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, \( S \) and \( T \). This problem is known as undirected \( S-T \)-connectivity, or USTCON. It can be solved in time \( O(m) \) and space \( O(n) \) deterministically, by breadth first search (BFS). (Throughout, \( O \) and \( \tilde{O} \) notation is used in order to suppress polylogarithmic terms. E.g., \( n \log n = O(n) \).) We also use the convention that suppressed terms are polylogarithmic in \( n \), regardless of the explicit terms that appear in \( O(\cdot) \). Hence \( \log n = O(1) \), and \( p \log n = O(p) \). 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 \( S \) and \( T \) are connected, then a random walk that starts at \( S \) is expected to reach \( T \) in \( mn \) steps. This leads to a randomized algorithm with one-sided error: start a walk at \( S \). If \( T \) is reached within \( 3mn \) steps, declare that \( S \) and \( T \) are connected; otherwise—not connected. The error probability can be made arbitrarily small by independent repetition of the algorithm. (A log space “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 trade-offs between time and space for randomized algorithms for USTCON. Observe that both BFS and the random walk algorithm achieve a time space trade-off of \( ST = \tilde{O}(mn) \) (where \( T \) denotes time and \( S \) denotes space). Is there a randomized algorithm that for any space bound \( n < S < n \), 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 trade-off. 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, \( S \) and \( T \). 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 \( S \) and \( T \) are in the same connected component. The algorithm is
randomized and may have a small (e.g., less than $\frac{1}{4}$) probability of error. The algorithm runs in time $O(m^2/p)$, giving a time-space product of $O(m^2)$.

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 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 $t = O(m^2/p^3)$.

Distributing landmarks. The algorithm decides randomly on $p + 2$ vertices to be denoted as landmarks in $G$. Two of these landmarks are $F$ and $S$. 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 $O(\log n)$ 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 $F$ and $S$ 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 $F$ and $S$ belong to the same component when in fact they do.) To reduce the probability of error below $\epsilon$, 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 $F$ and $S$ 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 $O(p)$ 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 $O(p)$, 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 $O(m^2/p)$ when $t = O(m^2/p^3)$. 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 $O(pm^2/p^3) = O(m^2)$.

For sparse graphs (where $m = O(n)$), the [BKRU] algorithm achieves a smooth trade-off between BFS and the random walk algorithm. However, for dense graphs, the quality of the [BKRU] trade-off 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 trade-off, 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 trade-offs for the modified algorithm MD, including $ST = O(m^3/n^{1/4})$. Barnes and Feige conjectured that the MD algorithm achieves a time-space trade-off of $ST = O(mn/d_{\min})$, where $d_{\min}$ is the minimum degree in the graph. This conjectured trade-off of $O(mn/d_{\min})$ for MD is patterned after the similar bound of Kahn et al. [KLNS] on the cover time of graphs. In particular, its value is $O(n^3)$ 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 ($probability that t receives a landmark is linear in d_t$) and $p/2$ landmarks according to an “inverse” distribution ($probability that t receives a landmark is linear in 1/d_t$). 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 trade-off 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 trade-off curve, which satisfies \( ST = \tilde{O}(p^2) \).

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 \( \hat{R} \leq n/d_{\text{min}} \). The main theorem of the paper is:

**Theorem 1.** Algorithm ID achieves a time–space trade-off of \( ST = \tilde{O}(m\hat{R}) \), where \( \hat{R} = \sum_{v \in V} 1/d_v \). 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 trade-off that we prove for ID, \( ST = \tilde{O}(m\hat{R}) \), can also be expressed elegantly as \( ST = \tilde{O}(\sum_{v \in V} 1/d_v) \). We also improve the known trade-offs for the previously suggested algorithms, in particular, proving the conjecture of [BF] regarding MD.

**Theorem 2.** Algorithm SD achieves a time–space trade-off of \( ST = \tilde{O}(m/d_{\text{min}}) \). Algorithm MD achieves a time–space trade-off of \( ST = \tilde{O}(mn/d_{\text{min}}) \). Neither algorithm achieves the trade-off \( ST = \tilde{O}(R) \) achieved by the ID algorithm.

Observe that both the trade-off guarantee achieved by ID and the trade-off guarantee achieved by MD improve over \( ST = \tilde{O}(m\hat{R}) \) (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 \( \tilde{O}(d) \). The \( \tilde{O}(m\hat{R}) \) guarantee on the trade-off achieved by ID is clearly stronger than the \( \tilde{O}(mn/d_{\text{min}}) \) guarantee on the trade-off 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 \( d_{\text{min}} \) (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 trade-offs 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 \( \tilde{O}(m\hat{R}) \) trade-off? Before attempting to answer this question, we should ask ourselves which other parameters of the graph may be relevant to random walks.

