A Spectrum of Time-Space Tradeoffs for Undirected s-t Connectivity

Uriel Feige *
Department of Applied Mathematics
The Weizmann Institute
Rehovot 76100, Israel.
feige@wisdom.weizmann.ac.il.

June 27, 1996

Abstract

We present a family of randomized algorithms that enjoys a wide range of time-space tradeoffs in deciding undirected $\mathcal{S}$-$\mathcal{T}$-connectivity. Our tradeoffs cover the whole range between Breadth First Search and the random walk procedure of Aleliunas *et al.*, and achieve a time-space product of $\tilde{O}(mn)$ (where n is the number of vertices in the graph, m is the number of edges, and $\tilde{O}$ notation is used in order to suppress logarithmic terms). Moreover, we obtain improved time-space tradeoffs of $\tilde{O}(n^2)$ for regular graphs. A convenient and informative way of expressing our tradeoffs, that implies the tradeoffs stated above, is as $\tilde{O}((\sum_{i=1}^{n} d_i)(\sum_{i=1}^{n} 1/d_i))$, where d_i is the degree of vertex i in the input graph.

In constructing our algorithms and analysing them, we build upon earlier work of Broder *et al.* (who achieved a time-space tradeoff of $\tilde{O}(m^2)$), Barnes and Feige (who achieved a time-space tradeoff of $\tilde{O}(m^{3/2}n^{1/2})$), and Aldous. In passing, we also improve previous results regarding the rate at which a random walk discovers new vertices in a graph.

1 Introduction

Let G be an undirected simple graph with n vertices, m edges. We consider the problem of determining if there is a path in G connecting two distinguished vertices, $\mathcal{S}$ and $\mathcal{T}$. This problem is known as *undirected $\mathcal{S}$-$\mathcal{T}$-connectivity*, or USTCON. It can be solved in time

*Incumbent of the Joseph and Celia Reskin Career Development Chair. Yigal Alon Fellow.

$\tilde{O}(m)$ and space $\tilde{O}(n)$ deterministically, by *breadth first search* (BFS). (Throughout, $\tilde{O}$ and $\tilde{\Omega}$ notation is used in order to suppress polylogarithmic terms. E.g., $n \log n$ is $\tilde{O}(n)$. We also use the convention that suppressed terms are polylogarithmic in n, regardless of the explicit terms that appear in $\tilde{O}(.)$. Hence $\log n$ is $\tilde{O}(1)$, and $p \log n$ is $\tilde{O}(p)$.) Aleliunas *et al.*[AKLLR] showed that a randomized algorithm can solve USTCON in space $O(\log n)$. Their algorithm performs a random walk on the vertices of G, where at each time step the walk moves to a vertex chosen uniformly at random from the neighbors of the current vertex. Aleliunas *et al.*[AKLLR] prove that if $\mathcal{S}$ and $\mathcal{T}$ are connected, then a random walk that starts at $\mathcal{S}$ is expected to reach $\mathcal{T}$ in mn steps. This leads to a randomized algorithm with one sided error: start a walk at $\mathcal{S}$. If $\mathcal{T}$ is reached within $3mn$ steps, declare that $\mathcal{S}$ and $\mathcal{T}$ are connected. Otherwise – not connected. The error probability can be made arbitrarily small by independent repetition of the algorithm. (A logspace "Las Vegas" randomized algorithm with no error was subsequently designed in [BCDRT].)

It is apparent that the random walk algorithm is wasteful in terms of time, as each edge is visited on average n times. Can the time complexity of the algorithm be improved, at the price of only a modest increase in the space complexity? Aleliunas *et al.*[AKLLR] asked whether there is a spectrum of tradeoffs between time and space for randomized algorithms for USTCON. Observe that both BFS and the random walk algorithm achieve a time space tradeoff of $ST = \tilde{O}(mn)$ (where T denotes time and S denotes space). Is there a randomized algorithm that for any space bound $\log n < S < n$, requires time $T = \tilde{O}(mn/S)$? In this paper we answer this question in the affirmative.

1.1 Previous related work

Broder *et al.* [BKRU] were the first to develop an algorithm that achieves a nontrivial tradeoff. Their algorithm is based on the following observation: the fraction of "wasted" steps of the random walk (steps in which the walk revisits previously visited vertices) increases as the walk grows longer. Hence many short random walks are more efficient in exploring a graph than one long random walk.

We now give a high level overview of the algorithm of Broder *et al.*. Some implementation details are left out, and can be found in [BKRU]. The algorithm that is developed in the current paper is obtained via simple modifications of [BKRU]'s algorithm. The overview is presented in a way that will simplify subsequent presentation of modifications to the algorithm.

Input to the algorithm: A graph G with n vertices and m edges, two distinguished vertices, $\mathcal{S}$ and $\mathcal{T}$. In addition, a parameter p is specified, $1 \leq p \leq n$, indicating that the algorithm can use $O(p \log n)$ space.

Properties of the algorithm: Decides whether $\mathcal{S}$ and $\mathcal{T}$ are in the same connected component. The algorithm is randomized, and may have a small (e.g., less that 1/3) probability of error. The algorithm runs in time $\tilde{O}(m^2/p)$, giving a time-space product of $\tilde{O}(m^2)$.

Setting up parameters: The algorithm fixes a distribution D on the vertices of G. This distribution must be simple enough so that the time and space complexity of sampling according to D will have only negligible effect on the total running time and space requirements of the algorithm. Broder *et al.* choose D to be the stationary distribution SD for random walks. That is, the probability that vertex i is selected under distribution SD is $d_i/2m$, where d_i denotes the degree of vertex i in G. In addition, a parameter t is selected. Broder *et al.* select t satisfying $t = \tilde{O}(m^2/p^2)$.

Distributing landmarks: The algorithm decides randomly on $p + 2$ vertices to be denoted as *landmarks* in G. Two of these landmarks are $\mathcal{S}$ and $\mathcal{T}$. The other landmarks are chosen independently at random according to distribution D. The landmarks are not required to be distinct – the same vertex may serve as several landmarks.

Short random walks: From each landmark, a pebble takes $\Theta(\log n)$ random walks. Each random walk continues for t steps. If a random walk that starts at landmark A hits landmark B then the algorithm notes that the two landmarks are in the same connected component of G. More generally, the algorithm notes that there is one connected component that contains the landmarks that were previously determined to be in the connected component of landmark A and the landmarks that were previously determined to be in the connected component of landmark B.

Termination: The output of the short random walks phase is a partition of the set of landmarks into subsets, such that two landmarks are in the same subset only if they are in the same connected component of G. If the landmarks of $\mathcal{S}$ and $\mathcal{T}$ belong to the same subset, the algorithm accepts. If not, the algorithm rejects. In the latter case, the output of the algorithm might be incorrect. (Due to unlucky coin tosses, the algorithm may have failed to determine that $\mathcal{S}$ and $\mathcal{T}$ belong to the same component when in fact they do.) To reduce the probability of error below ϵ, the main part of the algorithm (distributing the landmarks and taking short random walks) can be repeated $O(\log 1/\epsilon)$ times with independent coin tosses, rejecting only if all executions reject.

Broder *et al.* show that if $\mathcal{S}$ and $\mathcal{T}$ are connected, the algorithm will discover this fact with high probability. Hence the algorithm correctly decides USTCON. The space requirements of the algorithm are shown to be $\tilde{O}(p)$ as follows. It takes space $p \log n$ to remember the names of all landmarks. In addition, $\log n$ bits suffice in order to record the current location of the pebble that takes a random walk. Broder *et al.* show how other operations of the algorithm, such as performing unions of sets of landmarks when it is noted that they are in the same connected component, or sampling from the distribution D, can all be performed in space $\tilde{O}(p)$, under certain reasonable assumptions about the model of computation and encoding of the input graph. The analysis of the time complexity of the algorithm is based on counting the total number of random walk steps made by the algorithm, which is $O(pt\log n)$, giving $\tilde{O}(m^2/p)$ when $t = \tilde{O}(m^2/p^2)$. Similar to the space analysis, Broder *et al.* show that the time of performing all other operations is dominated by the number of random walk steps. This establishes a time-space product of $\tilde{O}(pm^2/p) = \tilde{O}(m^2)$.

For sparse graphs (where $m = O(n)$), the [BKRU] algorithm achieves a smooth tradeoff between BFS and the random walk algorithm. However, for dense graphs, the quality of the [BKRU] tradeoff deteriorates.

