We study the decentralized search problem in a family of parameterized spatial network models that are heterogenous 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 U.S. airline network which belongs to this class of networks and demonstrate that searchability is a generic property of the U.S. airline network. These results provide insights on designing the structure of distributed networks that need effective decentralized search algorithms.

I. 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 heterogenous 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 heterogenous, 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 heterogenous [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 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 search 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 be 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 search 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 heterogenous 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 neighbor. 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 U.S.
airline network which belongs to this class of networks and show that searchability is a generic property of the U.S. airline network.

II. LITERATURE AND PROBLEM DESCRIPTION

Decentralized search in networks can be broadly classified into search 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 high-degree seeking search is better than random-walk search. In random-walk search, the node that has the message passes it to a randomly chosen neighbor, and the process continues until it reaches the target node. Whereas, in high-degree search, the node that has the message passes it to the neighbor 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 heterogenous 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 neighbor 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 search 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 heterogenous 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 neighbors, and the degree of its neighbors. Using this information, the start node, and consecutively each node receiving the message, passes the message to one of its neighbors 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 traveled 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 neighbor is important and propose different ways of composing the direction of travel and the degree of the neighbor.

III. DECENTRALIZED SEARCH ALGORITHMS

A simple search algorithm in spatial networks is greedy search, where each node passes the message to the neighbor closest to the target node. Let $d_i$ be the distance to the target node from each neighbor $i$ (see figure 1(a)) and let $k_i$ be the degree of the neighbor $i$. Greedy search
chooses the neighbor with the smallest $d_i$. This will ensure that the message is always going to the neighbor 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 neighbor 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 neighbor 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 $Y_k = \text{Max}(X_1, X_2, X_3, \ldots, X_k)$, where $X_1, X_2, X_3, \ldots, X_k$ 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 $P(X_1 \leq y) \leq 1 \ \forall y,$

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

implies that

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

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

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

Now consider two neighbors $n_1$ and $n_2$ with degree $k_1$ and $k_2$. The expected distance the neighbors $n_1$ 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 $X_1, X_2, X_3, \ldots, X_k$ are independent which is valid when the number of edges is large. Hence, if we choose a neighbor 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 neighbor, $k_i$, 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 neighbor 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 neighbor.

2. **High degree search**: The node passes the message to the neighbor with the highest degree. The idea here is that by choosing a neighbor 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 neighbor $i$ with the smallest $d_i$. This will ensure that the message is always going to the neighbor closest to the target node.

FIG. 1: (a) Illustration of a spatial network. $d_i$ is the distance to the target node from each neighbor $i$ and $k_i$ is the degree of the neighbor $i$. (b) Illustration for demonstrating that sometimes it is better to choose a neighbor 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.
4. **Algorithm 4**: The node passes the message to the neighbor \(i\) with the smallest measure \(d_i - g(k_i)\). The function \(g(k)\) is an incentive for choosing a neighbor of higher degree. Ideally, \(g(k)\) 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 neighbor \(i\) that has the smallest measure \(\left(\frac{d_i}{d_m}\right)^k\), 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 neighbor that has lower \(d_i\) and higher \(k_i\).

6. **Algorithm 6**: The node passes the message to the neighbor \(i\) that has the smallest measure \(d_i\). Here, again, the algorithm prefers the neighbor that has lower \(d_i\) and higher \(k_i\).

7. **Algorithm 7**: The node passes the message to the neighbor \(i\) that has the smallest measure \(\left(\frac{d_i}{d_m}\right)^{\ln k_i + 1}\). This is a conservative version of algorithm 5 with respect to \(k_i\).

8. **Algorithm 8**: The node passes the message to the neighbor \(i\) 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 neighbors’ degree. Thus, we expect these algorithms to give smaller path lengths than other algorithms. In 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 neighbor. 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.

### IV. 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]:

- with probability \(p\), a new edge is created between two existing nodes in the network;
- 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

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

where \(k_i\) is the degree of node \(i\), \(d_{ij}\) is the Euclidian distance between node \(i\) and node \(j\) 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

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

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 \(d^\alpha\) [33] and \exp(d/d_{char}) [37].

### A. 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 neighbors, according to the search algorithm. For algorithm 4, we assume the incentive function \(g(k)\) 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 considered space. We assume that it is sufficient if the message reaches a small neighborhood 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 neighborhood of a destination city (especially when the city has multiple airports). The search process continues...
until the message reaches a neighbor of the target node or a node within a circle of radius \(D = 50\) centered around the target node. In order to avoid passing the message to a neighbor that has already received the message, a list \(L\) is maintained. During the search process, if the message reaches a node \(i\) whose neighbors are all in the list \(L\), then the message is passed to one of the neighbors 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 neighbor \(i\) 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 I compares the performance of different search algorithms for the spatial network, \(G(1000, 0.72, 1, d^*, 1)\) with \(r = 1, 2, \text{ and } 3\). We find that the decentralized search algorithms 5, 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 behavior 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 I). 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 I) but for a few node pairs it is unable to find a path (compare \(c^*\)’s in Table I). 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 II 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 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.