The value \( \Omega(m\hat{R}) \) is an upper bound on the expected time that it takes a random walk that starts at \( \mathcal{S} \) to hit \( \mathcal{F} \), 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 trade-offs 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 \( \mathcal{F} \). Another possibility is to consider the aggregate diameter \( D \) (the sum of diameters of connected components), or the aggregate resistance \( R \) (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 \( \mathcal{F} \) are not connected. It is not known whether \( D \) or \( R \) can be computed in randomized logspace, or whether USTCON can be decided in randomized logspace and time \( \tilde{O}(mD) \) (or \( \tilde{O}(mR) \)). Nevertheless, let us assume that \( D \) is given explicitly as input to our algorithm and ask whether USTCON has a \( \tilde{O}(mD) \) time–space trade-off. 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 trade-off of \( \tilde{O}(mD) \), even if \( D \) (the aggregate diameter) is given explicitly as input to the algorithm. More explicitly, there is a *family of graphs* with \( D = \Theta(\sqrt{n}) \) for which even if space \( n^3 \) is used by an FD algorithm, where \( \delta < 1/2 \), the time requirement remains \( \Omega(mD) \).

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 trade-offs 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 \( \Omega(n^3) \) is proved for a model that allows for an initial phase of distributing landmarks, but which 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 trade-offs 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 \( E[T(N, t)] \) denote the expected time until a walk discovers its \( v \)th distinct vertex, where \( N \leq n \).

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

\[
E[T(N, t)] = O\left( t + \frac{N^2}{d_{\text{min}}} \min\{N, d_{\text{max}}\} \right).
\]

This improves a result of [BF] by a factor of \( \log N \).

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 trade-offs. By carefully considering the work of [BKRU], it becomes apparent that this property is \( E[N(t)] \), 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 \( E[N(t)] \), the better the trade-off achieved. It is important to realize that the upper bound of \( E[N(t)] = O(\sqrt{t} \log n) \) [BKRU] can be improved, despite the fact that it is tight (up to the \( \log n \) factor), by introducing new parameters, such as \( d_{\text{min}} \).

The second point is a crucial observation made in [BF], which opened the way to the 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 \( E[N(t)] \), in particular, one gets better trade-offs.

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

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 \( 1/d_{\ast} \), then we would like \( \text{ave}[1/d_{\ast}] \) 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 trade-off that depends on \( d_{\text{min}} \), 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 emphasis 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 \( O(mR) \) time-space trade-off 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
trade-offs for USTCON are virtually the same for both versions of ID.

Recall from Section 1.1 that we need to specify a probability distribution $D$ (of ID, in our case) over the vertices of the input graph. Recall that $d_i$ denotes the degree of vertex $i$ and that $R = \sum_{i=1}^{n} 1/d_i$. ID assigns to vertex $i$ the probability $(d_i + m(d,R)^{-1})/3m$. Observe that $\sum_{i=1}^{n} d_i = 2m$ and $\sum_{i=1}^{n} (d_i + m(d,R)^{-1}) = 1$, so $\sum_{i=1}^{n} (d_i + m(d,R)^{-1})/3m = 1$, 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 $1 \leq p \leq p^2$. For these values, we will have $t = mR/p^2$, up to multiplicative terms that are polylogarithmic in $n$. Note that $t$ ranges between $O(mR)$ and $\Omega(mR)$, 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 $R$ in this representation (or a close approximation of it) can be done in time $O(m)$ and space $O(\log n)$. Now let us 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 $\sum_{i=1}^{j} (d_i + m(d,R)^{-1})$. On the fly, we give vertex $j$ the landmarks that have values that are between $\sum_{i=1}^{j} (d_i + m(d,R)^{-1})$ and $\sum_{i=1}^{j+1} (d_i + m(d,R)^{-1})$. The whole procedure takes time $O(m)$, 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 trade-offs 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 $u = (u, v)$ 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 $d_i$ of a vertex is the number of its neighbors, or equivalently the number of outgoing arcs. For an arc $e = (u, v)$, we define $s_e = mRd_e$ and $s_e$ to be the self-loops (arcs of the form $(v, v)$) to each vertex $v$. For simplicity we assume that $mRd_e$ is an integer for every vertex $v$. (The analysis that follows goes through virtually without change even if $mRd_e$ are nonintegers, by considering graphs with weighted arcs.) The nonisomorphic 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 $\pi_R$ (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 ourself 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 $S$ and $T$ 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 $v$, $P_{H}[T(u) \leq t] = \Omega(\log n)/p$, 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 $e$, $P_{H}[T(e) \leq t] = \Omega(\log n)/p$, 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 ensure (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 $S \in P_1$, $T \in P_2$, and walks of length $t$ that start 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 $(P_1, P_2)$ 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, it is 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 trade-off that is achievable as a function of $E_e[N_e(t)]$. In the lemma we introduce a parameter $k$ that shows 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 $k$, $1 \leq k \leq m$, 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 $t = O(m^2/kp^2)$ that satisfies conditions 1 and 2 above.

