

Search in spatial scale-free networks

This article has been downloaded from IOPscience. Please scroll down to see the full text article.

2007 New J. Phys. 9 190

(<http://iopscience.iop.org/1367-2630/9/6/190>)

View [the table of contents for this issue](#), or go to the [journal homepage](#) for more

Download details:

IP Address: 38.107.179.210

The article was downloaded on 20/02/2012 at 10:18

Please note that [terms and conditions apply](#).

Search in spatial scale-free networks

H P Thadakamalla^{1,3}, R Albert² and S R T Kumara¹

¹ Department of Industrial Engineering, The Pennsylvania State University, University Park, Pennsylvania, 16802, USA

² Department of Physics, The Pennsylvania State University, University Park, Pennsylvania, 16802, USA

E-mail: hpt102@psu.edu, ralbert@phys.psu.edu and skumara@psu.edu

New Journal of Physics **9** (2007) 190

Received 12 March 2007

Published 28 June 2007

Online at <http://www.njp.org/>

doi:10.1088/1367-2630/9/6/190

Abstract. We study the decentralized search problem in a family of parameterized spatial network models that are heterogeneous in node degree. We investigate several algorithms and illustrate that some of these algorithms exploit the heterogeneity in the network to find short paths by using only local information. In addition, we demonstrate that the spatial network model belongs to a class of *searchable networks* for a wide range of parameter space. Further, we test these algorithms on the US airline network which belongs to this class of networks and demonstrate that searchability is a generic property of the US airline network. These results provide insights on designing the structure of distributed networks that need effective decentralized search algorithms.

³ Author to whom any correspondence should be addressed.

Contents

1. Introduction	2
2. Literature and problem description	3
3. Decentralized search algorithms	4
4. Spatial network model and search analysis	7
4.1. Simulation and analysis	7
5. Search in the US airline network	11
5.1. Properties of the US airline network	11
5.2. Search results and analysis	12
6. Conclusions and discussion	15
Acknowledgments	16
References	16

1. Introduction

Recently, many large-scale distributed systems in communications, sociology, and biology have been represented as networks and their macroscopic properties have been extensively studied [1]–[4]. One of the major findings is the presence of heterogeneity in network properties. For example, the distribution of node degree (i.e. the number of edges incident on a node) for many real-world networks including the Internet, the World Wide Web, phone call networks, scientific collaboration networks and metabolic networks is found to be highly heterogeneous and to follow a power-law, $p(k) \sim k^{-\gamma}$ where $p(k)$ is the fraction of nodes with degree k . The clustering coefficients, quantifying local order and cohesiveness [5], are also found to be heterogeneous, i.e. $C(k) \sim k^{-1}$ [6]. Further, in many networks the node betweenness centrality, which quantifies the number of shortest paths that pass through a node, is found to be heterogeneous [7]. These heterogeneities have a demonstrably large impact on the network’s resilience [8, 9] as well as navigation, local search [10, 11], and spreading processes [12].

Another interesting property exhibited by these networks is the ‘small-world phenomenon’ whereby almost every node is connected to every other node by a path with a small number of edges. This phenomenon was first demonstrated by Milgram’s famous experiment in 1960 [13]. Milgram randomly selected individuals from Wichita, Kansas and Omaha, Nebraska and requested them to direct letters to a target person in Boston, Massachusetts. The participants, and consecutively each person receiving the letter, were asked to send it to an acquaintance whom they judged to be closer to the target. Surprisingly, the average length of these paths (i.e. the number of edges in the path) was approximately 6, illustrating the small-world property of social networks. An even more striking observation, which was later pointed out by Kleinberg [14]–[16], is that the nodes (participants) were able to find short paths by using only local information. Currently, Dodds *et al* are carrying out an Internet-based study to verify this phenomenon, and initial findings are published in [17].

The observation by Kleinberg raises two fundamental questions: (i) Why should social networks be structured in a way that local search is efficient? (ii) What is the structure of networks that exhibit this phenomenon? Kleinberg [14] and later Watts *et al* [18] argued that the emergence of such a phenomenon requires special topological features. They termed the

networks in which short paths can be found using only local information as *searchable networks*. These studies along with a few others [10, 19] stimulated research on decentralized searching in complex networks [11], [20]–[26], a problem with many practical applications. In many networks, information such as data files and sensor data is stored at the nodes of a distributed network. In addition, the nodes have only limited or local information about the network. Hence, to access this information quickly, one should have efficient algorithms that can find the target node using the available local information. Examples include routing of sensor data in wireless sensor networks [27, 28], locating data files in peer-to-peer networks [26, 29], and finding information in distributed databases [30]. For the search process to be efficient, it is important that these networks are designed to be searchable. The importance of search efficiency becomes even more imminent in the case of ad-hoc networks, where the networks are decentralized and distributed, and real time searching is required to find the target node.

In this paper, we study the decentralized search problem in a family of parameterized spatial network models that are heterogeneous in node degree. We propose several decentralized search algorithms and examine their performance by simulating them on the spatial network model for various parameters. As pointed out in [25], our analysis reveals that the optimal search algorithm should effectively incorporate the direction of travel and the degree of the neighbour. We illustrate that some of these algorithms exploit the heterogeneities present in the network to find paths as short as the paths found by using global information; thus we demonstrate that the spatial network model considered defines a class of searchable networks. Further, we test these algorithms on the US airline network which belongs to this class of networks and show that searchability is a generic property of the US airline network.

2. Literature and problem description

Decentralized searching in networks can be broadly classified into searching in unstructured networks (as in peer-to-peer networks such as Gnutella [29]) and in structured/spatial networks (as in wireless sensor networks). In unstructured networks, the global position of a node cannot be quantified and it is difficult to know whether a step in the search process is towards the target node or away from the target node. Hence, it is difficult to obtain short paths using local information. In unstructured networks with power-law degree distributions, Adamic *et al* [10] showed that a high-degree seeking search is better than a random-walk search. In a random-walk search, the node that has the message passes it to a randomly chosen neighbour, and the process continues until it reaches the target node. Whereas, in a high-degree search, the node that has the message passes it to the neighbour with highest degree. Thadakamalla *et al* [11] proposed a more general algorithm based on a local measure, local betweenness centrality (LBC), for networks which are heterogeneous both in edge weights and in node degree. They demonstrated that the search based on LBC utilizes the heterogeneities in edge weights and node degree to perform the best in power-law (scale-free) weighted networks.