It was observed on toy examples that landmark distribution schemes other than SD (stationary distribution) lead to improved performance. However, in order to rigorously prove that these schemes obtain a better tradeoff, further improvement in the analysis of random walks was necessary. Barnes and Feige [BF] analysed short random walks, and used this analysis to obtain partial results on MD, the "mixed distribution" scheme (this scheme was suggested by Ruzzo [ruzzo]). MD scatters $p/2$ landmarks at random similarly to SD, and $p/2$ additional landmarks uniformly at random over the vertices. Barnes and Feige showed various tradeoffs for the modified algorithm MD, including $ST = \tilde{O}(m^{3/2}n^{1/2})$. Barnes and Feige conjectured that the MD algorithm achieves a time space tradeoff of $ST = \tilde{O}(mn/d_{min})$, where d_{min} is the minimum degree in the graph. This conjectured tradeoff of $\tilde{O}(mn/d_{min})$ for MD is patterned after the similar bound of Kahn *et al.* [KLNS] on the cover time of graphs. In particular, its value is $\tilde{O}(n^2)$ for regular graphs.

1.2 Our results

We suggest a new landmark distribution scheme, ID (*inverse distribution*). ID distributes $p/2$ landmarks according to the stationary distribution (probability that v receives a landmark is linear in d_v), and $p/2$ landmarks according to an "inverse" distribution (probability that v receives a landmark is linear in $1/d_v$). Thus vertices of relatively high degree or relatively low degree are more likely to receive landmarks than vertices of average degree.

The three algorithms SD (stationary distribution, presented in [BKRU]), MD (mixed distribution, presented in [BF]), and ID (inverse distribution, to be described in Section 2), are all patterned after [BKRU]'s SD algorithm. They differ only in the phase of setting up parameters, namely, in the choice of probability distribution D, and in the length t of the short random walks. The analysis of each new algorithm builds upon the analysis of previous ones. In each algorithm, the time space tradeoff achieved depends on t, the length of the individual random walks that the pebble takes. For each distribution scheme, we seek to minimize t as a function of p so as to achieve the best possible tradeoff curve, which satisfies $ST = \tilde{O}(p^2t)$.

Before stating our main theorem, we introduce the *virtual resistance* of a graph, defined as $\hat{R} = \sum_{v\in V} 1/d_v$. Note that $\hat{R} \le n/d_{min}$. The main theorem of the paper is:

Theorem 1 *Algorithm ID achieves a time space tradeoff of* $ST = \tilde{O}(m\hat{R})$*, where* $\hat{R} = \sum_v 1/d_v$*. More explicitly, for any* T *in the range* $m \leq T \leq m\hat{R}$*, algorithm ID can work in time* $\tilde{O}(T)$ *and space* $\tilde{O}(m\hat{R}/T)$*. For any* S *in the range* $1 \leq S \leq \hat{R}$*, algorithm ID can work in space* $\tilde{O}(S)$ *and time* $\tilde{O}(m\hat{R}/S)$*.*

The tradeoff that we prove for ID, $ST = \tilde{O}(m\hat{R})$, can also be expressed elegantly as $ST = \tilde{O}((\sum_{v \in V} d_v)(\sum_{v \in V} 1/d_v))$.

We also improve the known tradeoffs for the previously suggested algorithms, in particular, proving the conjecture of [BF] regarding MD.

Theorem 2 *Algorithm SD achieves a time space tradeoff of* $ST = \tilde{O}((m/d_{min})^2)$*. Algorithm MD achieves a time space tradeoff of* $ST = \tilde{O}(mn/d_{min})$*. Neither algorithm achieves the tradeoff* $ST = \tilde{O}(m\hat{R})$ *achieved by the ID algorithm.*

Observe that both the tradeoff guarantee achieved by ID and the tradeoff guarantee achieved by MD improve over $\tilde{O}(mn)$ (and hence also over BFS!). In particular, for d-regular graphs, we obtain a randomized algorithm that decides USTCON in time $\tilde{O}(m)$ and space $\tilde{O}(n/d)$. The $\tilde{O}(m\hat{R})$ guarantee on the tradeoff achieved by ID is clearly stronger than the $\tilde{O}(mn/d_{min})$ guarantee on the tradeoff achieved by MD. Moreover, it is more "robust" – a minor change in the input graph (splitting an edge into two and placing a vertex in the middle) can change d_{min} (and hence the latter guarantee) by a factor of $\Omega(n)$, whereas such large changes are not possible for the former guarantee.

Algorithms MD and ID differ from the Broder *et al.*algorithm only in the way in which they distribute landmarks on the vertices of the graph. All other details of the algorithms are identical. The improved tradeoffs that we get are a result of more sophisticated landmark distribution schemes, coupled with tighter analysis. Can the landmark distribution scheme be further modified in a way that improves the $\tilde{O}(m\hat{R})$ tradeoff? Before attempting to answer this question, we should ask ourselves which other parameters of the graph may be relevant to random walks.

The value $O(m\hat{R})$ is an upper bound on the expected time that it takes a random walk that starts at $\mathcal{S}$ to hit $\mathcal{T}$, if they are connected (see Corollary 8). Better upper bounds on the hitting time are mD, where D is the *diameter* of the graph, and mR, where R is the *effective resistance* of the graph (definitions will follow). Is it possible to achieve time-space tradeoffs of $\tilde{O}(mD)$, or $\tilde{O}(mR)$, for USTCON? This question should be refined, to allow for the possibility that the input graph is not connected, in which D (and R) is infinite. One way to overcome this problem is to consider the diameter of the connected component of $\mathcal{S}$ or of $\mathcal{T}$. Another possibility is to consider the *aggregate diameter* $\bar{D}$ (the sum of diameters of connected components), or the *aggregate resistance* $\bar{R}$ (the sum of resistances of connected components) of the graph. Note that an upper bound on the expected hitting time does not necessarily lead to a similar bound on random-walk based algorithms for USTCON. If this upper bound is not easily computable (e.g., in space $\log n$ and time proportional to the bound itself), then one may not know when to abandon the random-walk process and decide that $\mathcal{S}$ and $\mathcal{T}$ are not connected. It is not known whether $\bar{D}$ or $\bar{R}$ can be computed in randomized logspace, or whether USTCON can be decided in randomized log-space and time $O(m\bar{D})$ (or $O(m\bar{R})$). Nevertheless, let us assume that $\bar{D}$ is given explicitly as input to our algorithm, and ask whether USTCON has a $\tilde{O}(m\bar{D})$ time-space tradeoff. We consider the class of algorithms FD, *fair distribution*, that is a generalization of SD, MD, and ID. We give a bound on the extent of improvement that is possible if one uses an algorithm from the class FD.

Theorem 3 *No algorithm of the class FD can achieve a time space tradeoff of $\tilde{O}(m\bar{D})$, even if $\bar{D}$ (the* aggregate diameter*) is given explicitly as input to the algorithm. More explicitly, there is a family of graphs with $\bar{D} = \Theta(\sqrt{n})$ for which even if space n^δ is used by an FD algorithm, where $\delta < 1/2$, the time requirement remains $\tilde{\Omega}(m\bar{D})$.*

The above lower bound result is proved only for a very restricted class of algorithms. In particular, the pebble in our algorithm is forced to take random walks, and is not allowed to bias them in any way. Others have studied the issue of lower bounds for time-space tradeoffs for USTCON in a more general (yet still structured) setting. The models on which lower bounds are proved are known as JAG models ("Jumping Automata on Graphs"), and there are several versions of these models. In [BBRRT], a lower bound of $\Omega(n^2)$ is proved for a model that allows for an initial phase of distributing landmarks, but requires that the pebble takes deterministic walks on the graph, rather than random walks. Lower bounds on other variants of the JAG model are presented in [BBRRT, edmonds]. In particular, Edmonds [edmonds] allows for randomized algorithms, and proves that when the number of pebbles is sublogarithmic, the time to decide USTCON is superlinear (on the JAG model of computation).

The main part of our paper is concerned with the analysis of the short term behavior of random walks, as this governs the tradeoffs that we can guarantee for our algorithms (theorems 1 and 2). The results of this analysis may be of independent interest. In particular, we improve upon some of the results obtained in [BF] regarding the rate at which a random walk discovers new vertices in a graph. Let $E[T(\mathcal{N})]$ denote the expected time until a walk discovers its $\mathcal{N}$th distinct vertex, where $\mathcal{N} \leq n$.

Theorem 4 *For any simple connected graph on n vertices with maximum degree d_{max} and minimum degree d_{min}, and for $\mathcal{N} \leq n$,*

$$E[T(\mathcal{N})] = O(\mathcal{N} + \frac{\mathcal{N}^2}{d_{min}} \min[\mathcal{N}, d_{max}])$$

This improves a result of [BF] by a factor of $\log \mathcal{N}$.

1.3 Highlight of main ideas in our analysis

We emphasize four main points in our analysis.

The first point is to identify the key property of short random walks that one has to concentrate on, if one wants to get improved tradeoffs. By carefully considering the work of [BKRU], it becomes apparent that this property is $E_e[N_e(t)]$, which denotes the number of times that a random walk of length t starting at edge e is expected to return to e. The lower the bound on $E_e[N_e(t)]$, the better the tradeoff achieved. It is important to realize that the upper bound of $E_e[N_e(t)] = O(\sqrt{t \log n})$ [BKRU] can be improved, *despite the fact that it is tight* (upto the $\log n$ factor), by introducing new parameters, such as d_{min}.

