Optimal transport on complex networks

Bogdan Danila,[1,*] Yong Yu,[1] John A. Marsh,[2] and Kevin E. Bassler[1,†]
[1]*Department of Physics, The University of Houston, Houston, Texas 77004, USA*
[2]*SI International, Beeches Professional Campus, Rome, New York 13440, USA*
(Received 11 July 2006; published 6 October 2006)

We present a heuristic algorithm for the optimization of transport on complex networks. Previously proposed network transport optimization algorithms aim at avoiding or reducing link overload. Our algorithm balances traffic on a network by minimizing the maximum node betweenness with as little path lengthening as possible, thus being useful in cases when networks are jamming due to node congestion. By using the resulting routing, a network can sustain significantly higher traffic without jamming than in the case of shortest path routing.

DOI: 10.1103/PhysRevE.74.046106 PACS number(s): 89.75.Hc, 89.20.Hh, 89.75.Da

Transport optimization is a problem encountered in connection with many different types of complex networks. Examples include biological networks like the genetic regulatory system, social acquaintance networks, and a variety of natural and human-made transport and communication networks. Moreover, the ever-increasing amount of information and goods transported on complex human-made networks raises a question about their ultimate transport capacity. The transport capacity of a network is limited by two factors: link capacity and node processing power (bandwidth and, respectively, router latency in the case of communication networks). In this paper we present an algorithm for transport optimization in the case of communication networks. However, the scope of application for our algorithm is not limited to this problem.

Generally, communication network routing is based on the idea of using the shortest paths (or the best available approximation of the shortest paths) between any two nodes of the network [1,2]. The length of a path is computed as the sum of the weights assigned to the links that form the path. In the case of the Internet, link weights are typically assigned manually by operators according to simple rules based on experience [1]. Recently, a series of heuristic algorithms have also been proposed for network traffic optimization [1–5]. These rules and algorithms are aimed at avoiding or reducing link overload by a judicious link weight assignment. The cost of each link is assessed as a monotonically increasing function of the ratio between traffic and capacity and then the weights are adjusted to minimize the sum of the costs of all links.

This approach, however, does not take into account node congestion and has the disadvantage that too many of the shortest paths pass through a few nodes, called hubs. As a result, in situations of high network traffic, these hubs will experience congestion and eventually jamming, causing the network to break apart in a multitude of disconnected subnetworks. In light of this behavior, the optimality of the shortest path routing as currently implemented has been recently questioned [6–15]. It has been shown, for example, that dynamic routing protocols which allow for a certain degree of stochasticity or take into account the congestion status of the nearest neighbors significantly improve the transport capacity of a network [9–15].

A more systematic approach is to find better static routing protocols that avoid the hubs whenever possible (i.e., when avoiding a hub does not lead to congestion on another node). Recent studies [6,7] have shown that this is possible and propose new routing algorithms which lead to improved transport capacity (quantified by the packet insertion rate at which jamming occurs). An open question is how much larger the actual transport capacity can be than the results presented in Refs. [6,7]. We show that significant improvement in the transport capacity of a network can be achieved by systematically adjusting the traffic routing to minimize the maximum betweenness on the network. Our algorithm leads to higher transport capacity than those presented in Refs. [6,7]. The transport capacity also exceeds the analytical estimate for its maximum value given in [7]. Furthermore, we argue that our algorithm achieves near-optimal routing for uncorrelated scale-free networks.

To facilitate comparison, we use the same network model as in Ref. [7]. We present results for undirected, uncorrelated scale-free networks with an exponent of the power-law degree distribution $\gamma=2.5$, generated using the configuration model. The number of nodes N varies between 25 and 1600. For simplicity, we assume that all nodes have the same processing capacity of 1 packet per time step and that new information packets are inserted at every node at the same average rate of r packets per time step. The destinations of the packets are chosen at random from among the other $N-1$ nodes on the network. However, the algorithm can be generalized for nodes with different processing capacities and for arbitrary traffic demands.