In structured networks the nodes are embedded in a metric space and they are connected based on the metric distance. Here, the global position of the target node in the space can guide the search process to reach the target node more quickly. In [14, 15], Kleinberg studied search in a family of grid-based models that generalize the Watts–Strogatz [5] model. He proved that only one particular model among this infinite family can support efficient decentralized algorithms. In this model, a simple greedy search, where the node passes the message to the neighbour closest

to the target node based on the grid distance, is able to give short paths. He further extended this model to hierarchical networks [16], where, again, the network was proven to be searchable only for a specific parameter value. Unfortunately, the model given by Kleinberg represents only a very small subset of complex networks. Independently, Watts *et al* presented another model based upon plausible hierarchical social structures [18], to explain the phenomena observed in Milgram's experiment. The networks were shown to be searchable by a greedy search algorithm for a wide range of parameter space. Other works on decentralized searching include [20]–[26]. Simsek and Jensen [25] use homophily between nodes and degree disparity in the network to design a better algorithm for finding the target node. However, finding an optimal way to combine location and degree information is yet to be investigated (see [21] for a review). Another interesting problem studied by Clauset and Moore [31], and by Sandberg [24], is the question of how real-world networks evolve to become searchable. They propose a simple feedback mechanism where the nodes continuously conduct decentralized searches, and in the process partially rewire the edges to form a searchable network.

In this paper, we consider search in a family of parameterized spatial network models that are heterogeneous in node degree. In this model, nodes are placed in an n -dimensional space and are connected, based on preferential attachment and geographical constraints, to form spatial scale-free networks. Preferential attachment to high-degree nodes is believed to be responsible for the emergence of the power-law degree distribution observed in many real-world networks [32], and geographical constraints account for the fact that nodes tend to connect to nodes that are nearby. Many real-world networks such as the Internet [33] and the worldwide airline network [34], can be described by this family of spatial network models. Our objective is to design decentralized search algorithms for this type of network model and demonstrate that this simple model defines a class of searchable networks. The decentralized search algorithm attempts to send a message from a starting node s to the target node t along the edges of the network using local information. Each node has information about the position of the target node, the position of its neighbours, and the degree of its neighbours. Using this information, the start node, and consecutively each node receiving the message, passes the message to one of its neighbours based on the search algorithm until it reaches the target node. We evaluate each algorithm based on the number of hops taken for the message to reach the target node; the lower the number, the better the performance of the algorithm. Another potentially relevant measure is the physical distance travelled by each search algorithm. However, the number of hops is the most pertinent distance measure in many networks, including social networks, the Internet and even airline networks, as the delays associated with switching between edges are comparable to the delays associated with traversing an edge.

As observed in previous studies [10, 11], we expect that the heterogeneity present in spatial scale-free networks influences the search process. In the following section, we discuss why the degree of a node's neighbour is important and propose different ways of composing the direction of travel and the degree of the neighbour.

3. Decentralized search algorithms

A simple search algorithm in spatial networks is *greedy search*, where each node passes the message to the neighbour closest to the target node. Let d_i be the distance to the target node from each neighbour i (see figure 1(a)) and let k_i be the degree of the neighbour i .

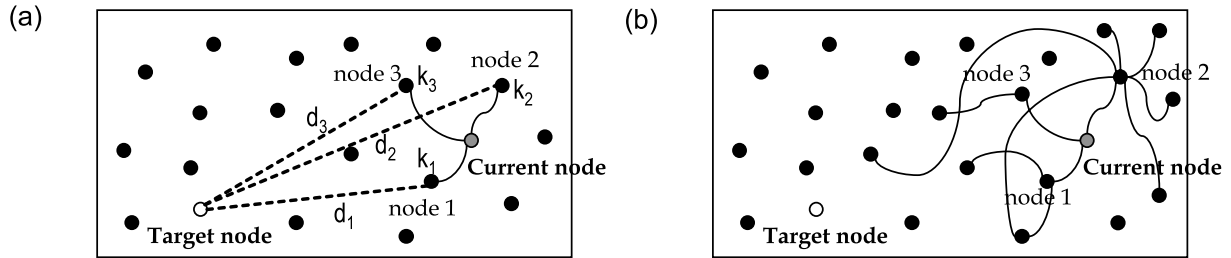


Figure 1. (a) Illustration of a spatial network. d_i is the distance to the target node from each neighbour i and k_i is the degree of the neighbour i . (b) Illustration for demonstrating that sometimes it is better to choose a neighbour with higher degree i.e. node 2 over node 1, even if we are going away from the target. This will give higher probability of taking a longer step in the next iteration.

Greedy search chooses the neighbour with the smallest d_i . This will ensure that the message is always going to the neighbour closest to the target node. However, greedy search may not be optimal in spatial scale-free networks that have high heterogeneity in node degree. Adamic *et al* [10] and Thadakamalla *et al* [11] have shown that search algorithms that utilize the heterogeneities present in the network perform substantially better than those that do not. Indeed, choosing a neighbour with higher degree, even by going away from the target node, gives a higher probability of taking a longer step in the next iteration. For instance, in figure 1(b), it is better to choose node 2 instead of node 1 since node 2 can take a longer step towards the target node in the next iteration. In the following paragraph, we show that the expected distance a neighbour can take in the next iteration is a strictly increasing function of its degree.

We define the length of an edge as the Euclidian distance between the two nodes connected by the edge. Let $P(X)$ be the probability distribution of edge lengths. Let $Y_k = \text{Max}\{X_1, X_2, X_3, \dots, X_k\}$, where $X_1, X_2, X_3, \dots, X_k$ are independent and identically distributed (i.i.d.) random variables with distribution function $P(X)$. The cumulative distribution function of Y_k is

$$P[Y_k \leq y] = \prod_{i=1}^k P[X_i \leq y] = [P(X_1 \leq y)]^k.$$

This implies

$$E(Y_k) = \int_0^{\infty} (1 - [P(X_1 \leq y)]^k) dy.$$

Since $P(X_1 \leq y) \leq 1 \forall y$,

$$[P(X_1 \leq y)]^{k_1} \leq [P(X_1 \leq y)]^{k_2} \quad \text{if } k_1 \geq k_2,$$

