from $0.486N^{0.68}$). In all figures we will be using the same kinds of lines to indicate results for various search methods.

themselves, each according to the random edge distribution. In particular, this determines the number a of the "remaining neighbors" of the neighbors of t (each of them has one neighbor selected already, namely, t).

The delivery process performs a walk. In a step, it first randomly selects the neighbors of the current node (that were not selected yet), and then selects the edge (and thus the next node) according to the criterion that is used. Recall that the degrees were selected already, and the new neighbors are selected according to "edge distribution." Each selection of a new neighbor is a Bernoulli trial, and the success happens if the new neighbor is also a neigbor of t. After a success, u is the second neighbor of the target t, which means that the delivery will be completed in two steps.

Each selection has the same chance of success, namely, the ratio $a/|E|$ where E is the set of all edges (there is some simplification here, as we have already eliminated some possibilities, but on the average we inspect a rather small fraction of edges), so the average number of selections is $|E|/a$. The question is how much we have to pay for these selections.

In a random walk, we traverse an edge with an average cost of 1, and if we reach a node of degree d, this allows us to perform $d-1$ selections, so their cost is $1/(d-1)$. The probability that we reach a node of degree d is proportional to $d^{-\tau+1}$, and thus probability for a selection to be made with cost $1/(d-1)$ is $d^{-\tau+1}(d-1)$.

If we select an edge to the node of the highest degree, we obviously skew the distribution of the degree of visited node toward nodes of high degree. Not only we do it in this particular step, but in the next step we will have more neighbors than (random walk's) average, so the highest degree from the larger set will be on the average higher than from a smaller set.

If the edges have random costs, then when we traverse to u of degree d using an edge of cost c we make selections that cost $c/(d-1)$. A greedy choice would be to minimize the cost of the selections made in this step.

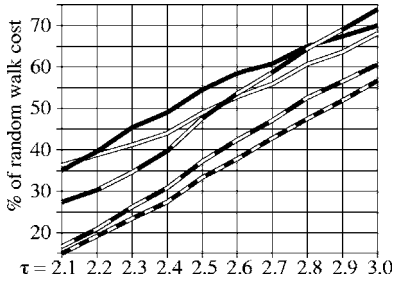

FIG. 2. Performance of various search methods as a function of τ. The results come from simulations in graphs with $N=10^{4.5}$ nodes and with Poisson distribution of edge costs that was normalized to have expectation and variance equal to 1. Costs are the average costs divided by the average cost of a random walk [which was within 2% from $0.099N(\tau-1.78)^2$].

However, there may be a conflict between the low cost in the current step and the low costs in the subsequent steps. The higher the node degree, the more choice we have, and thus the lower (on the average) cost of the selections according to the best choice. On the other hand, low-degree neighbors are more numerous than high-degree neighbors so we have a chance that one of them will be a "winner," in which case we make some very cheap selections and in the next step, due to dearth of choice, we make a much more expensive selection. The exact trade off depends on the distributions of node degrees and of edge costs.

We can improve the odds of walking through the nodes of high degree while still paying attention to the edge costs as follows: rather than selecting a node that maximizes d/c or $(d-1)/c$, select a node that maximizes d^2/c (or minimizes c/d^2). For example, if a node of degree 3 competes with a node of degree 6, and the node of degree 3 can be reached with an edge of cost 1, it may win if the edge to the node of degree 3 is more expensive than 2 (criterion d/c), or more expensive than 2.5 [criterion $(d-1)/c$], or more expensive than 4 (criterion d^2/c).

We tested the conjecture that ratio criteria improve the search costs in computer simulations. We checked the impact of the following aspects of the situation: graph size (see Fig. 1), the distribution of node degrees (see Fig. 2), and the distribution of edge costs (see Fig. 3). The latter had the

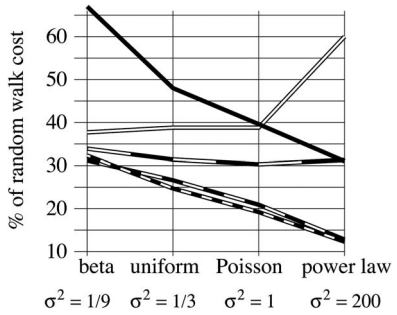

FIG. 3. Performance of various search methods as a function of edge-cost distribution. The results come from simulations in graphs with $N=10^{4.5}$. Costs are the average costs divided by the average cost of a random walk (which was $0.071N$). All distributions have mean 1 and variances as indicated.