Given a routing table, the betweenness B_i of node i is defined [16] as the sum of all fractional paths that pass through that node. The fraction of times a message passes through node i on its way from a source node s to a target node t is computed as follows: the source node s is assigned a weight 1 and then the weight of every node along each path is split evenly among its predecessors in the routing table on the way from t to s and added to the weights of the predecessors. The average number of packets passing through a given node i is then $\langle w\rangle_i = rB_i/(N-1)$. Jamming occurs at the critical average insertion rate r_c at which the average number

*Electronic address: dbogdan@mail.uh.edu
†Electronic address: bassler@uh.edu

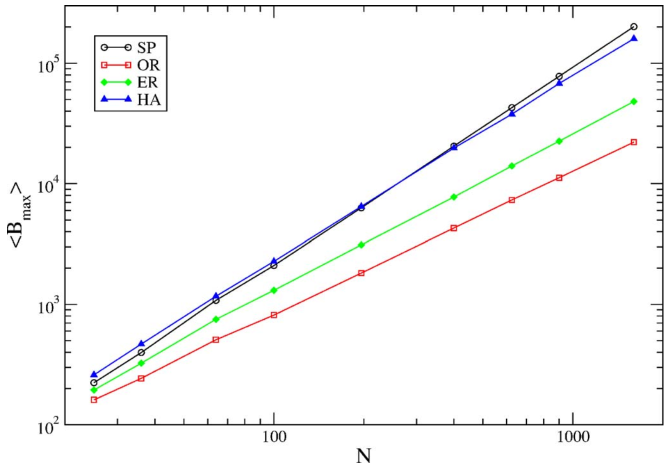

FIG. 1. (Color online) Ensemble average of the network maximum betweenness as a function of the number of nodes for four routing protocols.

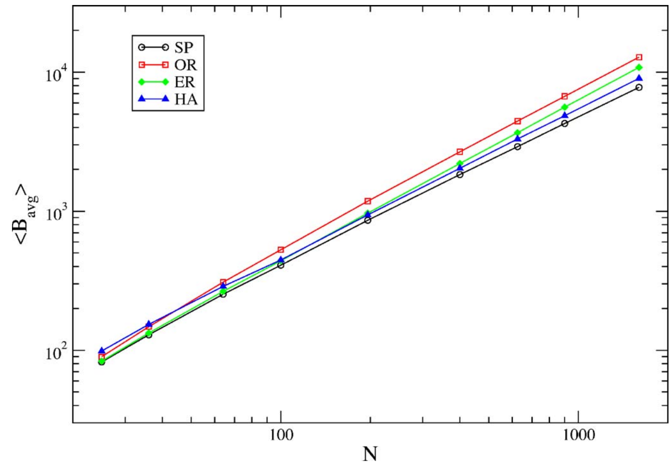

FIG. 2. (Color online) Ensemble average of the network average betweenness as a function of the number of nodes for four routing protocols.

of packets processed by the busiest node reaches unity. Consequently, r_c is given by [8]

$$r_c = \frac{N-1}{B_{max}}, \tag{1}$$

where B_{max} is the highest betweenness of any node on the network. Thus, to achieve optimal routing, the highest betweenness B_{max} should be minimized. An important point is that, even though the minimization procedure pertains to a single scalar quantity, such an optimization algorithm will implicitly reshape the betweenness landscape across the whole network, lowering traffic through the initially busy nodes at the expense of increased traffic through the initially idle nodes until the traffic spreads out and the narrowest possible betweenness distribution is achieved.

The problem of finding the exact optimal routing is mathematically tied to the problem of finding the minimal sparsity vertex separator [7], which has been shown [17] to be an *NP*-hard problem. This means that the number of flops necessary for the computation of an exact solution increases with the number of nodes N faster than any polynomial. We propose a heuristic algorithm which finds near-optimal solutions for the routing problem in time $O(N^4)$, at worst ($O(N^3)$ for one iteration, and requiring $O(N)$ iterations). In its simplest form, the algorithm proceeds as follows:

(1) Assign weight 2 to every link and compute the shortest paths between all pairs of nodes.

(2) Compute the betweenness of every node.

(3) Find the node which has the highest betweenness B_{max} and add 1 to the weight of every link that connects it to other nodes.

(4) Recompute the shortest paths. Go back to step 2.

Note that the algorithm picks the "least fit" element of a set and changes its parameters. Therefore, it is a form of extremal optimization algorithm [18]. However, this algorithm assigns parameters in a deterministic way, unlike many of the other existing extremal optimization algorithms.