implying that

$$E(Y_{k_1}) \leq E(Y_{k_2}) \forall y \quad \text{if } k_1 \leq k_2$$

Similarly, we can show that if $P(X)$ is not a delta function then

$$E(Y_{k_1}) < E(Y_{k_2}) \quad \text{if } k_1 < k_2.$$

Now consider two neighbours n_1 and n_2 with degree k_1 and k_2 . The expected distance the neighbours n_1 and n_2 can take in the next iteration irrespective of the direction is given by $E[Y_{k_1-1}]$ and $E[Y_{k_2-1}]$ respectively. This implies that $E[Y_{k_1-1}] > E[Y_{k_2-1}]$ if $k_1 > k_2$. Here, we approximate that $X_1, X_2, X_3, \dots, X_k$ are independent which is valid when the number of edges is large. Hence, if we choose a neighbour with higher degree then there is a greater probability of taking a longer step in the next iteration. Thus one expects that in spatial scale-free networks the efficient algorithm should combine the direction of travel, quantified by d_i , and the degree of the neighbour, k_i , into one measure. Since the units of d_i and k_i are different, there is no trivial way of composition that is optimal. The aim of the measure is to choose a neighbour with smaller d_i and larger k_i with an intuition that a higher degree node should effectively decrease the distance from the target—a goal which can be achieved in many different ways. One could give an incentive $g(k_i)$, and then subtract it from the distance d_i ; one could also divide d_i either by k_i or by any increasing function of k_i . We investigated the following search algorithms, which cover a broad spectrum of possibilities.

1. *Random walk*: the node attempts to reach the target by passing the message to a randomly selected neighbour.
2. *High-degree search*: the node passes the message to the neighbour with the highest degree. The idea here is that by choosing a neighbour that is well-connected, there is a higher probability of reaching the target node. Note that this algorithm requires the fewest number of hops to reach the target in unstructured networks [10].
3. *Greedy search*: the node passes the message to the neighbour i with the smallest d_i . This will ensure that the message is always going to the neighbour closest to the target node.
4. *Algorithm 4*: the node passes the message to the neighbour i with the smallest measure $d_i - g(k_i)$. The function $g(k_i)$ is an incentive for choosing a neighbour of higher degree. Ideally, $g(k_i)$ should be the expected maximum length of an edge from a node with degree k_i .
5. *Algorithm 5*: the node passes the message to the neighbour i that has the smallest measure $(\frac{d_i}{d_m})^{k_i}$, where d_m is the Euclidian distance between the most spatially distant nodes in the network, and is used for normalizing d_i . We assume that d_m is known to all the nodes in the network. Note that the algorithm prefers the neighbour that has lower d_i and higher k_i .
6. *Algorithm 6*: the node passes the message to the neighbour i that has the smallest measure $\frac{d_i}{k_i}$. Here, again, the algorithm prefers the neighbour that has lower d_i and higher k_i .
7. *Algorithm 7*: the node passes the message to the neighbour i that has the smallest measure $(\frac{d_i}{d_m})^{\ln k_i + 1}$. This is a conservative version of algorithm 5 with respect to k_i .
8. *Algorithm 8*: the node passes the message to the neighbour i that has the smallest measure $\frac{d_i}{\ln k_i + 1}$. This algorithm is weaker version of algorithm 6 with respect to k_i .

Algorithms from 4 to 8 aim to capture both the direction of travel and the neighbours' degree. Thus, we expect these algorithms to give smaller path lengths than other algorithms. In the case of algorithm 4, it would be extremely difficult to define a function independent of the parameters of the network. Hence, it may not be realistic to use this form of composition for direction of travel and degree of neighbour. Even greedy search has a slight preference for high-degree nodes, since the probability of reaching a node with degree k is $\sim kp_k$ [35], where p_k is the fraction of nodes

with degree k . Hence, the proposed algorithms have to be extremely competitive to perform better than greedy search. The algorithms described above are mainly based on intuition. However, as we discuss later in the paper, the successful strategies are not restricted to these functional forms.

4. Spatial network model and search analysis

The spatial network model we consider incorporates both preferential attachment and geographical constraints. At each step during the evolution of the spatial network model one of the following occurs [36]:

1. with probability p , a new edge is created between two existing nodes in the network;
2. with probability $1 - p$, a new node is added and connected to m existing nodes in the network, with the constraint that multiple edges are not formed.

In both cases, the degrees of the nodes and the distances between them are considered when forming a new edge. In the first case, two nodes i and j are selected according to

$$\Pi_{ij} \propto \frac{k_i k_j}{F(d_{ij})},$$

where k_i is the degree of node i , d_{ij} is the Euclidian distance between nodes i and j and $F(d_{ij})$ is an increasing function of d_{ij} . A new node i is uniformly and randomly placed in an n -dimensional space and is connected to a pre-existing node j with probability

$$\Pi_j \propto \frac{k_j}{F(d_{ij})}.$$

The above process is simulated until the number of nodes in the network is N . Let the network generated be $G(N, p, m, F, n)$. Here, the preferential attachment mechanism leads to a power-law degree distribution where the exponent can be tuned by changing the value of p [36] (see figure 2(a)). $F(d)$ controls the truncation of the power-law decay, and if $F(d)$ increases rapidly, then the power-law decay regime can disappear altogether [37]. Two widely-used functions for $F(d)$ are d^r [33] and $\exp(d/d_{\text{char}})$ [37].

4.1. Simulation and analysis

We investigate the search algorithms by simulating them on the networks generated by the above spatial network model. We generate the network on a two-dimensional grid with length $a = 1000$, breadth $b = 500$, and $m = 1$ for different values of N , p , and different functions F . Once the network is formed, we randomly choose K pairs (source and target) of nodes and simulate the search algorithms. The source, and consecutively each node receiving the message, passes the message to one of its neighbours, according to the search algorithm. For algorithm 4, we assume the incentive function $g(k_i)$ to be the expected maximum distance a node with degree k_i can take for the next hop, that is, the expected maximum length of an edge from a node with degree k_i . Empirically we found that this function follows the form $c_1 * \ln k_i + c_2$ for all the spatial networks. For algorithms 5 and 7, we let d_m be $\sqrt{a^2 + b^2}$, the largest distance between two points in the

Table 1. Comparison of search algorithms on a spatial scale-free network of 1000 nodes in a two-dimensional space with length and breadth equal to 1000 and 500, respectively. l is the average path length for the paths found by the search algorithm, d_{path} is the average physical distance for the paths found by each search algorithm and c is the percentage number of times the path was not found. The table summarizes the average of l , d_{path} and c obtained from 10 simulations of the network with parameters $p = 0.72$ and r for 2000 pairs. Note that the decentralized algorithms 5, 6, 7 and 8 perform as well as the shortest paths found by using global information. Even though the greedy search performs well for the paths found (l and d_{path}), it is sometimes unable to find a path (c).