The second point is a crucial observation made in [BF], which opened the way to analysis of the MD distribution scheme. The observation is that the MD scheme on G can be analysed by considering the SD scheme on a modified graph H. The difference between H and G is that multiple self loops are introduced in H, thus artificially lifting the minimum degree. Since minimum degree is relevant to the short term behavior of random walks, and to $E_e[N_e(t)]$ in particular, one gets better tradeoffs.

The third point is a method of analysis of $E_e[N_e(t)]$. The method used in [BKRU] is not sensative to parameters such as d_{min}. However, one can use instead a method developed by Aldous [aldous], in his proof that for visiting vertices on regular graphs, $E_v[N_v(t)] = O(\sqrt{t})$. The proof technique of Aldous can be applied to the analysis of $E_e[N_e(t)]$ on arbitrary graphs, and can take into account parameters such as d_{min}.

The fourth point is the handling of the parameter $\hat{R}$. The analysis of the algorithm is based on the analysis of short random walks in local regions of the input graph. If we write $\hat{R}$ as n times the average of $1/d_v$, then we would like $ave[1/d_v]$ to be reflected in local regions of the graph. But this is certainly not the case, since local averages are not related to global averages. (This is unlike the case of a tradeoff that depends on d_{min}, where local minima are never smaller than global minimum.) The choice of the ID landmark distribution scheme and extensions of the second point above allow us to make $\hat{R}$ a property that is also reflected in local regions of the graph.

1.4 Outline of the paper

The emphasize in the current paper is on the analysis of the behavior of random walks on graphs, and not on how random walks should be implemented by a computer. As explained earlier, our algorithm ID differs from the algorithm of Broder *et al.* [BKRU] only in the way that landmarks are distributed. The reader is referred to [BKRU] in order to obtain more details about how the input is encoded, and how the algorithm can be implemented.

In Section 2 we analyse the ID algorithm, and prove Theorem 1. In Section 3 we prove Theorem 3. In Section 4 we prove Theorem 4. The proof of Theorem 2 is a simple modification of other proofs in the paper, and is sketched in Section 5. Section 6 includes a discussion of recent related results. A specification of the notation used in this paper is presented in the appendix.

2 The algorithm ID

In this section we prove Theorem 1. The proof is broken into four parts. In Section 2.1 we describe the inverse distribution scheme. We also explain how the input graph G can be viewed as a digraph H with multiple self loops, on which landmarks are distributed according to the stationary distribution rather than the inverse distribution. In Section 2.2 we identify the key property of random walks that needs to be proved in order to establish the $\tilde{O}(m\hat{R})$ time-space tradeoff for algorithm ID. In Section 2.3 we establish some relations between virtual resistance and electrical resistance of the graph when it is viewed as a network of 1 ohm resistors. These relations are important for our purpose, because of known relations between electrical resistance and the behavior of random walks. In Section 2.4 we use electrical resistance arguments in order to prove the key property of Section 2.2.

2.1 The inverse distribution scheme

We have previously described the inverse distribution scheme ID as distributing $p/2$ landmarks with probabilities proportional to degrees of vertices, and $p/2$ landmarks with probabilities inversely proportional to the degrees. We now describe ID in a slightly different way, so as to be fully compatible with the description of the algorithm in Section 1.1. Strictly speaking, the two descriptions are not equivalent, but they are sufficiently similar so that the time-space tradeoffs for USTCON are virtually the same for both versions of ID.

Recall from Section 1.1 that we need to specify a probability distribution D (=ID, in our case) over the vertices of the input graph. Recall that d_i denotes the degree of vertex i, and that $\hat{R} = \sum_{i=1}^{n} 1/d_i$. ID assigns to vertex i the probability $(d_i + m(d_i\hat{R})^{-1})/3m$. Observe that $\sum_{i=1}^{n} d_i = 2m$ and $\sum_{i=1}^{n}(d_i\hat{R})^{-1} = 1$, so $\sum_{i=1}^{n}(d_i + m(d_i\hat{R})^{-1})/3m = 1$, and we have indeed defined a probability distribution.

We also need to specify a length t for the short random walks. We will do so for values of p that satisfy $1 \leq p \leq \hat{R}$. For these values, we will have $t \simeq m\hat{R}/p^2$, up to multiplicative terms that are polylogarithmic in n. Note that t ranges between $\tilde{O}(m\hat{R})$ and $\tilde{\Omega}(m/\hat{R})$, depending on the value of p.

The above, together with Section 1.1, completes our description of algorithm ID. As remarked in Section 1.1, some details as to how the algorithm can actually be implemented are not given in the current paper, and appear in [BKRU]. Here we shall only discuss one of the issues that arise if the algorithm is to be implemented. This is the issue of selecting p landmarks according to the distribution D. Assume that the input graph is given as a linked list of vertices, and for each vertex, a linked list of its neighbors. Then the whole edge list can be scanned in time $O(m)$, and for each vertex, scanning its list of neighbors and computing its degree can be done in time proportional to the degree itself. Computing $\hat{R}$ in this representation (or a close approximation of it) can be done in time $\tilde{O}(m)$ and space $O(\log n)$. Now lets consider the procedure of selecting p landmarks. One can select p integer values independently at random in the range $[1, 3m]$, and sort them. Then one can scan the list of vertices, retaining after each vertex j is scanned the running sum of $\sum_{i=1}^{j}(d_i + m(d_i\hat{R})^{-1})$. On the fly, we give vertex j the landmarks that have values that are between $\sum_{i=1}^{j-1}(d_i + m(d_i\hat{R})^{-1})$ and $\sum_{i=1}^{j}(d_i + m(d_i\hat{R})^{-1})$. The whole procedure takes time $\tilde{O}(m)$, and distributes landmarks in a way similar to ID (except for rounding of probabilities to multiples of $1/3m$), which suffices for the analysis of the time-space tradeoffs to work.

We now present a different way (patterned after [BF]) of viewing the inverse distribution scheme. View the input (undirected) graph G as a symmetric directed graph with $2m$ edges, which we call *arcs*. For an arc $e = (u, v)$ leading from vertex u to vertex v, we call u the *head* of e and v the *tail*. Observe that arc (u, v) exists if and only if arc (v, u) exists. The degree d_v of a vertex is the number of its neighbors, similar to the undirected version of the graph. Now modify the graph by introducing $s_v = m/\hat{R}d_v$ self-loops (arcs of the form (v, v)) to each vertex v. For simplicity we assume that $m/\hat{R}d_v$ is an integer for every vertex v. (The analysis that follows goes through virtually without change even if $m/\hat{R}d_v$ are noninteger, by considering graphs with weighted arcs.) The nonsimple graph thus obtained is denoted by H. Observe that H has n vertices and $3m$ arcs. On H, we employ SD, the landmark distribution scheme used in [BKRU], placing all p landmarks uniformly according to the stationary distribution π_H (the probability that a vertex receives a landmark is directly proportional to $d_v + s_v$, as if landmarks are distributed uniformly at random over the arcs of H). The resulting distribution of landmarks on the vertices of H is identical to the one obtained by ID on the vertices of G.

From now on, we shall view the algorithm ID on graph G as if it is the algorithm SD run on the digraph H. Moreover, we shall "waste" some of the steps of the short random walks by allowing them to traverse the self loops. That is, at vertex v, the pebble sees $d_v + s_v$ outgoing arcs, and selects one of them uniformly at random. Clearly, any value of t that suffices for the SD algorithm on H also suffices for the ID algorithm on G. Thus we can restrict ourselve to analysing the performance of the SD algorithm on H. It remains to establish the value of t (length of short random walks on H) that suffices in order to detect with high probability that $\mathcal{S}$ and $\mathcal{T}$ are in the same component.

2.2 A key property: the expected number of returns

By considering the analysis provided by Broder *et al.* [BKRU], it follows that the value of t should be set so that the following two conditions are met:

1. Walks starting at any arbitrary vertex are expected to hit some landmark (provided that the connected component of the starting vertex is large enough so that if landmarks are placed at random, some of them are expected to be in this component). More precisely, for any vertex v, $P_v[T(u) \le t] = \Omega((\log n)/p)$, where the probability is taken over the choice of random landmark u and the walk W.

2. p random walks (of length t each) that start at random landmarks jointly cover all arcs of H, with high probability. More precisely, for any arc e, $P_\pi[T(e) \le t] = \Omega((\log n)/p)$, where the probability is taken over the choice of starting vertex and walk W.