**Proof.** We assume that for some constant $c$ and every edge $e$, $E_e[N_e(t)] = O(\sqrt{t/k})$, and show that $P_e[T((v, u)) \leq t] \geq c \log_2/p$. 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

$$E_e[N_e(v, u)(t)] \leq P_e[T((v, u)) \leq t][1 + E_e[N_e(v, u)(t)]]$$

and, hence,

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

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

$$t = \frac{m^2}{k} \left(c'(c + 1) \log_n\right)^2,$$

condition 1 follows. For condition 2, observe that

$$E_e[N_e(t)] \leq P_e[T(v, u) \leq t][1 + E_e[N_e(t)]]$$

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 $m/k < t < mR$ 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 $t = O(mR/p^2)$ that satisfies conditions 1 and 2 above.

**Proof.** Substitute $k = mR$ in Lemma 5.

It remains to show that for $mR < t < mR$, $E_e[N_e(t)] = O(\sqrt{tR/m})$. 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 $e$, denoted by $R(u, e)$, is the voltage that develops in $u$ if a current of 1 amp is forced into $u$, and $e$ is grounded (by Ohm’s law). The effective resistance of a graph is defined as $R = \max_{u, v \neq e} R(u, e)$. 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 \( R < D \).

We define the virtual resistance of a simple graph by 
\[
\bar{R} = \sum_{e \in E} 1/d_e. \tag{For the purpose of this definition, assume that there are no isolated vertices.} \]

The virtual resistance is an easily computable parameter of the graph. It can be approximated with arbitrary precision in deterministic logspace and time \( O(m) \) (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, \( \bar{R} < 2\bar{R} \).

**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 \( R = R[u, u] \). 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 Kirchoff’s current law, the amount of current that enters \( u \) must 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 from \( w \).

Divide the vertices into two sets: \( V_1 \) and \( V_2 \), where \( V_1 \) contains the vertices that have at least \( d_{ij}/3 \) edges connecting to \( V_1 \), and \( V_2 \) contains the vertices that have fewer than \( d_{ij}/3 \) edges to \( V_1 \). We can choose a pair of antiparallel arcs in \( G \) as two antiparallel arcs in \( \bar{G} \). The direction of current along these edges is from \( v \) to \( w \), because \( \bar{v}_j \) has at least \( d_{ij}/3 \) edges connecting it to \( \{v_{i1}, \ldots, v_{in}\} \). The direction of current flow along these edges is from \( v_j \) to \( v_i \), because \( \bar{v}_j \) has the higher voltage. The sum of currents along these \( d_{ij}/3 \) 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 \( v_j \), carries a current of not more than \( 3/d_e \) amperes. Applying Ohm’s law, \( R_j \leq R_i + 3/d_e \), and using \( R_j \leq R_i \), we conclude that \( R_j \leq R_i + 3/d_e \). \( R_j \) will serve as an upper bound to the voltages of all vertices up to and including \( v_j \), and we can update \( i \leftarrow j \).

There exists \( j, 1 \leq j \leq n \), such that \( v_j \) has at least \( d_{ij}/3 \) edges connecting it to \( \{v_{i1}, \ldots, v_{in}\} \). Then by an argument similar to the above, there is some vertex \( v_l \) of \( G \) such that \( R_l \leq R_j + 3/d_e \). Of the vertices that have this property, choose \( v_l \) for which \( l \) is maximal. Observe that \( R_l \leq R_j + 3/d_e \), because \( R_j \leq R_i \). \( R_l \) will serve as an upper bound to the voltages of all vertices up to and including \( v_l \), and we can update \( i \leftarrow l \).