	$r = 1$			$r = 2$			$r = 3$		
	l	d_{path}	$c(\%)$	l	d_{path}	$c(\%)$	l	d_{path}	$c(\%)$
Random walk	41.68	10957	0	70.47	9414	0	138.07	9024	0
High-degree search	28.35	8032	0	54.85	8805	0	120.15	9848	0
Greedy search	3.37	787	0.17	3.59	600	0.83	4.53	537	2.11
Algorithm 4	10.22	2303	0.12	14.07	1987	0.46	20.08	1806	1.87
Algorithm 5	2.47	646	0	2.97	594	0	4.51	677	0.02
Algorithm 6	2.45	636	0	2.85	565	0	3.73	573	0.02
Algorithm 7	2.54	631	0	2.80	539	0	3.52	527	0.02
Algorithm 8	2.66	646	0	2.87	537	< 0.01	3.54	514	0.07
Shortest path length	2.27	531	NA	2.55	435	NA	3.05	403	NA

considered space. We assume that it is sufficient if the message reaches a small neighbourhood of the target node defined by a circle with radius D . This is a realistic assumption in many real-world networks, e.g. it is sufficient if we reach one of the airports in the close neighbourhood of a destination city (especially when the city has multiple airports). The search process continues until the message reaches a neighbour of the target node or a node within a circle of radius $D = 50$ centred around the target node. In order to avoid passing the message to a neighbour that has already received the message, a list L is maintained. During the search process, if the message reaches a node i whose neighbours are all in the list L , then the message is passed to one of the neighbours using the same algorithm. In the case of random walk or high degree search, the message is routed back to the previous node and this particular neighbour i is marked to note that it cannot pass the message any further. If the number of hops exceeds $N/2$, then the search process stops, noting that the path was not found. For each search algorithm, the average path length, l , measured as the number of edges in the path, the average physical distance travelled along the path, d_{path} , and the percentage of times the search algorithm is unable to find a path, c , are computed from the search results obtained for K pairs in 10 instances of the network model. The lower the value of l , d_{path} and c , the better the performance of the search algorithm. We use the shortest average path length and average physical distance obtained by global breadth-first-search (BFS) algorithm and Dijkstra's algorithm [38] respectively, as a benchmark for comparing the performance of the search algorithms.

Table 1 compares the performance of different search algorithms for the spatial network, $G(1000, 0.72, 1, d^r, 2)$ with $r = 1, 2$ and 3. We find that the decentralized search algorithms 5,

Table 2. Comparison of search algorithms on spatial scale-free networks with different parameters. l is the average path length for the paths found by each search algorithm and c is the percentage number of times the path was not found. The table summarizes the average of l and c obtained from 10 simulations of the network with parameters N , p , r and d_{char} . Note that the decentralized algorithms 5, 6, 7 and 8 perform as well as the shortest path found by using global information. Even though the greedy search performs well for the paths found (l), it is sometimes unable to find a path (c).

	$N = 1000, r = 1$				$p = 0.72, r = 1$				$N = 1000, p = 0.72$			
	$p = 0.30$		$p = 0.80$		$N = 500$		$N = 1500$		$d_{\text{char}} = 0.5$		$d_{\text{char}} = 2.0$	
	l	$c(\%)$	l	$c(\%)$	l	$c(\%)$	l	$c(\%)$	l	$c(\%)$	l	$c(\%)$
Greedy search	6.55	7.93	2.90	0.09	4.09	0.24	3.10	0.44	3.64	0.18	3.92	0.1
Algorithm 5	3.41	0.02	2.35	0	2.83	0	2.40	0	2.46	0.03	2.55	0
Algorithm 6	3.38	0.04	2.38	0	2.81	0	2.38	0	2.49	0	2.59	0
Algorithm 7	3.59	0.19	2.40	0	2.95	0	2.43	0.01	2.66	0.02	2.78	0
Algorithm 8	4.12	0.73	2.49	< 0.01	3.16	< 0.01	2.54	0	2.79	0.04	3.01	0.01
Shortest path length	2.91	NA	2.16	NA	2.30	NA	2.26	NA	2.23	NA	2.23	NA

6, 7 and 8 perform as well as the shortest path obtained using global information of the network. Specifically, the difference between the shortest path and the path obtained by algorithms 6 and 7 is less than a hop. These results are surprising because the latter algorithms only use the local information in the network, yet they perform as well as the BFS algorithm. This behaviour is mainly due to the power-law nature of the spatial network: the few nodes with high-degree are allowing the algorithms to make big jumps during the search process (see table 1). This conclusion is corroborated by the fact that an increase in r , meaning a decrease in the power-law regime in the degree distribution [37], induces an increase in the path length. Greedy search which uses only the direction of travel is able to find short paths (compare l 's in table 1) but for a few node pairs it is unable to find a path (compare c 's in table 1). Greedy search does not consider the degree of the nodes and sometimes the algorithm gets stuck in a loop in sparsely connected regions of the network. In the case of algorithm 4, the composition was not very effective. It is likely that the values of the coefficients, which are difficult to compute, were not optimal. Moreover, the optimal values are highly dependent on the parameters and the configuration of the spatial network. Hence, it would be difficult to generalize the algorithm for all networks and we will not consider it further in our analysis. Random-walk and high-degree search do not consider the direction of travel and hence take an exorbitantly large number of hops. Further, we found that the search algorithms' performance with respect to the path length l and physical distance metric d_{path} was similar. Hence, in the rest of our analysis, we do not discuss these two algorithms and the physical distance metric since the results do not add significant new information.