We provide some intuition why a value of t that satisfies the conditions above suffices to insure (w.h.p.) that the algorithm correctly decides USTCON. Assume that these conditions do not suffice. That is, the graph is connected, but the landmarks can be partitioned into two groups, P_1 and P_2, such that $\mathcal{S} \in P_1$, $\mathcal{T} \in P_2$, and walks of length t that started at landmarks of one of the groups never hit the landmarks of the other group (and vice versa). Let us argue informally that this is an unlikely event. By condition 2 above, every arc e of H is highly likely to be covered by at least one of the p short walks that start at the p landmarks. Assume w.l.o.g. that e is highly likely to be covered by at least one of the short walks that start at P_1. Now by condition 1 above, short walks that start at the tail of e are highly likely to cover some landmarks. If they are likely to cover landmarks from P_2, then we are essentially done, since it is likely that a walk that starts at a landmark of P_1 will reach e, and then reach a landmark of P_2. Hence the only problem that might arise is that for every arc e, the group (P_1 or P_2) from which it is likely to hit e is the same group that is likely to be hit by random walks that start at e. Associate this group with the arc e. Then it can easily be shown that there must be two adjacent arcs that have different groups associated with them. If these arcs have the same tail, then we are done, as then walks from this tail are likely to hit landmarks in both P_1 and P_2. If they have the same head, then we are done, because walks from both P_1 and P_2 are likely to hit this head, and once the head is hit, both arcs are equally likely to be traversed. If the tail of one arc is the head of the other, then walks from both P_1 and P_2 are likely to reach this common vertex, thereafter, its likely that a short walk crosses the arc, and thereafter it is likely that a short walk hits a landmark. The intuitive explanation in the above paragraph contains several gaps, and the reader is referred to [BKRU] for a full proof.

The analysis of Broder *et al.* [BKRU] identifies a key property of random walks that provides upper bounds for the value t that simultaneously satisfies both conditions above.

This property is the number of times that a random walk of length t starting at the tail of arc e is expected to return to e, denoted by $E_e[N_e(t)]$. The following lemma characterizes the time space tradeoff that is achievable as a function of $E_e[N_e(t)]$. In the lemma we introduce a parameter k that shows by how much an improved bound on $E_e[N_e(t)]$ improves the time space product. Essentially, this lemma was used in [BKRU] with $k = 1$.

Lemma 5 *Let k, where $1 \leq k \leq m$ be an arbitrary parameter that may depend on the input graph. Assume that for every arc e and for any value of t in the range $k < t < m^2/k$ the condition $E_e[N_e(t)] = O(\sqrt{t/k})$ holds. Then for any value of p up to $p \leq m/k$, there is some value $t = \tilde{O}(\frac{m^2}{kp^2})$ that satisfies conditions 1 and 2 above.*

Proof: We assume that for some constant c and every edge e, $E_e[N_e(t)] \leq c\sqrt{t/k}$, and prove conditions 1 and 2 above. For condition 1, we let (v, u) be an arc chosen uniformly at random, and show that $P_e[T((v,u)) \leq t] \geq c' \log_n /p$. Note that the probability of crossing a random arc (v, u) is a lower bound on the probability of hitting a landmark, because under ID, landmarks are scattered uniformly at random over the heads (or tails) of arcs.

Throughout the proof of this lemma, probabilities and expectations are taken both over the random steps of the random walk, and over the choice of random arc (v, u). More precisely, if (v, u) appears more than once in the same expression, then all occurrences of (v, u) denote the same arc, but this arc is chosen uniformly at random.

The expected number of times that (v, u) is crossed by a walk of length t that starts at e satisfies:

$$E_e[N_{(v,u)}(t)] \leq P_e[T((v,u)) \leq t](1 + E_{(v,u)}[N_{(v,u)}(t)])$$

and hence

$$P_e[T((v,u)) \leq t] \geq \frac{E_e[N_{(v,u)}(t)]}{1 + E_{(v,u)}[N_{(v,u)}(t)]}$$

Now $E_e[N_{(v,u)}(t)] = t/3m$ (recall that $3m$ is the total number if arcs), and by our assumption $E_{(v,u)}[N_{(v,u)}(t)] \leq c\sqrt{t/k}$. Therefore:

$$P_e[T((v,u)) \leq t] \geq \frac{t}{3m(1 + c\sqrt{t/k})} \geq \frac{\sqrt{tk}}{3m(c+1)}$$

Substituting $t = \frac{m^2}{k}(\frac{c'(c+1)\log n}{p})^2$, condition 1 follows.

For condition 2, observe that

$$E_\pi[N_e(t)] \leq P_\pi[T(e) \leq t](1 + E_e[N_e(t)])$$

and the proof of condition 2 is similar to the proof of condition 1. □

Corollary 6 *Assume that for every arc e and for any value of t in the range $m/\hat{R} < t < m\hat{R}$ the condition $E_e[N_e(t)] = O(\sqrt{t\hat{R}/m})$ holds. Then for any value of p up to $p \leq \hat{R}$, there is some value $t = \tilde{O}(\frac{m\hat{R}}{p^2})$ that satisfies conditions 1 and 2 above.*

Proof: Substitute $k = m/\hat{R}$ in Lemma 5. $\square$

It remains to show that for $m/\hat{R} < t < m\hat{R}$, $E_e[N_e(t)] = O(\sqrt{t\hat{R}/m})$. For this, we first establish some connections between virtual resistance and electrical resistance in graphs.

2.3 Properties of virtual resistance

There are tight connections between behavior of random walks on graphs, and the interpretation of the graph as an electrical network, where each edge represents a resistance of 1 ohm [DS, CRRST, tetali]. The *effective resistance* between vertices u and v, denoted by $R[u,v]$, is the voltage that develops in u if a current of 1 amp is forced into u, and v is grounded (by Ohm's law). The *effective resistance* of a graph, is defined as $R = \max_{u,v \in V}[R[u,v]]$. If the graph is disconnected, then $R = \infty$. The *aggregate resistance* of the graph, denoted by $\bar{R}$, is the sum of effective resistances of the connected components of G. A trivial consequence of electrical theory is that $\bar{R} \leq \bar{D}$.

We define the *virtual resistance* of a simple graph by $\hat{R} = \sum_{v \in V} 1/d_v$. (For the purpose of this definition, assume that there are no isolated vertices.) The virtual resistance is an easily computable parameter of the graph. It can be approximated with arbitrary precision in deterministic logspace and time $\tilde{O}(m)$ (the time bound assumes that the graph is encoded in a suitable compact form, such as an adjacency list).

Lemma 7 *For any simple graph, $\bar{R} < 9\hat{R}$.*

Proof: We prove the lemma for connected graphs, and the proof for disconnected graphs follows from additivity.

Let the vertices u and v be such that $R = R[u,v]$. Inject a current of 1 amp into u, and ground v. Consider the voltages that develop on the vertices of G. The voltage at v is 0, and the voltage at u is R. By the fact that u is the only source of current and v is the only sink, the voltage at any other vertex is some value between 0 and R. (By Kirchoff's current law, the amount of current that enters a vertex must be equal to the amount that leaves the vertex. Hence if a vertex w is not a source/sink, it can have a neighbor of higher voltage if and only if it also has a neighbor of lower voltage, since the first of these neighbors pushes current into w and the second neighbor pulls current, by Ohm's law.)

Let $\{v_1 = v, v_2, ..., v_n = u\}$ be the vertices of G sorted in increasing order of their voltages. Let d_i be the degree of v_i and let R_i be the voltage on v_i. We compute an upper bound on $R_n = R$ based on the sequence of degrees of the vertices.

Assume that we computed an upper bound on R_i. We distinguish between three cases:

1. There exists j, $i < j \leq n$, such that v_j has at least $d_j/3$ edges connecting it to $\{v_1, ..., v_i\}$. The direction of current flow along these edges is from v_j outwards, because v_j has the higher voltage. The sum of currents along these $d_j/3$ edges is at most 1 amp, because 1 amp is the total current in the network. Hence at least one of these edges, say leading to vertex v_ℓ, carries a current of not more than $3/d_j$ amps. Applyings Ohm's law, $R_j \leq R_\ell + 3/d_{v_j}$, and using $R_\ell \leq R_i$ we conclude that $R_j \leq R_i + 3/d_{v_j}$. R_j will serve as an upper bound to the voltages of all vertices up to and including v_j, and we can update $i \leftarrow j$.

2. There exists j, $1 \leq j \leq i$, such that v_j has at least $d_j/3$ edges connecting it to $\{v_{i+1}, ..., v_n\}$. Then by an argument similar to the above, there is some vertex v_ℓ (where $i < \ell \leq n$), such that $R_\ell \leq R_j + 3/d_j$. Of the vertices that have this property, choose v_ℓ for which ℓ is maximal. Observe that $R_\ell \leq R_i + 3/d_j$, because $R_i \geq R_j$. R_ℓ will serve as an upper bound to the voltages of all vertices up to and including v_ℓ, and we can update $i \leftarrow \ell$.

3. Neither Case 1 nor Case 2 above hold. Then we upper bound $R_{i+1} \leq R_i + 1$ (this follows from the fact that $\{v_1, ..., v_i\}$ is connected to $\{v_{i+1}, ..., v_n\}$, and that vertices are sorted in order of increasing voltage), and update $i \leftarrow i + 1$.

