Trading Space for Time in Undirected s-t Connectivity

Andrei Z. Broder*
Anna R. Karlin†
Prabhakar Raghavan‡
Eli Upfal§

November 24, 1992

Abstract

Aleliunas *et al.* [1] posed the following question: "The reachability problem for undirected graphs can be solved in logspace and $O(mn)$ time [m is the number of edges and n is the number of vertices] by a probabilistic algorithm that simulates a random walk, or in linear time and space by a conventional deterministic graph traversal algorithm. Is there a spectrum of time-space trade-offs between these extremes?" We answer this question in the affirmative for sparse graphs by presenting an algorithm that is faster than the random walk by a factor essentially proportional to the size of its workspace. For denser graphs, our algorithm is faster than the random walk but the speed-up factor is smaller.

*DEC Systems Research Center, Palo Alto, CA 94301.

†DEC Systems Research Center, Palo Alto, CA 94301. Part of this research was done while the author was a research associate at Princeton University. Research supported in part by NSF grant DCR-8605961 and ONR contract N00014-87-K-0467.

‡IBM T.J. Watson Research Center, Yorktown Heights, NY 10598.

§IBM Almaden Research Center, San Jose, CA 95120 and Department of Applied Mathematics, Weizmann Institute of Science, Rehovot, Israel. Work at the Weizmann Institute supported in part by a Bat-Sheva de Rothschild Award and by a Revson Career Development Award.

1 Motivation and results

We consider the problem of s-t connectivity on an undirected graph (USTCON). Given a graph G with n vertices and m edges, and given two vertices s and t of G, we are to decide if s and t are in the same connected component. We are interested in space-bounded algorithms for USTCON, which is an important problem in the study of space-bounded complexity classes [3, 8]. Throughout this paper, we assume that our workspace takes the form of p registers, each capable of storing a $\log n$-bit number.

There are two well-known approaches to solving USTCON: via a deterministic graph search on G (e.g., depth-first search) and via a simulation of a random walk on G [1]. (The standard random walk on G is the stochastic process associated with a particle moving from vertex to vertex according to the following rule: if d_i is the degree of vertex i then the probability of a transition from vertex i to vertex j is $1/d_i$ if $\{i, j\}$ is an edge in G and 0 otherwise.)

The first approach can be implemented to run in time $O(m)$ using space $O(n)$. The latter requires space $O(1)$, and has been shown to decide USTCON in time $O(mn)$ with one-sided error (i.e., if s and t are in the same connected component, the algorithm outputs YES with probability at least 0.5; if they are in different components, the algorithm outputs NO). For both these algorithms, the product of time and space is $O(mn)$.

Given space that is insufficient for depth-first search, can we decide USTCON faster than via a random walk? More precisely, given space $p \leq n$, can we bridge the gap between the depth-first search and the random walk by devising an algorithm that runs in time $O(mn/p)$? Considering the time-space product achieved at the two extremes, this seems a likely conjecture.

In this paper we present an algorithm that runs in time $O(m^2 \log^5 n/p)$ in space $O(p)$. Therefore, for linear-sized graphs (i.e., $m = O(n)$), it achieves the bound conjectured above within a poly-log factor. For denser graphs, our algorithm does not achieve the bound; but it is faster than the random walk for m/p sufficiently small.

The informal description of the algorithm is as follows.

Algorithm STConn. Repeat $O(\log n)$ times

- Choose p random vertices according to the stationary distribution of the random walk on G. (The stationary distribution of the random walk is $\pi_v = d_v/(2m)$ where d_v is the degree of vertex v.) Call these vertices together with s and t, *leaders.*

- Repeat $O(\log n)$ times: Take a random walk of length $\tau_1 = O(m^2/p^2 \log^3 n)$ from each leader. If such walks connects two leaders, then mark them and all the other leaders known to be connected to them, as belonging to the same component. If at any point s and t are marked as being in the same component, then stop and report "connected."

Report "probably not connected."

End STConn.

A more precise description of the algorithm is given in Section 3. Clearly the space required is $O(p)$ and the time required is $O(m^2 \log^5 n/p)$. (The leaders are marked via a standard union-find algorithm.) Notice that this algorithm resembles standard search when $p = n$ and the random walk when $p = 0$. (However, throughout this paper we shall assume $p > 0$.)