Before presenting the results, we note that the average betweenness B_{avg} on a given network using a given routing table provides an absolute lower bound for the maximum betweenness. This lower bound would be achievable only if one could optimize routing to the point where all nodes experience the same traffic. Moreover, the average betweenness in the case of the shortest path routing with all weights set to 1, which henceforth will be called "shortest path routing," constitutes the lower bound for the average betweenness computed using any arbitrary routing table. This is because any changes in routing, including those resulting from an optimization algorithm, will result in longer paths, thus adding to the sum of all betweennesses on the network. It is thus apparent that a good optimization algorithm is required to have at least two properties: (1) minimize the difference between the maximum and average betweenness, and (2) do this while keeping the difference between the average betweenness computed using the optimized routing and the one computed using the shortest path routing as low as possible.

In the following, networks of a given size N are characterized by the ensemble averages of the maximum betweenness $\langle B_{max}\rangle$ and average betweenness $\langle B_{avg}\rangle$ computed over a set of 100 realizations of the network. Computer simulation results presented in Ref. [7] suggest a power-law functional dependence of $\langle B_{max}\rangle$ on the network size N. Our results confirm this power-law dependence for $\langle B_{max}\rangle$ and show a similar dependence in the case of $\langle B_{avg}\rangle$.

Results for the ensemble average of the maximum betweenness $\langle B_{max}\rangle$ as a function of the network size N for four routing protocols are presented in Fig. 1. Similar results for the ensemble average of the network average betweenness $\langle B_{avg}\rangle$ are shown in Fig. 2. The results are for the shortest path routing (SP), for our optimal routing (OR), and for the efficient routing (ER) and hub avoidance (HA) protocols described in Refs. [6] and [7], respectively. The exponents resulting from fitting the data points in each set are given in Table I, where the quoted errors are 2σ estimates. It is apparent from Figs. 1 and 2 that our optimization algorithm leads to a far smaller difference between the maximum and the average betweenness than in the case of the other three

TABLE I. Exponents of the $\langle B_{avg}\rangle$ and $\langle B_{max}\rangle$ power-law scaling with network size N.

	SP	OR	ER	HA
$\langle B_{avg}\rangle$	1.088±0.019	1.186±0.024	1.165±0.009	1.080±0.009
$\langle B_{max}\rangle$	1.634±0.010	1.185±0.009	1.315±0.017	1.542±0.010

routing protocols. Moreover, the maximum and the average betweenness scale with N with the same exponent, which is a strong argument in favor of the optimality of the routing. Finally, the difference $\langle B_{avg}\rangle_{\rm OR}-\langle B_{avg}\rangle_{\rm SP}$, while larger than $\langle B_{avg}\rangle_{\rm ER}-\langle B_{avg}\rangle_{\rm SP}$ or $\langle B_{avg}\rangle_{\rm HA}-\langle B_{avg}\rangle_{\rm SP}$ (which is explained by the need to have slightly longer paths around the hubs), is kept quite low and $\langle B_{avg}\rangle_{\rm OR}$ scales with the network size N with an exponent only slightly higher than $\langle B_{avg}\rangle_{\rm SP}$.

As expected with a heuristic algorithm, the evolution of the maximum betweenness as a function of the number of iterations is not monotonic, but exhibits a decreasing trend and eventually the maximum betweenness "converges" in the sense that it becomes confined to a narrow band. This is exemplified in Fig. 3, which is a plot of B_{max} versus the number of iterations for a network with 196 nodes. Figure 4 provides insight into how the algorithm works by comparing the initial and final betweenness distributions in the case of a network with 400 nodes. Figure 4(a) shows histograms of the betweenness distribution before and after optimization, while Fig. 4(b) shows the betweennesses plotted against the vertex index. Initially, the majority of the nodes have very low betweenness, but a small number of them are spread over a very wide range. After optimization, all node betweennesses are confined to a narrow band, whose upper edge is quite well-defined. Most of them are uniformly distributed within the band, but there is a very sharp peak at the upper edge.