Since every vertex can appear at most once as v_j in Case 1, the total voltage contributed by Case 1 is at most $\sum_{v \in V} 3/d_v \leq 3\hat{R}$. Since every vertex can appear at most once as v_j in Case 2 (by our rule of choosing the maximum respective k), the total voltage contributed by Case 2 is at most $\sum_{v \in V} 3/d_v \leq 3\hat{R}$. To bound the contribution of Case 3, let i_1, ..., i_ℓ be the sequence of indices on which Case 3 occurs, and set $i_0 = 0$ and $i_{\ell+1} = n$. The contribution of Case 3 is ℓ. To bound ℓ, observe that for any j, if $i_k < j \leq i_{k+1}$ then it must hold that $i_{k+1} - i_k \geq d_j/3$, as otherwise we could have employed either Case 1 or Case 2 above with v_j. This implies that $\ell < 3\hat{R}$, as follows. Compute $\hat{R} = \sum 1/d_i$ by summing $1/d_i$ over each interval $(i_k, i_{k+1}]$ separately. Within any such interval, $\sum 1/d_i \geq (i_{k+1} - i_k)/3(i_{k+1} - i_k) = 1/3$. Since there are $\ell + 1$ such intervals, it follows that $\hat{R} \geq (\ell + 1)/3$. $\square$

Remark: Often one uses the degree sequence of a connected graph in order to bound its diameter (and this in turn can be used in order to bound the resistance). This approach would not prove Lemma 7, since there are graphs for which $\bar{D} = \Omega(\hat{R}\frac{\log n}{loglog n})$.

In passing, we obtain the following corollary.

Corollary 8 *USTCON can be decided in randomized logspace by a random walk of length $O(m\hat{R})$.*

Proof: A central result of [CRRST] shows that $E_{\mathcal{S}}[T_{\mathcal{T}}] + E_{\mathcal{T}}[T_{\mathcal{S}}] = 2mR[\mathcal{S}, \mathcal{T}]$. It follows from Lemma 7 that if $\mathcal{S}$ and $\mathcal{T}$ are connected then $E_{\mathcal{S}}[T_{\mathcal{T}}] < 18m\hat{R}$.

To decide USTCON, start a random walk at $\mathcal{S}$. If by $36m\hat{R}$ steps it visits $\mathcal{T}$ output *connected*, otherwise *not connected*.

We assume here that $\hat{R}$ can be approximated in time $O(m\hat{R})$ and space $O(\log n)$, which is a reasonable assumption, as discussed in Section 2.1. $\square$

Lemma 7 establishes a relation between $\bar{R}$ and $\hat{R}$ on arbitrary simple graphs. We shall now prove that a "scaled down" version of this relation holds for local regions of graphs, provided that self loops are added as described in Section 2.1. Recall that there we modified our graph G to a directed graph H with $3m$ arcs. $2m$ of this arcs came from considering each edge of G as two antiparallel arcs in H. The other m arcs are self loops that are placed on vertices, where vertex v has $s_v = \frac{m}{\hat{R}d_v}$ self loops. (Recall our assumption that we can treat $\frac{m}{\hat{R}d_v}$ as an integer.) We remark that we define the virtual resistance $\hat{R}$ of H to be that of the original graph G. Likewise, the effective resistance between any two vertices of H is the same as the resistance between these vertices in G (self loops to not change the resistance, and we can treat a pair of antiparallel arcs as one resistor of 1 ohm).

Definition: Let A be a proper subset of arcs in H. Let $I_A \subset V$ denote the set of vertices that are in the "interior" of A. That is, for each $v \in I_A$ all arcs incident with v are in A. Let $B_A \subset V$ denote the "boundary" of A. That is, each $v \in B_A$ has a neighbor in I_A, and has at least one arc not in A. Add arcs to the subgraphs induced on B_A such that any two vertices are connected by a pair of antiparallel arcs. Consider now the subgraph induced on $I_A \bigcup B_A$. Then the *exit resistance* of A is $R_A = \max_{u \in I_A} \min_{v \in B_A}[R[u,v]]$. If I_A is empty, then $R_A = 0$.

Intuitively, the exit resistance of a set A measures the resistance between the interior of the set to its boundary. By the analogy between random walks and resistance, the exit resistance can be used to bound the expected time that it takes a random walk to leave A.

The following lemma scales Lemma 7 to the region of A. The scaling factor is $|A|/m$. In this lemma, the exit resistance plays the role that was originaly played by aggregate resistance.

Lemma 9 *Let A be a proper subset of edges in H. Then $R_A \leq 9(1 + \hat{R}\frac{|A|}{m})$.*

Proof: Let E_A be the set of arcs incident with I_A. Then $E_A \subset A$, hence $|E_A| \leq |A|$. $|E_A| \geq \sum_{v \in I_A}(d_v + s_v) = \sum_{v \in I_A}(d_v + \frac{m}{\hat{R}d_v}) \geq \frac{m}{\hat{R}} \sum_{v \in I_A} 1/d_v$. Define $\hat{R}_A = \sum_{v \in I_A} 1/d_v$. Rearranging gives $\hat{R}_A \leq \frac{|A|}{m}\hat{R}$.

To compute R_A, observe that $R_A \leq max_{u,v \in I_A \bigcup B_A} R[u,v]$. Using Lemma 7 on the subgraph induced by $I_A \bigcup B_A$, and noting that $\sum_{v \in I_A} 1/d_v = \hat{R}_A$ and that $\sum_{v \in B_A} 1/d_v \leq 1$ (by the fact that vertices in B_A form a clique, and in addition, each such vertex is adjacent to a vertex in I_A), we obtain that $R_A \leq 9(\hat{R}_A + 1)$. $\square$

For additional relations between virtual resistance and resistance, see Section 6.

2.4 Bounding the expected number of returns

We bound $E_e[N_e(t)]$, the key property identified in Section 2.2, by extending a proof technique developed by Aldous [aldous] in his proof that $E_v[N_v(t)] = O(\sqrt{t})$ for vertices on regular graphs.

Lemma 10 *Let H be a digraph with n vertices and $3m$ arcs, where each vertex has d_v neighbors and s_v self loops as described above. Let e be an arbitrary arc in H, and let $m/\hat{R} < t < m\hat{R}$. Then $E_e[N_e(t)] \leq 10 + 6\sqrt{t\hat{R}/m} \leq 16\sqrt{t\hat{R}/m}$.*

Proof: Let $A_s = \{(u,v)|E_{(u,v)}[N_e(t)] > s\}$ be the set of arcs from (the tail of) which random walks of length t are expected to hit e more than s times, and let A_s^c be the complement of this set. Then $E_e[N_e(t)] \leq E_e[N_e(T_{A_s^c})] + s$. To compute $E_e[N_e(T_{A_s^c})]$, the expected number of times that e is hit before A_s^c is reached, we consider the size of A_s.

Claim 11 *The number of arcs in A_s satisfies $|A_s| \leq t/s$.*

Proof: Let $\bar{e}$ denote the arc antiparallel to e, and let $B_s = \{(u,v)|E_{\bar{e}}[N_{(u,v)}(t)] > s\}$. We show that $(u,v) \in B_s$ if and only if $(v,u) \in A_s$. Let (x,y) and (u,v) be two arcs. For any t', the probability that a walk that starts at y crosses (u,v) at time t' is equal to the probability that a walk that starts at u crosses (y,x) at time t' (by the reversibility of the random walk process). It follows that $E_{(x,y)}[N_{(u,v)}(t)] = E_{(v,u)}[N_{(y,x)}(t)]$, and therefore an arc is in B_s if and only if the anti-parallel arc is in A_s.

To complete the proof of the claim, observe that by linearity of expectation, $|B_s| < t/s$. It follows that $|A_s| < t/s$. $\square$

Claim 12 *The expected number of times that e is traversed before exiting the set A_c satisfies $E_e[N_e(T_{A_s^c})] \leq 10 + \frac{9|A_s|\hat{R}}{m}$.*

Proof: We shall use the notion of exit resistance from Section 2.3. Let w be the head of e. Let u be a vertex on the boundary B_{A_c} that maximizes $R[w,u]$. From Lemma 9 we obtain that $R[w,u] \leq R_{A_s} \leq 9(1 + |A_s|\hat{R}/m)$. Now consider an arc $(u,v) \notin A_c$ (such an arc must exist). Then $R[w,v] \leq 1 + 9(1 + |A_s|\hat{R}/m) = 10 + 9|A_s|\hat{R}/m)$. When the walk reaches v it must have left A_c.

By the results of [CRRST], $R[w,v]$ is exactly the expected number of times that any arc leading out of w is traversed in a random walk that starts at w and continues until it first hits v. Hence $E_e[N_e(T_{A_s^c})] \leq 10 + 9|A_s|\hat{R}/m$. $\square$

Set $s = 3\sqrt{t\hat{R}/m}$. Then $|A_s| \leq \frac{1}{3}\sqrt{tm/\hat{R}}$ and $E_e[N_e(T_{A_s^c})] \leq 10 + 3\sqrt{t\hat{R}/m}$. Hence $E_e[N_e(t)] \leq 10 + 6\sqrt{t\hat{R}/m}$. $\square$

The proof of Theorem 1 is a straightforward consequence of Corollary 6 and Lemma 10.