Similar results are obtained for a wide range of parameters for the spatial network model. Table 2 summarizes the results for some of these parameter values. This parameter space covers a broad range of power-law networks with different properties. For example, as the value of p changes from 0.3 to 0.8, the power-law exponent of the degree distribution changes from 2.4 to 1.7 (see figure 2(a)), which is the usual range of many real-world networks [1]–[4]. Hence

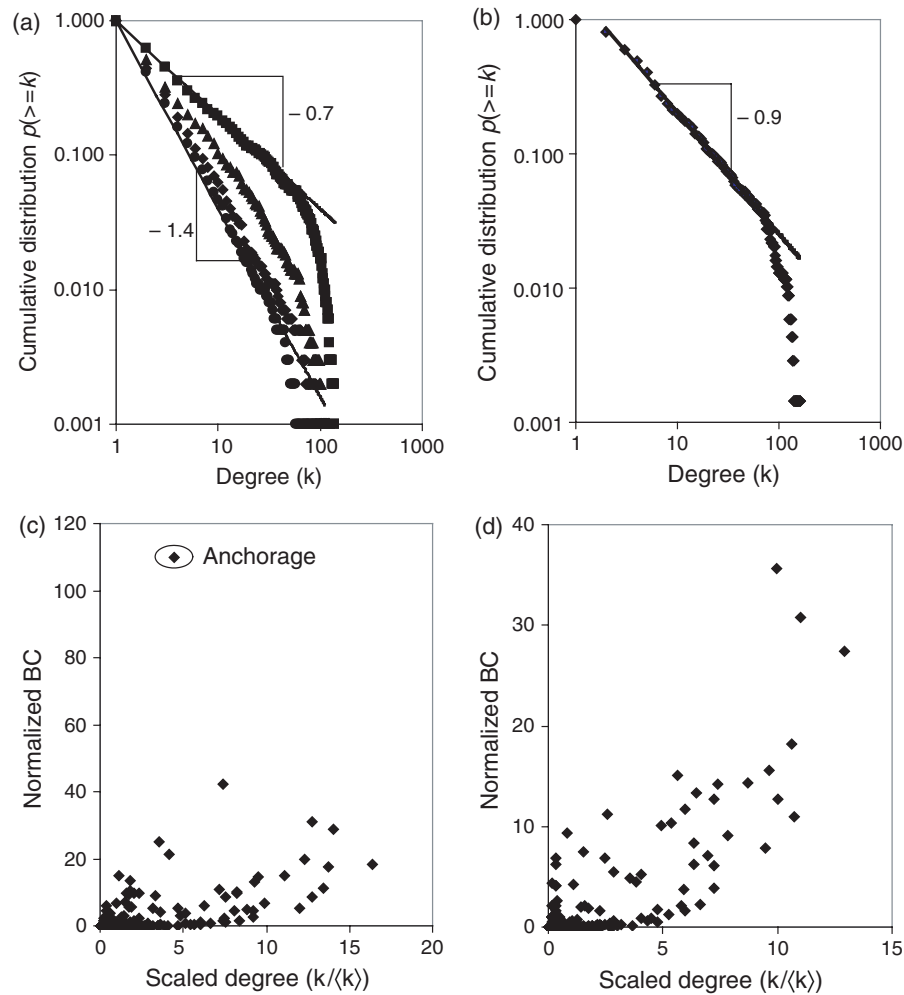


Figure 2. (a) Cumulative degree distribution of the networks generated by the spatial network model for different values of p . The symbols represent $p = 0.3$ (\bullet), 0.4 (\blacklozenge), 0.6 (\blacktriangle), and 0.8 (\blacksquare). The power-law exponent of the network can be tuned by changing the value of p . (b) Cumulative degree distribution of the US airline network. (c) Scaling of normalized BC of a node i with its scaled degree for the US airline network. Note that unlike random graphs, there exists no scaling between BC and degree of the node. (d) Scaling of normalized BC of a node i with its scaled degree for the US airline network without Alaska. Note that there is better correlation between BC and degree of the node when compared with the US airline network.

we can affirm that the spatial network model belongs to a general class of *searchable networks*. Although we have restricted our results to a discussion of two-dimensional spatial networks, it is easy to verify that these results will be valid for higher dimensions. Further, a large number of decentralized search algorithms are efficient. For instance, in algorithm 6 we divide d_i by k_i , whereas in algorithm 8 we divide d_i by $\ln k_i + 1$ which scales logarithmically with k_i . Both algorithms are found to be efficient. This implies that a wide range of functions $f(x)$ that scale

between x and $\ln x$ can be used for decentralized search. Hence, we find that the dependence of the search algorithms on the functional forms is weak and the searchability of these networks lies in their heterogeneous structure rather than the functional forms used in the search algorithm.

5. Search in the US airline network

Let us consider the US airline network, where nodes are the airports and two nodes are connected by an edge if there is a direct flight from one airport to another. In this network, navigating along an edge from one node to another represents flying from one airport to another. Suppose our objective is to travel from one place to another using the US airline network. In real life, one can obtain a choice of itineraries from the closest airport to the departure location (departure airport) to the closest airport to the destination location (destination airport) using various sources such as travel agents, airline offices or the World Wide Web. These sources have global information about the network and one can choose the itinerary based on different criteria, such as travel fare, number of stopovers, or total time of travel. Now consider a different scenario—one in which we do not have access to the global information of the network, and each airport has only local information. In other words, each airport has information about the location of the airports it can fly to and how well these neighbouring airports are connected (their degree). We do know the location of the departure airport and the destination airport. The objective is to find a path with the fewest stopovers from the departure airport to the destination. From the departure airport, and consecutively from each intermediate airport, we choose to fly to one of its neighbours based on the degree of the neighbouring airport, its location and the location of the destination airport. This process continues until we reach the destination airport or any other airport within a small neighbourhood of the destination airport. In real life, it is sufficient if we reach one of the airports near the destination airport. For example, it is sufficient to reach LaGuardia Airport (LGA), New York City if the objective is to reach John F Kennedy International Airport (JFK), New York City. In our study, as a first-order approximation we do not consider the type of airline or travel fare as important parameters. Even though this method of travel is unrealistic, it provides insights on the performance of decentralized search algorithms on real-world networks.

5.1. Properties of the US airline network

The Bureau of Transportation Statistics [39] has a well-documented database on the departure schedule, number of passengers, flight type etc for all the flights in the USA. We considered the data collected for the service class F (scheduled passenger service) flights during the month of January 2006 to form the US airline network. Each airport is represented as a node and a direct flight connection from one airport to another is depicted as a directed edge. We filtered the data to remove the anomalous edges formed due to redirected flights caused by environmental disturbances or random failures. Further, one would expect to have a flight from airport A to airport B if there is one from B to A; but for a small number of instances this was not true. To simplify the analysis, we added edges to make the network undirected.