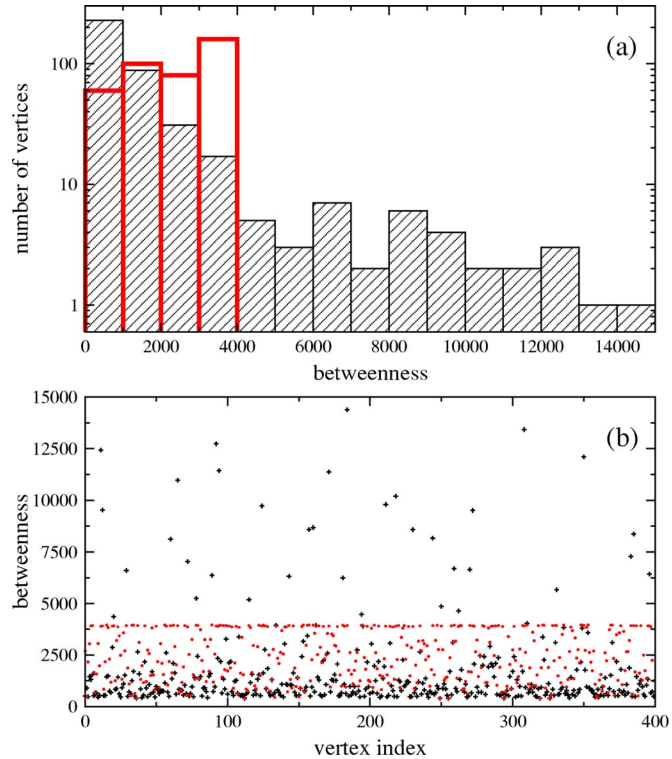

FIG. 4. (Color online) Distribution of node betweennesses before [black shaded bins in (a) and black crosses in (b)] and after [red hollow bins in (a) and red circles in (b)] optimization for a network with 400 nodes.

There is a significant decrease in the number of very low betweenness nodes.

A plot of the final (OR) betweenness versus the initial (SP) betweenness in the case of a network with 400 nodes is shown in Fig. 5. It is apparent from this plot that the algorithm performs remarkably well, by lowering traffic through all nodes whose initial betweenness lies above a certain critical value until they all reach essentially the same critical betweenness. On the other hand, virtually all nodes whose initial betweenness lies below the critical value experience

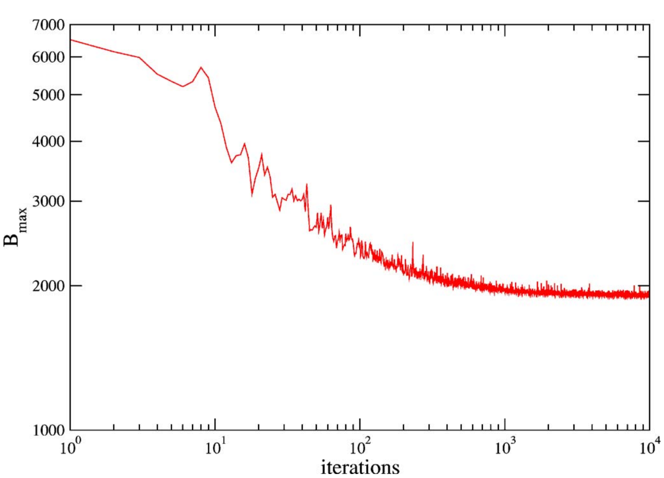

FIG. 3. (Color online) Maximum betweenness as a function of the number of iterations for a network with 196 nodes.

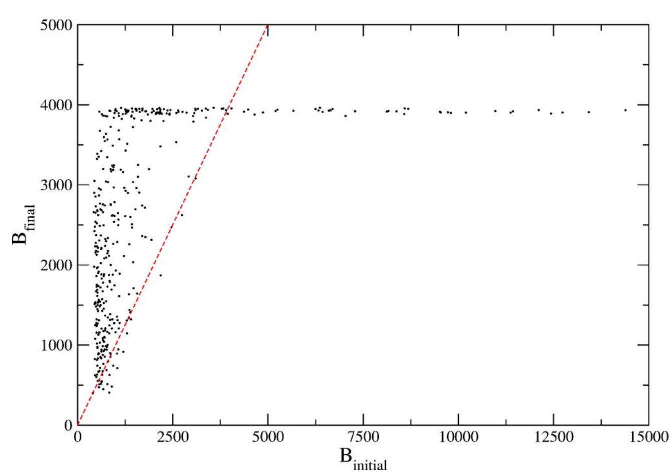

FIG. 5. (Color online) Correlation plot of the final (OR) versus initial (SP) betweenness for a network with 400 nodes.

higher traffic, many of them (especially those with higher initial betweenness) reaching the critical value. It is still an open question whether an improved algorithm can achieve a lower critical betweenness by further raising traffic through some initially low betweenness nodes. On the other hand, it is clear that not all low betweenness nodes can have their betweenness increased without unduly lengthening paths or increasing traffic through other nodes which are prone to congestion. The simplest examples are those of a small subnetwork whose only connection to the rest of the network is through a single link to a high degree (or otherwise high SP betweenness) node, or a triangle connected to the rest of the network only by containing such a node. In these cases, there is no way of diverting any of the high SP betweenness node's traffic between other nodes through the structures mentioned above. The latter will have low betweenness even in the case of rigorously optimal routing.