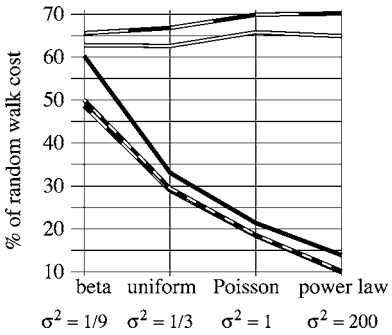

FIG. 4. Performance of various search methods as a function of edge-cost distribution for Erdös-Rényi graphs. The results come from simulations in graphs with the same sizes, edge-cost distributions, and average node degree as in Fig. 3 (average node degree is 6.29). Costs are the average costs divided by the average cost of a random walk (which was $0.031N$). The performance of degree/edge cost and degree^2/edge cost are barely distinguishable, with the latter being about 2% worse.

largest impact on the relative efficacy of various search criteria.

We repeated the test with different edge-cost distribution for Erdös-Rényigraphs (see Fig. 4). Because of a small variability of node degrees, the criterion of low edge cost performs very well in those graphs, but even so the ratio criteria performed somewhat better.

IV. CONCLUSIONS

The data shows that criteria based on ratios of a function of the degree with the edge cost are performing better than other criteria in all random networks we used for testing. The types of networks we have used in those tests include all types tested by Adamic *et al.* [1] and Thadamakalla *et al.* [7].

This shows that one can expect more interesting results on the search processes in random networks. One should stress that the model investigated here is quite arbitrary from the perspective of network design. A standard approach is to "alter the reality" to make it more suitable for routing or searching algorithms.

Under the assumptions of GNUTELLA-type networks we cannot change the network that is created as a result of social interactions. Neither can we impose a centralized management. Nevertheless, we can alter the arbitrary assumption that the knowledge of the target identity (file location) is limited to the nodes at most two edges apart. It is more realistic to assume that the dissemination of the information about target location has a cost, for example, the number of edges (and thus, the number of messages) used to disseminate the information about the target.

According to our sketch of the analysis, the time or cost of finding a target t with a_t second neighbors is proportional to a_t^{-1}. This means that we can improve the average if for the lowest values of a_t we disseminate the information a bit more. In our tests, the least node degree was 2 and so was the least possible number of the second neighbors. Suppose that information about a node t is annotated with a_t; then a node with a second neighbor t that has $a_t=2$ can forward the information about t to its highest degree neighbor, say s, while s deletes the information about its second neighbor, say u, with the maximal a_u. Then the contribution of these two nodes to the average cost decreases from $a_t^{-1}+a_u^{-1}$ to $(a_t+1)^{-1}+(a_u-1)^{-1}$.

After that change, each node has the same size of the table with information about "distant neighbors" and the same number of messages is used to disseminate the information about the files, and yet we have somewhat smaller average search time. One should be able to develop a strategy that would perform similar changes with some optimized frequency.

ACKNOWLEDGMENTS

We would like to thank Dr. Reka Albert who made complex networks simple and interesting, and who introduced the first author to the problem, as well as Dr. Marek Cieplak and Jinyoul Park for their valuable comments.

[1] L. A. Adamic, R. M. Lukose, A. R. Puniyani, and B. A. Huberman, Phys. Rev. E **64**, 046135 (2001).
[2] J. Kleinberg, *The Small World Phenomenon: An Algorithmic Perspective*, in Proceedings of the 32nd SToC (2000), pp. 163–170.
[3] D. Watts and S. Strogatz, Nature (London) **393**, 440 (1998).
[4] A-L. Barabasi and R. Albert, Science **286**, 509 (1999).
[5] W. Aiello, F. Chung, and L. Lu, in *A Random Graph Model for Massive Graphs*, Proceedings of the 32nd SToC (2000), pp. 171–180.
[6] Clip2 Company, GNUTELLA, http://www.clip2.com/gnutella
[7] H. P. Thadakamalla, R. Albert, and S. R. T. Kumara, Phys. Rev. E **72**, 066128 (2005).
[8] L. C. Freeman, Soc. Networks **1**, 215 (1979).