After filtering the data, the airline network had 710 nodes and 3414 edges. The number of nodes and edges in the largest connected component (LCC) were 690 and 3412 respectively. The rest of the analysis in the paper considers only the LCC of the network. Not surprisingly, the properties of the US airline network are very similar to the properties of the world wide airline

network (WWN) [7]. The average path length for the airline network, which is the average minimum number of flights one has to take to go from one airport to any other, is 3.6. The clustering coefficient, which quantifies local order of the network measured in terms of the number of triangles (3-cliques) present, is 0.41. Hence, the US airline network is also a small-world network [5]. The degree distribution of the network follows a power-law $p(k) \sim k^{-\gamma}$ with exponent $\gamma = 1.9 \pm 0.1$ (see figure 2(b)), which is close to the exponent of the WWN, 2.0 ± 0.1 [7]. Further, as observed in the WWN, we find that the most connected airports are not necessarily the most central airports. Figure 2(c) plots the normalized betweenness centrality (BC) of a node i , $(b_i/\langle b \rangle)$, where $\langle b \rangle$ is the average BC of the network, versus its scaled degree $k_i/\langle k \rangle$, where $\langle k \rangle$ is the average degree of the network. The geopolitical considerations used to explain this phenomenon in the WWN [34] do not apply to the US airline network, as it belongs to a single country. In fact, this behaviour is due to Alaska which contains a significant percentage of the airports (255 of 690, close to 34%) yet only a few (around 6) are connected to airports outside of Alaska. For instance, the BC of Anchorage, Alaska is significantly higher than its degree (see figure 2(c)). If we remove the Alaska airports from the network, then we observe better correlation between the degree of a node and its BC (see figure 2(d)).

If an area is separated from the US mainland (such as Alaska and Hawaii), then very few airports connect it to the mainland and it may be difficult for search algorithms to capture these connections between the mainland and the other areas. To investigate the effects of this property on the search process, we simulate the algorithms on three different networks, namely, the US airline network, the US airline network without Alaska and the US mainland airline network without Alaska, Hawaii, Puerto Rico, the US Virgin Islands and the US Pacific Trust Territories and Possessions (US mainland network). The latter two networks have statistical properties similar to those of the US airline network. The US airline network without Alaska has 459 nodes and 2857 edges with 455 nodes and 2856 edges in the LCC; the US mainland network has 431 nodes and 2729 edges with 427 nodes and 2728 edges in the LCC.

5.2. Search results and analysis

We simulated the search algorithms for all $N * (N - 1)$ pairs in each network, where N is the number of nodes. The US airline network, the US airline network without Alaska, and the US mainland network had 475 410, 206 570, and 181 902 pairs respectively. We chose d_m to be the largest distance between two airports in the network and the neighbourhood distance D to be 100 miles. Table 3 summarizes the results obtained by each search algorithm. l is the average path length obtained for the paths found by the search algorithm, and c is the number of times the search algorithm was unable to find a path. The results are similar to the results obtained for the spatial scale-free network model. Algorithms 6, 7 and 8 are able to find paths as short as the paths obtained by the BFS algorithm. Again, greedy search is able to give short paths when it is able to find paths, but there were instances in which it was unable to find any path. In the case of the US airline network without Alaska and the US mainland network, the performance of the search algorithms is even better, especially for algorithm 5 which did not perform well for the complete US airline network. Figure 3 visualizes the paths obtained in a characteristic case when greedy search takes a higher number of hops. Often the greedy search reaches the nodes which are near to the destination node but are not well-connected. Hence, it results in travelling many hops within that region before reaching the destination. The proposed search algorithms avoid the low-connected nodes and reach the destination node in fewer hops.

Table 3. Comparison of search algorithms on the US airline network, the US network without Alaska, and the US mainland network. l is the average path length for the paths found by the search algorithms and c is the number of times the path was not found. The table summarizes the average of l and c obtained for all the possible pairs in the network. In the US airline network, algorithms 6, 7 and 8 give paths close to the shortest path length. In the other two networks, algorithms 5, 6, 7 and 8 give short paths. Here again, the greedy search performs well for the paths found (l) but it is sometimes unable to find a path (c).

	US airline network ($N = 690$, Pairs = 475 410)		US network without Alaska ($N = 455$, Pairs = 206 570)		US mainland network ($N = 427$, Pairs = 181 902)	
	l	c	l	c	l	c
Greedy search	3.93	16806 (3.54%)	2.83	4015 (1.94%)	2.74	3729 (2.05%)
Algorithm 5	5.53	13870 (2.92%)	3.75	456 (0.22%)	2.85	425 (0.23%)
Algorithm 6	4.01	752 (0.16%)	3.17	454 (0.22%)	2.68	425 (0.23%)
Algorithm 7	3.37	688 (0.14%)	2.68	453 (0.22%)	2.93	1 ($\ll 0.01\%$)
Algorithm 8	3.37	41 ($< 0.01\%$)	2.76	38 (0.02%)	2.75	39 (0.02%)
Shortest path length	3.02	NA	2.39	NA	2.32	NA

When we looked at the search results in more detail we found a few more interesting behaviours. The greedy search and algorithm 5 were unable to find paths for approximately the same number of pairs in the US airline network (3.54% in the case of the former and 2.92% for the latter). However, there is a difference in the type of paths these search algorithms could not find. The paths not found by greedy search were distributed uniformly for all departure and destination nodes; the paths not found by algorithm 5 were due predominantly to the 18 airports in Alaska, which were unreachable, almost regardless of the starting point. It was interesting to see that even if we start from Anchorage International Airport (ANC), the most connected airport in Alaska, these airports were not reachable. This is mainly due to the high affinity of algorithm 5 for high-degree nodes. The degree of neighbours of ANC which are in Alaska is small compared to the degree of neighbours on the US mainland. Hence, when we start from an airport, the algorithm was able to reach Anchorage but afterward selected one of the highly-connected airports on the US mainland. From that point on, it is difficult to return to Alaska, since the search algorithm is self-avoiding and since the only other airport that flies to Alaska, excluding ANC, is Seattle-Tacoma International Airport (SEA). The US airline network without Alaska and the US mainland network do not have these constraints, and hence algorithm 5 was able to perform better.