The difference between the SP and OR distribution of the travel times between the various nodes in situations when the SP routing does not lead to jamming is still an open question and will be the subject of a future study. However, we argue that, at least in situations of congested traffic, most travel times are shorter in the case of OR routing. It is known from queuing theory that the average queue length $\langle q\rangle$ is given (assuming unity processing power) by [8,19]

$$\langle q\rangle = \frac{\langle w\rangle}{1-\langle w\rangle}, \tag{2}$$

with $\langle w\rangle$ defined above Eq. (1). Due to this strongly nonlinear relationship, which diverges as $\langle w\rangle$ approaches unity, it seems reasonable to assume that by avoiding the passage through hubs with very high betweenness, most travel times become shorter, in spite of the fact that the routes pass through more nodes. This conclusion is also supported by the results for the average path length and average travel time presented in Ref. [6].

In summary, we have presented a simple heuristic algorithm for routing optimization on complex networks and demonstrated its usefulness for scale-free networks. This algorithm is useful in situations when network jamming is primarily due to node congestion. Our results show that the application of this algorithm allows a network to bear significantly higher traffic than in the case of shortest path routing. Network capacity is improved by a factor which increases with network size according to a power law.

The authors acknowledge support from the NSF through Grant No. DMR-0427538 and also from SI International through A. Williams of the Air Force Research Laboratory Information Directorate under Contract No. FA8750-04-C-0258.

[1] M. Ericsson, M. G. C. Resende, and P. M. Pardalos, J. Comb. Optim. **6**, 299 (2002).
[2] B. Fortz and M. Thorup, IEEE J. Sel. Areas Commun. **20**(4), (2002).
[3] V. Gabrel, A. Knippel, and M. Minoux, J. Heuristics **9**, 429 (2003).
[4] D. Allen, I. Ismail, J. Kennington, and E. Olinick, J. Heuristics **9**, 375 (2003).
[5] E. Mulyana and U. Killat, European Transactions on Telecommunications **16**(3), 253–261 (2005).
[6] G. Yan, T. Zhou, B. Hu, Z.-Q. Fu, and B.-H. Wang, Phys. Rev. E **73**, 046108 (2006).
[7] S. Sreenivasan, R. Cohen, E. López, Z. Toroczkai, and H. E. Stanley, e-print cs.NI/0604023.
[8] R. Guimerà, A. Díaz-Guilera, F. Vega-Redondo, A. Cabrales, and A. Arenas, Phys. Rev. Lett. **89**, 248701 (2002).
[9] P. Echenique, J. Gómez-Gardeñes, and Y. Moreno, Phys. Rev. E **70**, 056105 (2004).
[10] P. Echenique, J. Gómez-Gardeñes, and Y. Moreno, Europhys. Lett. **71**, 325 (2005).
[11] L. Zhao, Y.-C. Lai, K. Park, and N. Ye, Phys. Rev. E **71**, 026125 (2005).
[12] K. Park, Y.-C. Lai, L. Zhao, and N. Ye, Phys. Rev. E **71**, 065105(R) (2005).
[13] Z. Toroczkai and K. E. Bassler, Nature (London) **428**, 716 (2004).
[14] Z. Toroczkai, B. Kozma, K. E. Bassler, N. W. Hengartner, and G. Korniss, e-print cond-mat/0408262.
[15] D. J. Ashton, T. C. Jarrett, and N. F. Johnson, Phys. Rev. Lett. **94**, 058701 (2005).
[16] M. E. J. Newman, Phys. Rev. E **64**, 016132 (2001).
[17] T. N. Bui and C. Jones, Inf. Process. Lett. **42**, 153 (1992).
[18] S. Boettcher and A. G. Percus, Phys. Rev. Lett. **86**(23), 5211 (2001).
[19] O. Allen, *Probability, Statistics and Queuing Theory with Computer Science Application*, 2nd ed. (Academic Press, New York, 1990).