V. SEARCH IN THE U.S. AIRLINE NETWORK

Let us consider the U.S. 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 U.S. 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 neighboring 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
TABLE I: 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, \( d_{\text{path}} \) is the shortest 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 \( l \), \( d_{\text{path}} \) 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 \( d_{\text{path}} \)), it is sometimes unable to find a path (\( c \)).

<table>
<thead>
<tr>
<th>Search Algorithm</th>
<th>( r = 1 )</th>
<th>( r = 2 )</th>
<th>( r = 3 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( l )</td>
<td>( d_{\text{path}} )</td>
<td>( c ) (%)</td>
<td>( l )</td>
</tr>
<tr>
<td>Random walk</td>
<td>41.68</td>
<td>10957</td>
<td>0</td>
</tr>
<tr>
<td>High-degree search</td>
<td>28.35</td>
<td>8032</td>
<td>0</td>
</tr>
<tr>
<td>Greedy search</td>
<td>3.37</td>
<td>787</td>
<td>0.17</td>
</tr>
<tr>
<td>Algorithm 4</td>
<td>10.22</td>
<td>2303</td>
<td>0.12</td>
</tr>
<tr>
<td>Algorithm 5</td>
<td>2.47</td>
<td>646</td>
<td>0</td>
</tr>
<tr>
<td>Algorithm 6</td>
<td>2.45</td>
<td>636</td>
<td>0</td>
</tr>
<tr>
<td>Algorithm 7</td>
<td>2.54</td>
<td>631</td>
<td>0</td>
</tr>
<tr>
<td>Algorithm 8</td>
<td>2.66</td>
<td>646</td>
<td>0</td>
</tr>
<tr>
<td>Shortest path length</td>
<td>2.27</td>
<td>531</td>
<td>NA</td>
</tr>
</tbody>
</table>

TABLE II: 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 \( N \), \( p \), and \( r \). 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 (\( c \)).

<table>
<thead>
<tr>
<th>( N = 1000 ), ( r = 1 )</th>
<th>( p = 0.30 )</th>
<th>( p = 0.80 )</th>
<th>( N = 500 )</th>
<th>( N = 1500 )</th>
<th>( d_{\text{char}} = 0.5 )</th>
<th>( d_{\text{char}} = 2.0 )</th>
</tr>
</thead>
<tbody>
<tr>
<td>( l )</td>
<td>( c ) (%)</td>
<td>( l )</td>
<td>( c ) (%)</td>
<td>( l )</td>
<td>( c ) (%)</td>
<td>( l )</td>
</tr>
<tr>
<td>Greedy search</td>
<td>6.55</td>
<td>7.93</td>
<td>2.90</td>
<td>0.09</td>
<td>4.09</td>
<td>0.24</td>
</tr>
<tr>
<td>Algorithm 5</td>
<td>3.41</td>
<td>0.02</td>
<td>2.35</td>
<td>0</td>
<td>2.83</td>
<td>0</td>
</tr>
<tr>
<td>Algorithm 6</td>
<td>3.38</td>
<td>0.04</td>
<td>2.38</td>
<td>0</td>
<td>2.81</td>
<td>0</td>
</tr>
<tr>
<td>Algorithm 7</td>
<td>3.59</td>
<td>0.19</td>
<td>2.40</td>
<td>0</td>
<td>2.95</td>
<td>0</td>
</tr>
<tr>
<td>Algorithm 8</td>
<td>4.12</td>
<td>0.73</td>
<td>2.49 &lt; 0.01</td>
<td>3.16</td>
<td>&lt; 0.01</td>
<td>2.54</td>
</tr>
<tr>
<td>Shortest path length</td>
<td>2.91</td>
<td>NA</td>
<td>2.16</td>
<td>NA</td>
<td>2.30</td>
<td>NA</td>
</tr>
</tbody>
</table>

airport, we choose to fly to one of its neighbors based on the degree of the neighboring 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 neighborhood 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.

A. Properties of the U.S. 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 United States of America. We considered the data collected for the service class F (scheduled passenger service) flights during the month of January 2006 to form the U.S. 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 un-directed.
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 U.S. 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 U.S. 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 \( \pm 0.1 \) [7]. Further, as observed in 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 \( i \), \( b_i / < b > \), where \( < b > \) is the average BC of the network, versus its scaled degree \( k_i / < k > \), where \( < k > \) is the average degree of the network. The geopolitical considerations used to explain this phenomenon in the WWN [34] do not apply to the U.S. airline network, as it belongs to a single country. In fact, this behavior 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 U.S. 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 U.S. airline network, the U.S. airline network without Alaska, and the U.S. mainland airline network without Alaska, Hawaii, Puerto Rico, the U.S. Virgin Islands, and the U.S. Pacific Trust Territories and Possessions (U.S. mainland network). The latter two networks have statistical properties similar to those of the U.S. airline network. The U.S. airline network without Alaska has 459 nodes and 2857 edges with 455 nodes and 2856 edges in the LCC; the U.S. mainland network has 431 nodes and 2729 edges with 427 nodes and 2728 edges in the LCC.