3 On the limitations of the method

Broder *et al.* [BKRU] gave the general scheme of randomized algorithms for USTCON. The basic components of their scheme are a landmark distribution scheme D, and exploration of the graph by short random walks of length t. The landmark distribution scheme of [BKRU] (which we call SD) was improved in [BF], and then again in the current paper. Can a new yet unstudied landmark distribution scheme (or just tighter analysis leading to a lower value of t) offer further improvements?

Clearly, improvement in special cases is possible. For example, if a graph contains a vertex of degree $n-1$, or if $d_{\mathcal{S}} + d_{\mathcal{T}} > n-2$, then $\mathcal{S}$ and $\mathcal{T}$ are connected, but none of our algorithms currently check for these conditions. Hence checking for special conditions leads to improvement in special cases. Also, it is relatively straightforward to improve landmark distribution scheme, using the fact that we ignore constant factors in our analysis. Thus we can augment the landmark distribution scheme ID by throwing in p additional landmarks on vertices that satisfy certain new conditions (e.g., have degree close to the average degree, or have degree that is significantly smaller then the average degrees of their neighbors), and the increased time and space requirement of the new algorithm compared to ID will be suppressed by the $\tilde{O}$ notation. It is certainly possible that such augmented landmark distribution schemes would significantly improve upon ID for some class of graphs. Is there a limit to what can be achieved by changing the landmark distribution scheme and improving the analysis? To answer this question we introduce the notion of a *fair* landmark distribution scheme (FD). A similar notion in the context of deterministic algorithms for USTCON was considered in [BBRRT].

Definition: The *ℓ-neighborhood* of vertex v is the subgraph induced by the vertices of distance at most ℓ from v. Two vertices, u and v, are *ℓ-similar*, if their ℓ-neighborhoods are isomorphic.

Definition: A landmark distribution scheme is *ℓ-fair* if it does not distinguish between ℓ-similar vertices. That is, any landmark is equally likely to be placed on u or v, if these are ℓ-similar vertices. A landmark distribution scheme is *fair* (FD), if it is ℓ-fair for some constant ℓ. An FD algorithm is the algorithm described in Section 1.1, in which D (the landmark distribution) is a fair distribution scheme and t (the length of short random walks) is set to a value that ensures correctness of the UNSTCON algorithm.

Clearly, SD, MD, and ID, are all fair distribution schemes. We now proceed to show that FD algorithms cannot achieve a tradeoff of $\tilde{O}(m\bar{D})$. Our proof actually applies to

ℓ-fair algorithms for $\ell < \sqrt{n}/2$.

Consider the *3-star* graph. It has 3 *centers.* From each center extend $\sqrt{n}/3$ "rays" (simple paths), each of length $\sqrt{n}$. Two of the centers are connected by a connecting ray of length $\sqrt{n}$, and the third center is disconnected from the other two centers. The 3-star graph has roughly n edges, two connected components, and aggregate diameter $\bar{D} \simeq 2\sqrt{n} + 3\sqrt{n} = 5\sqrt{n}$.

Consider an arbitrary FD algorithm which gets as input the 3-star graph, with the vertices labeled at random. $\mathcal{S}$ and $\mathcal{T}$ are placed on two of the centers, and the algorithm must decide if they are connected. If $p = 1$ (one pebble at $\mathcal{S}$), the algorithm can decide connectivity in time $\tilde{O}(m\bar{D}) = \tilde{O}(n^{3/2})$. But increasing the space (increasing p up to about $\sqrt{n}$) cannot result in a significant improvement in the running time (which is $t \cdot p$), foiling an attempt to get a time-space tradeoff of $\tilde{O}(m\bar{D})$.

Lemma 13 *On the 3-star graph, any FD algorithm that uses $p << \sqrt{n}/\log n$ landmarks must make short random walks of length $t = \tilde{\Omega}(n^{3/2}/p)$ in order to decide USTCON, giving a time-space product of $\tilde{O}(n^{3/2}p)$.*

Proof: Assume otherwise, that t is small, and derive a contradiction.

Assume that $\mathcal{S}$ and $\mathcal{T}$ are the two connected centers. Locally, the i^{th} vertex on the connecting ray is indistinguishable from the i^{th} vertex on any other ray (this holds up to neighborhoods of distance $\sqrt{n}/2$). By the fairness property of FD algorithms, the probability that the connecting ray receives a landmark is at most $3p/\sqrt{n} << 1$. Thus we can assume that the connecting ray does not receive a landmark. Hence the only way that the algorithm can detect that $\mathcal{S}$ and $\mathcal{T}$ are connected is by a random walk that crosses the connecting ray.

W.l.o.g., assume that the direction of crossing is from $\mathcal{S}$ to $\mathcal{T}$. Regardless of the location of the landmark from which the random walk originated, the pebble that crosses the connecting ray must go through $\mathcal{S}$.

Claim 14 *For $t \geq n$, $P_{\mathcal{S}}[T_{\mathcal{T}} \leq t] \leq 12t/n^{3/2}$.*

Proof: Assume that $P_{\mathcal{S}}[T_{\mathcal{T}} \leq t] > 12t/n^{3/2}$. We shall compute $E_{\mathcal{S}}[T_{\mathcal{T}}]$ (the expected hitting time from $\mathcal{S}$ to $\mathcal{T}$) in two different ways, and derive a contradiction. For the purpose of this computation, we can disconnect from $\mathcal{T}$ all vertices except the connecting ray, and remain with a star centered at $\mathcal{S}$ which has $\sqrt{n}/3$ rays, each of length $\sqrt{n}$. $\mathcal{T}$ is the endpoint of one of these rays.

One way of computing $E_{\mathcal{S}}[T_{\mathcal{T}}]$ is based on electrical resistance. The total number of edges in the graph under consideration is $n/3$. The electrical resistance $R[\mathcal{S}, \mathcal{T}]$ between $\mathcal{S}$ and $\mathcal{T}$ is exactly $\sqrt{n}$. Then it follows from [CRRST] that $E_{\mathcal{S}}[T_{\mathcal{T}}] + E_{\mathcal{T}}[T_{\mathcal{S}}] = 2n^{3/2}/3$.

Standard analysis of random walks on a line with $\sqrt{n}$ edges shows that $E_{\mathcal{T}}[T_{\mathcal{S}}] = n$. Hence $E_{\mathcal{S}}[T_{\mathcal{T}}] = 2n^{3/2}/3 - n$.

The other way of computing $E_{\mathcal{S}}[T_{\mathcal{T}}]$ is based on the (false) assumption that $P_{\mathcal{S}}[T_{\mathcal{T}} \leq t] > 12t/n^{3/2}$. Observe that for any vertex u, $E_u[T_{\mathcal{S}}] \leq n$, and hence by Markov's inequality, $P_u[T_{\mathcal{S}} \geq 2n] \leq 1/2$. It follows that $P_u[T_{\mathcal{T}} \leq t + 2n] > 12t/2n^{3/2} = 6t/n^{3/2}$. Hence if we partition a random walk of arbitrary length to segments of length $t + 2n$, then in each segment, independently of past segments, there is probability at least $6t/n^{3/2}$ of hitting $\mathcal{T}$. If follows that $E_{\mathcal{S}}[T_{\mathcal{T}}] \leq (t + 2n)n^{3/2}/6t$. For $t \geq n$, we obtain that $E_{\mathcal{S}}[T_{\mathcal{T}}] \leq n^{3/2}/2$, a contradiction (when n is sufficiently large). $\Box$

It follows from the claim that whenever $t \geq n$, at least $n^{3/2}/24t$ random walks of length t are needed in order to have probability 1/2 of discovering that $\mathcal{S}$ and $\mathcal{T}$ are connected. Hence the time spent on random walk steps is at least $n^{3/2}/24$. For $t < n$, the number of random walks of length t that are needed is at least the number of random walks of length n that are needed, which is $\sqrt{n}/24$. Algorithm FD makes this many walks only if the number of landmarks is $\Omega(\sqrt{n}/\log n)$. Hence for $p << \sqrt{n}/\log n$, the time-space product obtained is $\tilde{\Omega}(n^{3/2}p)$. $\Box$

The proof of Theorem 3 follows from the fact that for the 3-star graph, $m\bar{D} \simeq n^{3/2}$.

The number of edges (and hence $m\bar{D}$) in the above example can be increased, by replacing the centers of stars by cliques of size $O(n)$. Then the proof of Lemma 13 can be modified to show that for $S << \sqrt{n}$, FD algorithms still require time $T = \tilde{\Omega}(m\bar{D}) = \tilde{\Omega}(n^{5/2})$.