Among the 475 410 pairs of source and destination nodes searched, algorithms 6 and 7 could not reach the destination node 752 and 688 times, respectively. Again, it turns out that the failure to reach the destination was mainly due to a particular airport, namely, Havre City-County Airport (HVR) in Montana. Similar behaviour was observed for these algorithms in the US airline network without Alaska and the US mainland network. HVR is a single-degree node that is connected to Lewistown Airport (LWT), Montana and the only other airport to which LWT is connected is Billings Logan International Airport (BIL), Montana which is a well-connected airport. Hence, the only way to reach HVR would be to reach BIL first and then to fly to LWT. Unfortunately, none of the algorithms, other than the greedy search, can choose LWT from BIL

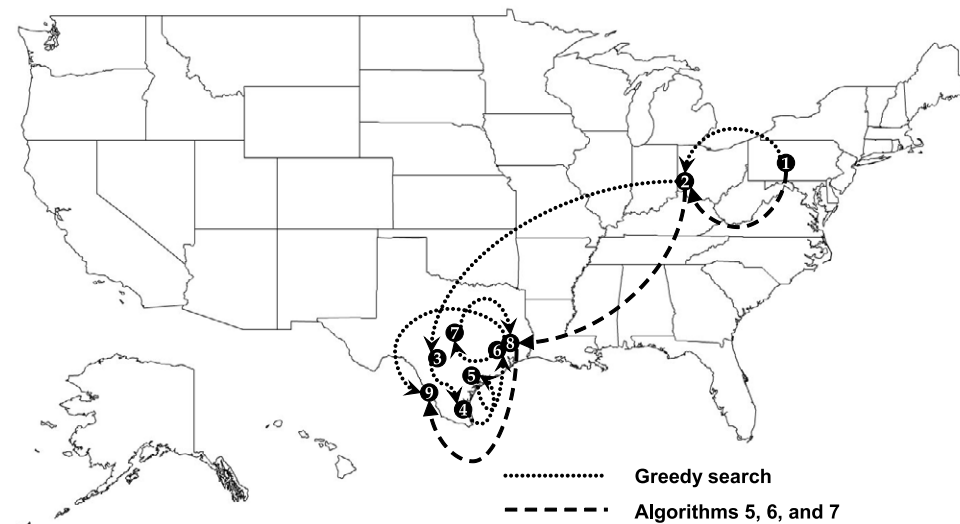


Figure 3. Visualization of the paths obtained in a characteristic case when greedy search takes a higher number of hops. In this case, the departure airport is State College, PA (node 1) and the destination airport is Laredo, Texas (node 9). The airline codes and degrees corresponding to the nodes are: 1, SCE, degree 5; 2, CVG, degree 118; 3, SAT, degree 29; 4, HRL, degree 6; 5, CRP, degree 5; 6, HOU, degree 31; 7, AUS, degree 34; 8, IAH, degree 118; 9, LRD, degree 2. The path obtained for the greedy search is $1 \rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 7 \rightarrow 8 \rightarrow 9$ and for the algorithms 5, 6 and 7 is $1 \rightarrow 2 \rightarrow 8 \rightarrow 9$. Algorithm 8, not shown on the map, takes 4 hops ($1 \rightarrow 2 \rightarrow 3 \rightarrow 8 \rightarrow 9$). Often the greedy search reaches the nodes which are near to the destination node but are not well-connected. Hence, it ends up travelling many hops within that region before it reaches the destination. Whereas, the proposed search algorithms avoid the low-connected nodes and reach the destination node in a lesser number of hops.

when the destination is HVR. Here again, even though the algorithms 5, 6, 7 and 8 are able to reach BIL, they do not choose LWT as the first choice. Moreover, once they fly out of BIL, they take many hops to reach BIL again due to the self-avoiding nature of the algorithms. For instance, when the destination is HVR, algorithms 7 and 8 take, on an average, only 2.5 and 3.44 hops respectively to reach BIL. However, to reach HVR they take around 170 and 102 hops, respectively. The reason why this behaviour is not observed for other single-degree nodes in the US mainland network is that single-degree nodes are usually connected to high-degree nodes. The average degree of the neighbours of the single-degree nodes was found to be 82.86, which is significantly higher than the average degree in the network (12.78). In addition, the only airport (LWT) that flies to HVR (or to a neighbourhood of HVR) is not chosen by the only other airport (BIL) that can fly to LWT.

Table 4 gives the percentage of times the path length found by the search algorithms is the same as the shortest path length. In approximately 90% of the pairs, the path length found by algorithms 6, 7 and 8 was the same as the shortest path length. Further, in 97% of the pairs, the path length found was more than the shortest path by a maximum of two hops. Given that

Table 4. Comparison of search algorithms on the US airline network, the US network without Alaska, and the US mainland network. ‘Diff = 0’ is the percentage of pairs for which the path length found by the search algorithms is the same as the shortest path length. Algorithms 6, 7 and 8 are able find the shortest paths in more than 90% of the pairs. ‘Diff ≤ 2 ’ is the percentage of pairs for which the path length found was more than the shortest path by a maximum of two hops. Given that the search algorithms use only local information, these results on the US airline network are quite fascinating.

	US airline network		US network without Alaska		US mainland network	
	Diff = 0(%)	Diff ≤ 2 (%)	Diff = 0(%)	Diff ≤ 2 (%)	Diff = 0(%)	Diff ≤ 2 (%)
Greedy search	66.3	85.8	75.3	92.3	75.8	92.7
Algorithm 5	66.9	72.1	88.2	93.7	90.8	96.0
Algorithm 6	88.8	96.6	90.8	95.6	92.2	96.8
Algorithm 7	91.3	98.0	92.0	97.6	92.4	98.1
Algorithm 8	88.4	97.5	89.5	97.8	89.0	97.6

the search algorithms use only local information these results on the airline networks are quite fascinating. Note that this behaviour is due mainly to the inherent structure of the US airline network, which can be considered a ‘searchable network’.

6. Conclusions and discussion

In this paper, we studied decentralized search in spatial scale-free networks. We proposed different search algorithms that combine the direction of travel and the degree of the neighbour and illustrated that some of these algorithms can find short paths by using the local information alone. We demonstrated that a family of parameterized spatial network model belongs to a class of searchable networks for a wide range of parameter space. Further, we tested these algorithms on the US airline network. Surprisingly, we found that one can travel from one place to another in fewer than four hops while using only local information. This implies that searchability is a generic property of the US airline network, as is also the case for social networks.

In addition, the spatial network model and the airline network are searchable for a wide range of search algorithms. For example, algorithms 6 and 8 are both able to find short paths in these networks. Hence, any search algorithm with a function $f(x)$ that scales between x and $\ln x$ should give short paths. Moreover, the algorithms can be extended to other power-law networks if we can embed the network in an n -dimensional metric space in which nodes are connected based on the metric distance. The algorithms are relevant to other networks such as the Internet and road networks. As demonstrated in [33], the Internet can be described by the family of spatial network models considered in this paper and hence we expect that these search algorithms can find short paths in the Internet. However, road networks do not follow a power-law degree distribution. Investigating the algorithms on the dual form of the road networks, which do exhibit scale-free properties [40], is a topic of future work.