B. Search results and analysis

We simulated the search algorithms for all \( N \times (N-1) \) pairs in each network, where \( N \) is the number of nodes. The U.S. airline network, the U.S. airline network without Alaska, and the U.S. mainland network had 475410, 206570, and 181902 pairs, respectively. We chose \( d_m \) to be the largest distance between two airports in the network and the neighborhood distance \( D \) to be 100 miles. Table III 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 U.S. mainland network, the performance of the search algorithms is even better especially for algorithm 5, which did not perform well for the complete U.S. airline network. Figure 3 visualizes the paths ob-
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 behavior is not observed for other
single-degree nodes in the U.S. mainland network is that single-degree nodes are usually connected to high degree
nodes. The average degree of the neighbors 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 neighborhood of HVR) is not chosen by the only other airport (BIL) that can fly to LWT.

Table IV 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 the search algorithms use only local information these results on the
airline networks are quite fascinating. Note that this behavior is due mainly to the inherent structure of the U.S.
airline network, which can be considered a “searchable network”.

VI. CONCLUSIONS AND DISCUSSION

In this paper, we studied decentralized search in spatial scale-free networks. We proposed different search algo-
rithms that combine the direction of travel and the degree of the neighbor and illustrated that some of these algo-
rithms 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 U.S. 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 U.S. 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 algo-
rithms. For example, algorithms 6 and 8 are able to find short paths in these networks. Hence, any search algo-
rithm 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 In-
ternet 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 Inter-
net. 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.
FIG. 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 traveling 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.

TABLE III: Comparison of search algorithms on the U.S. airline network, the U.S. network without Alaska, and the U.S. 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 U.S. 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 \((c)\).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(l)</td>
<td>(c)</td>
<td>(l)</td>
</tr>
<tr>
<td>Greedy search</td>
<td>3.93</td>
<td>16806 (3.54%)</td>
<td>2.83</td>
</tr>
<tr>
<td>Algorithm 5</td>
<td>5.53</td>
<td>13870 (2.92%)</td>
<td>3.75</td>
</tr>
<tr>
<td>Algorithm 6</td>
<td>4.01</td>
<td>752 (0.16%)</td>
<td>3.17</td>
</tr>
<tr>
<td>Algorithm 7</td>
<td>3.37</td>
<td>688 (0.14%)</td>
<td>2.68</td>
</tr>
<tr>
<td>Algorithm 8</td>
<td>3.37</td>
<td>41 (&lt; 0.01%)</td>
<td>2.76</td>
</tr>
<tr>
<td>Shortest path length</td>
<td>3.02</td>
<td>NA</td>
<td>2.39</td>
</tr>
</tbody>
</table>

We notice that algorithm 8, the most conservative with respect to degree, performs the best in the U.S. airline network. This implies that direction plays the most important role in efficient searchability, 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 U.S. airline network and the U.S. mainland network is not significant (especially for algorithms 7 and 8). This implies that the results can probably be extended to the world-
phenomenon/intuition in an algorithm. Definitely, the
ways find a short path – if not always the shortest path.
work are intuitive. For instance, in real life if one is
availability of data on the WWN.
We feel that algorithms 7 and 8 would be able to find
will have many more of these separated areas which are
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
tavel from one separated area to another. The WWN
structure to the U.S. airline network. In the U.S. airline
network, the U.S. network without Alaska [7] which has a very similar
fascinating.

TABLE IV: Comparison of search algorithms on the U.S. airline network, the U.S. network without Alaska, and the U.S. mainland network. “Diff = 0” is the percentage number 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 number 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 U.S. airline network are quite

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>U.S. airline network</th>
<th>U.S. network without Alaska</th>
<th>U.S. mainland network</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Diff = 0 (%)</td>
<td>Diff = 0 (%)</td>
<td>Diff = 0 (%)</td>
</tr>
<tr>
<td>Greedy search</td>
<td>66.3</td>
<td>75.3</td>
<td>92.3</td>
</tr>
<tr>
<td>Algorithm 5</td>
<td>66.9</td>
<td>88.2</td>
<td>93.7</td>
</tr>
<tr>
<td>Algorithm 6</td>
<td>88.8</td>
<td>95.6</td>
<td>92.2</td>
</tr>
<tr>
<td>Algorithm 7</td>
<td>91.3</td>
<td>97.6</td>
<td>92.4</td>
</tr>
<tr>
<td>Algorithm 8</td>
<td>88.4</td>
<td>97.5</td>
<td>97.6</td>
</tr>
</tbody>
</table>

wide airline network (WWN) [7] which has a very similar structure to the U.S. airline network. In the U.S. 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 WWN; verification would be subject to the availability of data on the WWN.

Probably, the results obtained for the U.S. 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.

[24] O. Sandberg, Algorithm engineering and experiments