Remark: On the latter class of graphs, in which the centers of stars are replaced by cliques, a simple randomized algorithm not in the class FD can decide USTCON in logspace and time $\tilde{O}(n^{3/2})$. The algorithm selects a random vertex, and takes two random walks from it, each of length $O(n)$. If one walk hits $\mathcal{S}$ and the other hits $\mathcal{T}$, then the algorithm declares that they are connected. If the algorithm is repeated $O(\sqrt{n})$ times, with independently chosen random starting vertices, and if it fails to determine that $\mathcal{S}$ and $\mathcal{T}$ are connected, it decides that they are not connected.

4 More on short random walks

Barnes and Feige [BF] proved a conjecture of Linial that the expected time for a random walk to discover its $\mathcal{N}$th distinct vertex is $O(\mathcal{N}^3)$. They in turn conjectured that this bound can be refined to $O(\mathcal{N} + \mathcal{N}^2 min[\mathcal{N}, d_{max}]/d_{min})$, and proved this bound up to a multiplicative logarithmic factor. The conjecture is known to hold if $\mathcal{N} = \Omega(n)$, by the results of Kahn *et al.* [KLNS]. We prove the conjecture for arbitrary graphs and arbitrary values of $\mathcal{N}$. The proof combines ideas that appear in Section 2, some of which originate in the works of [aldous] and [BF]. We remark that for the special case of regular graphs, the proof can be simplified, and it can be obtained as a corollary of the work of Aldous on regular graphs [aldous].

Lemma 15 *Let H be a symmetric digraph with n vertices and M arcs, including multiple self loops on vertices. Let d_* denote the minimum number of arcs incident with a vertex (including self-loops), and let d_{min} denote the minimum number of vertices adjacent to a vertex. Let e be an arbitrary arc in H, and let t satisfy $t < 3M^2/d_{min}d_*$. Then the expected number of times that a walk of length t returns to e satisfies $E_e[N_e(t)] \leq 2 + 2\sqrt{3t/d_{min}d_*}$.*

Proof: The proof is a simplification of the proof of Lemma 10, and is sketched below.

Let $A_s = \{(u,v)|E_{(u,v)}[N_e(t)] > s\}$. Then $E_e[N_e(t)] \leq E_e[N_e(T_{A_s^c})] + s$. Claim 11 shows that $|A_s| < t/s$.

Claim 16 $E_e[N_e(T_{A_s^c})] \leq 2 + \frac{3|A_s|}{d_{min}d_*}$.

Proof: Let I_{A_s} denote the set of vertices in the interior of the arc set A_s. Then $|I_{A_s}| \leq |A_s|/d_*$. The diameter of the subgraph induced on I_{A_s} is at most $3|I_{A_s}|/d_{min}$ (see e.g. [aldous]). Let v be the head of e and w be a vertex in the boundary set B_{A_s}. Using the distance bound on electrical resistance, $R[v,w] \leq 1 + 3|I_{A_s}|/d_{min} \leq 1 + 3|A_s|/d_{min}d_*$. By the identity relating resistance to the number of returns to an arc [CRRST], $E_e[N_e(T_{A_s^c})] \leq R[v,w] + 1 \leq 2 + 3|A_s|/d_{min}d_*$. □

Set $s = \sqrt{3t/d_{min}d_*}$. □

Lemma 17 *Let H be a symmetric digraph with n vertices and M arcs, including multiple self loops on vertices. Let d_* denote the minimum number of arcs incident with a vertex (including self-loops), and let d_{min} denote the minimum number of vertices adjacent to a vertex. Let $E[T(\mathcal{M})]$ denote the expected time to discover $\mathcal{M}$ arcs, where $d_{min}d_* < \mathcal{M} < \varepsilon M$, for some positive constant $\varepsilon < 1$ to be determined later. Then $E[T(\mathcal{M})] = O(\mathcal{M}^2/d_{min}d_*)$.*

Proof: By Lemma 15, $E_e[N_e(t)] = O(\sqrt{1 + t/d_{min}d_*})$. Intuitively, in $\mathcal{M}^2/d_{min}d_*$ steps, we expect any arc e to be visited at most $O(\mathcal{M}/d_{min}d_*)$ times, hence $\mathcal{M}$ distinct arcs are visited. We now make this intuition precise.

We want the expected time to visit $\mathcal{M}$ arcs. Consider an arbitrary starting arc, and walks of length $c\mathcal{M}^2/d_{min}d_*$, where c is a large enough constant. It suffices to show that with probability at least 1/4, such a walk is *successful* (that is, visits $\mathcal{M}$ distinct arcs). Thereafter, noting that this fraction of walks is successful for any starting arc, the expected time to visit $\mathcal{M}$ arcs is at most $\frac{4c\mathcal{M}^2}{d_{min}d_*}$ (by repetition). So assume the contrary - that for some starting arc e, with probability 3/4 walks visit less than $\mathcal{M}$ edges.

We now derive a contradiction. Consider the following random procedure for selecting an arc e'. Select an integer j uniformly at random in the range $[0, c\mathcal{M}^2/d_{min}d_*]$. Start a random walk at e, and stop after j steps. Denote the arc crossed at the j^{th} step by e'. We show that $E_{e'}[N_{e'}(c\mathcal{M}^2/d_{min}d_*)] > 4\sqrt{3c}\mathcal{M}/d_{min}d_*$, contradicting Lemma 15 (the expectation in E'_e is taken both over the choice of e' and over the random steps of the random walk).

Consider the whole $c\mathcal{M}^2/d_{min}d_*$ steps of the walk from which e' is selected. With probability 3/4, the walk is not successful (i.e., it visits less than $\mathcal{M}$ distinct arcs). For an unsuccessful walk, there are at most $\mathcal{M}k$ choices of j for which the number of subsequent returns of this walk to the arc e' selected is at most k. Hence,

$$P_{e'}[N_{e'}(c\mathcal{M}^2/d_{min}d_*) \leq k] < \frac{1}{4} + \frac{3}{4} \cdot kd_{min}d_*/c\mathcal{M}$$

where the probability is taken over the choice of the random walk and of the index j (leading to the selection of arc e').

For $k = 8\sqrt{3c}\mathcal{M}/d_{min}d_*$ and large enough c (e.g., $c > 3^3 \cdot 2^6$) we obtain that $P_{e'}[N_{e'}(c\mathcal{M}^2/d_{min}d_*) > k] \geq 1/2$, implying $E_{e'}[N_{e'}(c\mathcal{M}^2/d_{min}d_*)] > k/2 = 4\sqrt{3c}\mathcal{M}/d_{min}d_*$.

To contradict Lemma 15, ε in Lemma 17 should be small enough so that $c\mathcal{M}^2/d_{min}d_*$, the length of walks considered in the proof of Lemma 17, would still be in the range required in Lemma 15. Hence we require that $3^3 \cdot 2^6\mathcal{M}^2/d_{min}d_* \leq 3M^2/d_{min}d_*$, which holds when $\varepsilon < 1/48$. □

Lemma 18 *Let G be an undirected simple graph (no self-loops) with n vertices and m edges, and minimum degree d_{min}. Let $E[T(\mathcal{N}, \mathcal{M})]$ denote the expected time to discover either $\mathcal{N}$ vertices or $\mathcal{M}$ edges (whichever comes first), where $\mathcal{M} > \mathcal{N}$, and $d_{min} < \mathcal{N} < \epsilon n$, for some positive constant $\epsilon < 1$ to be determined later. Then $E[T(\mathcal{N}, \mathcal{M})] = O(\mathcal{N}\mathcal{M}/d_{min})$.*

Proof: Obtain a symmetric digraph H from G, by replacing each edge with two antiparallel arcs, and by placing $d_{self} = \lceil \mathcal{M}/\mathcal{N} \rceil$ self loops on each vertex. Then H has $M \geq n\mathcal{M}/\mathcal{N}$ arcs. On H, we want to discover $3\mathcal{M} + \mathcal{N} < 4\mathcal{M}$ arcs, since then we either discover $\mathcal{M}$ of the original edges, or $\mathcal{M} + \mathcal{N}$ distinct self loops, implying the discovery of $\mathcal{N}$ vertices. For $\epsilon < \frac{1}{4} \cdot \varepsilon < \frac{1}{4} \cdot \frac{1}{48}$, we can apply Lemma 17, and obtain

$$E[T(4\mathcal{M})] = O(\mathcal{M}^2/d_{min}d_*) = O(\mathcal{M}^2/d_{min}d_{self}) = O(\mathcal{N}\mathcal{M}/d_{min})$$

□

Proof of Theorem 4: We assume that $\mathcal{N} > d_{min}$, as otherwise the proof of the theorem is trivial. We can assume that $\mathcal{N} < \epsilon n$, as otherwise the proof follows from [KLNS] (see also [CFS] for improved bounds). Hence we can apply Lemma 18 with $\mathcal{M} = \min[\mathcal{N}^2, \mathcal{N}d_{max}]$, where visiting $\mathcal{M}$ edges guarantees the visit of $\mathcal{N}$ vertices.