We notice that algorithm 8, the most conservative with respect to degree, performs the best in the US airline network. This implies that direction plays the most important role in efficient

searching, and even slight blending of direction with degree is sufficient to drastically improve the efficiency of search algorithms. In other words, a search algorithm which traverses based on direction and that cautiously avoids low-degree nodes should give short paths. However, as observed with algorithm 5, sometimes high preference for degree may lead the algorithm to the nodes far away from the destination node. Further, we can conclude that searchability is a property of the network rather than of the functional forms used for the search algorithm.

The difference between the results obtained on the US airline network and the US mainland network is not significant (especially for algorithms 7 and 8). This implies that the results can probably be extended to the WWN [7] which has a very similar structure to the US airline network. In the US airline network, we have separated areas which are connected to the mainland by only a few airports. Algorithms 7 and 8 are able to capture these connections in order to travel from one separated area to another. The WWN will have many more of these separated areas which are well-connected locally but are sparsely inter-connected. We feel that algorithms 7 and 8 would be able to find short paths in the WWN; verification would be subject to the availability of data on the WWN.

Probably, the results obtained for the US airline network are intuitive. For instance, in real life if one is asked to travel with local information, he/she can always find a short path—if not always the shortest path. But the significance of the results lies in capturing this phenomenon/intuition in an algorithm. Definitely, the structure of the network facilitates its searchability. As conjectured by others, the results presented in this paper support the hypothesis [10, 21] that many real-world networks evolve to inherently facilitate decentralized search. Furthermore, these results provide insights for designing the structure of decentralized networks that need effective search algorithms.

Acknowledgments

The authors would like to acknowledge the National Science Foundation (grants DMI 0537992 and CCF 0643529) for making this work feasible. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

- [1] Albert R and Barabási A L 2002 *Rev. Mod. Phys.* **74** 47
- [2] Boccaletti S, Latora V, Moreno Y, Chavez M and Hwang D U 2006 *Phys. Rep.* **424** 175
- [3] Dorogovtsev S N and Mendes J F F 2002 *Adv. Phys.* **51** 1079
- [4] Newman M E J 2003 *SIAM Rev.* **45** 167
- [5] Watts D J and Strogatz S H 1998 *Nature* **393** 440
- [6] Ravasz E, Somera A L, Mongru D A, Oltvai Z N and Barabási A L 2002 *Science* **297** 1551
- [7] Guimera R, Mossa S, Turtschi A and Amaral L A N 2005 *Proc. Natl Acad. Sci.* **102** 7794
- [8] Albert R, Jeong H and Barabási A L 2000 *Nature* **406** 378
- [9] Thadakamalla H P, Raghavan U N, Kumara S R T and Albert R 2004 *IEEE Intell. Syst.* **19** 24
- [10] Adamic L A, Lukose R M, Puniyani A R and Huberman B A 2001 *Phys. Rev. E* **64** 046135
- [11] Thadakamalla H P, Albert R and Kumara S R T 2005 *Phys. Rev. E* **72** 066128
- [12] Pastor-Satorras R and Vespignani A 2001 *Phys. Rev. Lett.* **86** 3200
- [13] Milgram S 1967 *Psychol. Today* **2** 60
- [14] Kleinberg J 2000 *Nature* **406** 845

- [15] Kleinberg J 2000 *Proc. 32nd ACM Symp. Theor. Comput.* pp 163–70
- [16] Kleinberg J 2001 *Adv. Neural Inform. Process. Syst.* **14** 431
- [17] Dodds P, Muhamad R and Watts D J 2003 *Science* **301** 827
- [18] Watts D J, Dodds P S and Newman M E J 2002 *Science* **296** 1302
- [19] Kim B J, Yoon C N, Han S K and Jeong H 2002 *Phys. Rev. E* **65** 027103
- [20] Arenas A, Cabrales A, Diaz-Guilera A, Guimera R and Vega F 2003 *Statistical mechanics of complex networks* (Berlin: Springer) chapter ‘Search and Congestion in Complex Networks’ pp 175–94
- [21] Kleinberg J 2006 *Proc. Int. Cong. Math.* **3** 1019
- [22] Liben-Nowell D, Novak J, Kumar R, Raghavan P and Tomkins A 2005 *Proc. Natl Acad. Sci.* **102** 11623
- [23] Menczer F 2002 *Proc. Natl Acad. Sci.* **99** 14014
- [24] Sandberg O 2006 *Proc. 8th Workshop on Algorithm engineering and experiments (ALENEX)* pp 144–55
- [25] Simsek O and Jensen D 2005 *Proc. 19th Int. Joint Conf. Artificial Intell.* pp 304–10
- [26] Zhang H, Goel A and Govindan R 2004 *Comput. Netw.* **46** 555
- [27] Akyildiz I F, Su W, Sankarasubramaniam Y and Cayirci E 2002 *Comput. Netw.* **38** 393
- [28] Raghavan U N and Kumara S R T 2007 *Int. J. Sensor Netw.* **2** 201
- [29] Kan G 2001 *Peer-to-Peer Harnessing the Power of Disruptive Technologies* (Beijing: O’Reilly) chapter ‘Gnutella’
- [30] Chakrabarti S, van den Berg M and Dom B 1999 *Comput. Netw.* **31** 1623
- [31] Clauset A and Moore C 2003 *Preprint cond-mat/0309415*
- [32] Barabási A L and Albert R 1999 *Science* **286** 509
- [33] Yook S H, Jeong H and Barabási A L 2002 *Proc. Natl Acad. Sci.* **99** 13382
- [34] Guimera R and Amaral L A N 2004 *Eur. Phys. J. B* **38** 381
- [35] Newman M E J, Strogatz S H and Watts D J 2001 *Phys. Rev. E* **64** 026118
- [36] Dorogovtsev S and Mendes J F F 2000 *Europhys. Lett.* **52** 33
- [37] Barthélemy M 2003 *Europhys. Lett.* **63** 915
- [38] Cormen T H, Leiserson C E, Rivest R L and Stein C 2001 *Introduction to Algorithms* 2nd edn (Cambridge: MIT Press)
- [39] The Bureau of Transportation Statistics online at <http://www.transtats.bts.gov/> (date accessed: 20 July 2006)
- [40] Kalapala V, Sanwalani V, Clauset A and Moore C 2006 *Phys. Rev. E* **73** 026130