Neither case 1 nor case 2 above hold. Then we upper bound \( R_{i1} \leq R_i + 1 \) (this follows from the fact that if the virtual resistance of \( G \) is the same as the resistance between these vertices in \( G \), and the effective resistance between any two vertices of \( H \) is the same as the resistance between these vertices in \( G \), 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 $I_A \cup B_A$. Then the exit resistance of $A$ is $R_A = \max_{v \in I_A} \min_{v \in B_A} R[u,v]$. 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 originally played by aggregate resistance.

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

Proof. Let $e$ be the set of arcs incident with $I_A$. Then $R_A \leq |A|$. Let $|A| \geq 1$. Then $R_A \leq \sum_{v \in I_A} (d_v + s_v) = \sum_{v \in I_A} (d_v + mRd_v) \geq (m\tilde{R}) \sum_{v \in I_A} 1/d_v$. Define $\tilde{R} = \sum_{v \in I_A} 1/d_v$. Rearranging gives $R_A \leq (|A|m) \tilde{R}$.

To compute $R_A$, observe that $R_A \leq \max_{u,v \in I_A \cup B_A} R[u,v]$. Using Lemma 7 on the subgraph induced by $I_A \cup B_A$, and noting that $\sum_{v \in I_A} 1/d_v = R_A$ and that $\sum_{v \in I_A} 1/d_v \leq 1$ (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 $R_A \leq 9(\tilde{R} + 1)$.

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

2.4. Bounding the Expected Number of Returns

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

Lemma 10. Let $H$ be a digraph with $n$ vertices and $3m$ arcs, where each vertex has $d$ neighbors and $s$, self-loops as described above. Let $e$ be an arbitrary arc in $H$, and let $m<\tilde{R}<e<\tilde{R}$. Then $E_t[N(t)] \leq 10 + 6/\sqrt{tRm} \leq 16/\sqrt{tRm}$.

Proof. Let $A_e = \{(u,v) \mid E_s[N(u,v)] > s\}$ be the set of arcs from (the tail of) which random walks of length $t$ are expected to hit more than $s$ times, and let $A'_e$ be the complement of this set. Then $E_t[N(t)] \leq E_t[N(A'_e)] + s$. To compute $E_t[N(A'_e)]$, the expected number of times that $e$ is hit before $A'_e$ is reached, we consider the size of $A_e$.

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

Proof. Let $\bar{e}$ denote the arc antiparallel to $e$, and let $B_e = \{(u,v) \mid E_s[N(u,v)] > s\}$. We show that $(u,v) \in B_e$ if and only if $(v,u) \in A_e$. 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 $E_{(x,y)}[N(y,x)] = E_{(y,x)}[N(x,y)]$, and therefore an arc is in $B_e$ if and only if the antiparallel arc is in $A_e$.

To complete the proof of the claim, observe that by linearity of expectation, $|B_e| < t/s$. It follows that $|A_e| < t/s$.

Claim 12. The expected number of times that $e$ is traversed before exiting the set $A_e$ satisfies $E_t[N(A'_e)] \leq 10 + 9 |A_e| \tilde{R} m$.

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 $B_e$ that minimizes $R[w,u]$. From Lemma 9 we obtain that $R[w,u] \leq R_A \leq 9(1+|A_e| \tilde{R} m)$. Now consider an arc $(w,v) \notin A_e$ (such an arc must exist). Then $R(w,v) \leq 9(1+|A_e| \tilde{R} m) = 10 + 9 |A_e| \tilde{R} m$. When the walk reaches $v$ it must have left $A_e$.

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 $E_t[N(s,v)] \leq 10 + 9 |A_e| \tilde{R} m$.

Set $s = 3/\sqrt{tRm}$. Then $|A_e| \leq 1/\sqrt{3mR}$ and $E_t[N(A'_e)] \leq 10 + 3/\sqrt{tRm}$. Hence, $E_t[N(t)] \leq 10 + 6/\sqrt{tRm}$.

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 $d_\bar{e} + d_e > n-2$, then $\bar{e}$ and $\bar{e}$ 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 the 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 than 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 \( O \) 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 trade-off of \( O(mD) \). Our proof actually applies to \( l \)-fair algorithms for \( l < \sqrt{n}/2 \).

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 \( D \approx 2 \sqrt{n} + 3 \sqrt{n} = 5 \sqrt{n} \).

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 \( O(mD) = O(n^{3/2}) \), 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 trade-off of \( O(mD) \).