5 The algorithms SD and MD

The ideas needed for the proof of Theorem 2 appear in previous sections. We shall only sketch the proof of Theorem 2.

Proof: The upper bounds on SD and MD follow from a combination of Lemma 5 and Lemma 15. Lemma 5 applies when landmarks are distributed according to the stationary distribution of random walk (uniformly on arcs). For algorithm SD we use Lemma 15 with $d_* = d_{min}$ and obtain $E_e[N_e(t)] \leq 2 + 2\sqrt{3t}/d_{min}$. Note that when $t \geq (d_{min})^2$ the additive factor of 2 in the upper bound is dominated by the term $2\sqrt{3t}/d_{min}$. This implies that Lemma 5 can be applied with $k = (d_{min})^2$, giving a value of $t = \tilde{O}((m/pd_{min})^2)$, for a time space tradeoff of $\tilde{O}((m/d_{min})^2)$. For algorithm MD, we transform the input graph G into a symmetric digraph H, placing m/n self loops on each vertex. Now algorithm MD on G becomes similar to SD on H, and we can apply Lemma 5. We first use Lemma 15 with $d_* \geq m/n$ to obtain that we can set $k = md_{min}/n$. We obtain $t = \tilde{O}(mn/d_{min}p^2)$ and a time space product of $\tilde{O}(mn/d_{min})$.

For the lower bounds on SD and MD, we present a family of graphs on which neither SD nor MD achieve a time space product of $O(m\hat{R})$. Consider three cliques of size roughly $(n - \sqrt{n})/3$, one containing $\mathcal{S}$, and another containing $\mathcal{T}$. Two of the cliques are connected by a path of length $\sqrt{n}$. For this graph $m = \Theta(n^2)$ and $\hat{R} = \Theta(\sqrt{n})$. Hence in logarithmic space, USTCON can be decided in time $O(m\hat{R}) = O(n^{5/2})$ on these graphs. Now if the number of landmarks p is much smaller than $\sqrt{n}$, then neither SD nor MD are likely to place landmarks on the connecting path. In order to discover that $\mathcal{S}$ and $\mathcal{T}$ are connected it is then necessary that some walk crosses the path from one clique to the other. But if all walks start from the cliques, this is not likely to happen unless the total number of random walk steps is $\Omega(n^{5/2})$ (this can be proved using techniques similar to those employed in Section 3). Hence neither SD nor MD can trade space for time, at least not for space $<< \sqrt{n}$. $\Box$

6 Discussion

One of the goals of this paper is to better understand the process of random walks on graphs. We identified the virtual resistance $\hat{R} = \sum_v 1/d_v$ as an important parameter related to random walks. Recent work has shed more light on the inter-relation between the virtual resistance and random walks. It turns out that a more "appropriate" definition of virtual resistance is $\hat{R} = \sum_v 1/(d_v+1)$. It can be shown that for any graph, $\bar{R} < 3\hat{R}$, and this bound is best possible, upto low order terms (e.g., for the *path*). This result implies automatic improvement of constants throughout Section 2. More interestingly, $\hat{R}$ provides a tight characterization (within constant factors) of certain graph properties (such as R_{span}, the resistance along the edges of the minimum resistance spanning tree) that relate to the *cover time* (the expected time for a random walk to visit all vertices of the graph). These results are described in [CFS].

The question of graph connectivity, rather than $\mathcal{S}$-$\mathcal{T}$-connectivity is studied in [feige]. There it is shown that connectivity can be decided in randomized logspace within $O(m\hat{R})$ random walk steps. The analysis in [feige] also takes particular attention to avoid unnecessary logarithmic overhead in the running time. Presumably, the same principles for saving logarithmic factors can be used also in the context of our time-space tradeoffs, though this requires a detailed examination of [BKRU].

Acknowledgements

It is a pleasure to acknowledge the great influence of David Aldous and Greg Barnes on the work reported here. I thank David Aldous for sending me the draft of his book, and Greg Barnes for extensive e-mail discussions. I would also like to thank Nati Linial and Prabhakar Raghavan for helpful discussions, and for directing me to relevant literature. Finally, I would like to thank two anonymous referees for many useful comments and suggestions.

References

[aldous] D. J. Aldous. "Reversible markov chains and random walks on graphs". *Draft of first six chapters of book*, January 26, 1993.

[AKLLR] R. Aleliunas, R. M. Karp, R. J. Lipton, L. Lovász, and C. Rackoff. "Random walks, universal traversal sequences, and the complexity of maze problems". In *Proc. of 20th Annual Symposium on Foundations of Computer Science*, 218–223, San Juan, Puerto Rico, October 1979.

[BF] G. Barnes and U. Feige. "Short random walks on graphs". *SIAM J. Discrete Math., 9(1), 19–28, 1996.* Some of the results cited appear only in the preliminary version, in the proceedings of *25th Symposium of the Theory of Computing, 728–737, 1993.*

[BBRRT] P. Beame, A. Borodin, P. Raghavan, W. Ruzzo, and M. Tompa. "Time-space tradeoffs for undirected graph traversal". In *Proc. of 31st Annual Symposium on Foundations of Computer Science*, 429–438, St. Louis, MO, October 1990.

[BCDRT] A. Borodin, S. Cook, P. Dymond, W. Ruzzo, and M. Tompa. "Two applications of inductive counting for complementation problems". *SIAM J. Comput.,* 18(3):559-578, 1989.

[BKRU] A. Z. Broder, A. R. Karlin, P. Raghavan, and E. Upfal. "Trading space for time in undirected s-t connectivity". *SIAM J. on Computing, Vol 23, 1994, 324–334.*

[CRRST] A. Chandra, P. Raghavan, W. Ruzzo, R. Smolensky, and P. Tiwari. "The electrical resistance of a graph captures its commute and cover times". In *Proceedings of the Twenty First Annual ACM Symposium on Theory of Computing*, 574–586, Seattle, WA, May 1989.

[CFS] D. Coppersmith, U. Feige, and J. Shearer. "Random walks on regular and irregular graphs". *Technical report CS93-15 of the Weizmann Institute, 1993.* To appear in *SIAM J. Discrete Math.*

[DS] P. G. Doyle and J. L. Snell. "Random Walks and Electrical Networks". *The Mathematical Association of America*, 1984.

[edmonds] J. Edmonds. "Time-Space Tradeoffs For Undirected ST-Connectivity on a JAG". In *Proc. of 25th Symposium of the Theory of Computing, 718–727, 1993.*

[feige] U. Feige. "A fast randomized logspace algorithm for graph connectivity". *ICALP94 Proceedings, Lecture Notes in Computer Science 820, Springer Verlag, 499–507.*

[KLNS] J. D. Kahn, N. Linial, N. Nisan, and M. E. Saks. "On the cover time of random walks on graphs". *Journal of Theoretical Probability*, 2(1):121–128, January 1989.

[ruzzo] W. L. Ruzzo. Lecture Notes for CS 533. Unpublished Manuscript, University of Washington, 1989.

[tetali] P. Tetali. "Random walks and the effective resistance of networks". *Journal of Theoretical Probability*, 4:101-109, 1991.

APPENDIX – Notation

$G(V, E)$ - a simple graph (no parallel edges and self-loops).

n - number of vertices in G.

m - number of edges in G.

$\mathcal{S}$, $\mathcal{T}$ - Vertices in G.

SD - *stationary distribution*, used in [BKRU].

MD - *mixed distribution*, used in [BF].

ID - *inverse distribution*, introduced in this paper.

d_v - degree of vertex v (not including self-loops).

d_{min} - $\min_{v \in V}[d_v]$.

D - diameter of graph.

$\bar{D}$ - sum of diameters of connected components.

R - maximum effective resistance in graph.

$\bar{R}$ - sum of maximum effective resistances of connected components.

$\hat{R}$ - virtual resistance. $\hat{R} = \sum_{v \in V} 1/d_v$.

H - a symmetric digraph with multiple self-loops.

s_v - number of self loops on v in H.

M - number of arcs in H.

d_* - minimum number of arcs connected to a vertex (including self-loops).

S - space used by algorithm.

T - time used by algorithm.

p - number of landmarks placed on random vertices.

t - length of short random walk taken by a pebble.

u,v,w, x,y - vertices.

e - edge or arc. Arcs are also denoted by their *heads* and *tails* (e.g., (u, v)).

$\bar{e}$ - the arc antiparallel to e.

A^c - complement of subset A.

$N_e(t)$ - number of times e is visited in a random walk of length t.

$T(A)$ - time in which A is reached.

$P[\mathcal{E}]$ - probability of event $\mathcal{E}$.

$E[\mathcal{X}]$ - expected value of random variable $\mathcal{X}$.

P_e, E_e - the subscript denotes the starting edge of the walks for which probability or expectation is computed.

π - stationary distribution for a random walk (probability of each vertex is directly proportional to its degree, including self-loops).