**Lemma 13.** On the 3-star graph, any FD algorithm that uses \( p \ll \sqrt{n}/\log n \) landmarks must make short random walks of length \( t = \tilde{O}(n^{1/2}/p) \) 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 \( 3p/\sqrt{n} \leq 1 \). 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 \( t \gg n \), \( P_{\mathcal{S}, \mathcal{T}}(T \leq t) \leq 12n^{3/2}/t^2 \).

**Proof.** Assume that \( P_{\mathcal{S}, \mathcal{T}}(T \leq t) \gg 12n^{3/2}/t^2 \). We shall compute \( E_{\mathcal{S}, \mathcal{T}}(T_{\mathcal{S}}) \) (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 \( \sqrt{n}/2 \). \( \mathcal{T} \) is the endpoint of one of these rays.

One way of computing \( E_{\mathcal{S}, \mathcal{T}}(T_{\mathcal{S}}) \) 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 \( E_{\mathcal{S}, \mathcal{T}}(T_{\mathcal{S}}) + E_{\mathcal{S}, \mathcal{T}}(T_{\mathcal{T}}) = 2n^{3/2}/3 \). Standard analysis of random walks on a line with \( \sqrt{n} \) edges shows that \( E_{\mathcal{S}, \mathcal{T}}(T_{\mathcal{S}}) = n \). Hence \( E_{\mathcal{S}, \mathcal{T}}(T_{\mathcal{T}}) = 2n^{3/2}/3 - n \).

The other way of computing \( E_{\mathcal{S}, \mathcal{T}}(T_{\mathcal{S}}) \) is based on the (false) assumption that \( P_{\mathcal{S}, \mathcal{T}}(T \leq t) \gg 12n^{3/2}/t^2 \). Observe that for any vertex \( u \), \( E_{\mathcal{S}, \mathcal{T}}(T_u) \leq n \), and hence by Markov’s inequality, \( P_{\mathcal{S}, \mathcal{T}}(T \geq 2n) \leq 1/n \). It follows that \( P_{\mathcal{S}, \mathcal{T}}(T \leq t + 2n) \gg 12/t^{3/2} \). 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 \( 6/n^{3/2} \) of hitting \( \mathcal{T} \). If follows that \( E_{\mathcal{S}, \mathcal{T}}(T_{\mathcal{S}}) \leq (t + 2n)n^{3/2}/6t \). For \( t \gg n \), we obtain that \( E_{\mathcal{S}, \mathcal{T}}(T_{\mathcal{S}}) \leq n^{3/2}/2 \), a contradiction (when \( n \) is sufficiently large).

It follows from the claim that whenever \( t \gg n \), 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 \( n^{3/2}/24 \).
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 \( \Delta \) th distinct vertex is \( O(k^3) \). They then conjectured that this bound can be refined to \( O(\Delta^3 + k+\Delta^3 \min(\Delta^2, d_{\text{max}})/d_{\text{min}}) \) and proved this bound up to a multiplicative logarithmic factor. The conjecture is known to hold if \( \Delta = O(n^\epsilon) \), by the results of Kahn et al. [KLNS]. We prove the conjecture for arbitrary graphs and arbitrary values of \( \Delta \). The proof combines ideas that appear in Section 2, some of which originate in the works of [Aldous, 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 \( d_{\ast} \) denote the minimum number of arcs incident with a vertex \((including self-loops)\), and let \( d_{\text{min}} \) denote the minimum number of vertices adjacent to a vertex. Let \( E[w,(u,v)] \) denote the expected time to discover \( \mathcal{A} \) arcs, where \( d_{\text{min}} d_{\ast} < \mathcal{A} < n \), for some positive constant \( c < 1 \) to be determined later. Then \( E[w,(u,v)] = O(d_{\text{min}}^2 d_{\ast}^2) \).

**Proof.** By Lemma 15, \( E[w,(u,v)] = O(1 + t/d_{\text{min}}^2 d_{\ast} \cdot d_{\text{min}}) \). Intuitively, in \( d_{\text{min}}^2 d_{\ast} \) steps, we expect any arc to be visited at most \( \mathcal{A}/d_{\text{min}} d_{\ast} \) times; hence \( \mathcal{A} \) distinct arcs are visited. We now make this intuition precise.

We want the expected time to visit \( \mathcal{A} \) arcs. Consider an arbitrary starting arc, and walks of length \( c \mathcal{A}^2 \) steps. For an unsuc-

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 \( [0, c \mathcal{A}^2/d_{\text{min}} d_{\ast}] \). 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 \( E[w,(e',c\mathcal{A}^2/d_{\text{min}} d_{\ast})] > 4 \mathcal{A}^2 c \mathcal{A}^2/d_{\text{min}} d_{\ast} \), 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 \( c \mathcal{A}^2/d_{\text{min}} d_{\ast} \) steps of the walk from which \( e' \) is selected. With probability \( 1/2 \), the walk is not successful (i.e., it visits less than \( \mathcal{A} \) distinct arcs). For an unsuccessful walk, there are at most \( \mathcal{A} k \) choices of \( j \) for which the number of subsequent returns of this walk to the arc \( e' \) selected is at most \( k \). Hence,

\[
P_E[w,(e',c\mathcal{A}^2/d_{\text{min}} d_{\ast}) < k] < \frac{1}{2} + \frac{k}{2} d_{\text{min}} d_{\ast}/c \mathcal{A}^2
\]

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 \( k = 8 \sqrt{3c \mathcal{A}^2/d_{\text{min}} d_{\ast}} \), and large enough \( c \) (e.g., \( c > 3^2 \cdot 2^2 \)), we obtain that \( P_E[w,(e',c\mathcal{A}^2/d_{\text{min}} d_{\ast}) < k] > \frac{1}{2}, \)

implying \( E[w,(e',c\mathcal{A}^2/d_{\text{min}} d_{\ast})] > k/2 = 4 \sqrt{3c \mathcal{A}^2/d_{\text{min}} d_{\ast}} \),
To contradict Lemma 15, ε in Lemma 17 should be small enough so that ε = 2^d min d a, the length of words considered in the proof of Lemma 17, would still be in the range required in Lemma 15. Hence we require that 3^1.2^3 d min d a \leq 3M^2/d min d a, which holds when ϵ < 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(N, М)] denote the expected time to discover either N vertices or М edges (whichever comes first), where М > N, and d min < N < en, for some positive constant ε < 1 to be determined later. Then E[T(N, М)] = O(N, М, d min).

**Proof.** Obtain a symmetric digraph H from G, by replacing each edge with two antiparallel arcs and by placing d a = \[N/2, N/2\] self-loops on each vertex. Then H has М > N, d arcs. On H, we want to discover 3. М + N < 4. d arcs, since then we either discover М of the original edges, or М + N distinct self-loops, impeding the discovery of N vertices. For ε < 1/4, the value of ε, we can apply Lemma 17 and obtain

\[E[T(4. H)] = O(M^2/d min d a).
\]

Proof of **Theorem 4.** We assume that N > d min, as otherwise the proof of the theorem is trivial. Hence we can apply Lemma 18 with \[M = \min\{N^2, M d \text{ max}\},\] where visiting M edges guarantees the visit of N vertices.

**5. THE ALGORITHMS SD AND MD**

The results 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 a = d min and obtain E[|N(t)|] \leq 2 + 2\sqrt{t/d min}. Note that when t \(d min^2\) the additive factor of 2 in the upper bound is dominated by the term 2\sqrt{t/d min}. This implies that Lemma 5 can be applied with k = (d min)^2, giving a value of t = O(m/d min^2), for a time–space trade-off of 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 a \(\geq m/n\) to obtain that we can set k = md min/n. We obtain t = O(md min n^2) and a time–space product of O(md min n^2).

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(mR). Consider three cliques of size roughly \(n - \sqrt{n}/3\), one containing S, and another containing T. Two of the cliques are connected by a path of length \(\sqrt{n}\). For this graph m = \(\Theta(n^2)\) and R = \(\Theta(\sqrt{n})\). Hence in logarithmic space, USTCON can be decided in time O(mR) = \(O(n^{5/2})\) 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 S and T are connected it is then necessary that some walk crosses the path from one cli-

**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 \(R = \sum_1/d\) as an important parameter related to random walks. Recent work has shed more light on the interrelation between the virtual resistance and random walks. It turns out that a more “appropriate” definition of virtual resistance is \(R = \sum_1/(d + 1)\). It can be shown that for any graph, \(R < 3\bar{d}\), and this bound is best possible, up to low order terms (e.g., for the path). This result implies automatic improvement of constants throughout Section 2. More interestingly, \(R\) provides a tight characterization of virtual 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].

**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.
- S, T—vertices in G.
- MD—minimum degree.
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.

ACKNOWLEDGMENTS

REFERENCES