There are three facts that must be proven in order to show that this algorithm works. The first is to show that a set of p random walks of length τ_1, one from each of the randomly chosen leaders, visits all the vertices of a connected graph with high probability. Otherwise an adversary could choose s and t among those vertices unlikely to be visited from the other leaders and conceivably foil the algorithm. In other words, we need to derive a bound on the expected time required by p parallel and independent random walks to cover the graph, a problem of interest in its own right. Typically, results about graph coverage rely heavily on the long-run behavior of the corresponding Markov chain and its convergence to a limit distribution. Here we must prove something about short-term behavior of the Markov chain and coverage of local neighborhoods in a graph.

The second fact to prove is that if s and t are in the same component and enough leaders are chosen within that component, then with high probability s and t are linked up after a small number of walks from each leader. Coverage of the graph as described above does not assure linkage, since s and t may be visited only by walks from two disjoint sets of leaders that are never linked. Furthermore all the vertices in G could be visited by the walks even with s and t in different components.

The third fact to show is that, with high probability, within $O(\log n)$ choices of the set of leaders, the component containing s and t gets enough leaders at least once.

To aid the intuition of the reader, let us consider the case when G is a simple path on n vertices. For p leaders chosen at random, with high probability, the maximum gap between two leaders is no more than $n \ln n/p$; the expected time to cover this maximum gap is $\Theta(n^2 \log^2 n/p^2)$. Hence $O(\log n)$ trials (random walks of length $O(n^2 \log^2 n/p^2)$ from each leader) will almost surely cover all the gaps between them for a total of $\Theta(n^2 \log^3 n/p)$ steps. Thus each leader "discovers" its closest neighbor leader in both directions, and therefore all leaders are marked as being in the same component.

Extending this technique to even 3-regular graphs requires considerably more complicated machinery and the general bound is weaker. (In particular, the walks need to have length $O(n^2 \log^5 n/p^2)$ and we need to try $O(\log n)$ choices for leaders.)

Our main results are:

Theorem 1 *Let G be a connected, undirected graph with n vertices and m edges. Let L be a subset of p vertices chosen at random according to the stationary distribution. Let $S_v(t)$ denote the set of vertices seen in a random walk of length t starting at v. Define the random variable C_p by*

$$C_p = \inf\{t \;:\; \bigcup_{l \in L} S_l(t) = V\},$$

that is, C_p is the time needed for p parallel random walks to visit all the vertices in the graph. Then

$$\mathbf{E}(C_p) = O\Big(\frac{m^2 \log^3 n}{p^2}\Big).$$

Theorem 2 *There is an algorithm that, given an undirected graph G with n vertices and m edges, and given two vertices s and t of G, decides* USTCON *with one-sided error using space p and time $O(m^2 \log^5 n/p)$. If s and t are in the same connected component, the algorithm outputs YES with probability $1 - O(n^{-1})$, otherwise it outputs NO.*

Remarks. The algorithm mentioned in Theorem 2 runs in time that is within a $\log^5 n$ factor of our target time-bound of $O(mn/p)$ for linear-sized graphs. It is conceivable that that a better analysis would lead to a similar algorithm with a lower overhead factor; however the path example shows that a factor of $\log^3 n$ is inherent to our approach.

2 Covering a graph with p random walks

In this section we derive an upper bound on the time taken by p parallel and independent walks to cover the graph (Theorem 1).

We denote by $\{v, w\}$ the undirected edge between vertices v and w and by $[v, w]$ its directed version. For the purposes of the proof, we need to look at the random walk in two ways: first, as a Markov chain $X(t)$ where each state is a vertex in G (the vertex process); second, as a Markov chain $Y(t)$ where each state is a directed edge (the edge process). The transition rule for the vertex process is that if $X(t) = v$, then $X(t+1)$ is equally likely to be any of the neighbors of vertex v. The edge process is defined by $Y(t) = [X(t-1), X(t)]$, $t \geq 1$. The stationary distribution of the vertex process, denoted π, is given by $\pi_v = d_v/(2m)$ where d_v is the degree of the vertex v, and the stationary distribution of the edge process, denoted π', is given by $\pi'_{[v,w]} = 1/(2m)$.

Let $N_v(u, T)$ (respectively $N_v([u, w], T)$) be the number of visits to the vertex u (respectively traversals of $[u, w]$) in a random walk of length T starting at v. (By definition, $N_v(v, 0) = 0$.) Let $S_v(T)$ (respectively $E_v(T)$) be the set of vertices (edges) visited in a random walk of length T starting at v. Finally, let $H_v(u)$ (respectively $H_v([u, w])$ be the first time the vertex u (the edge $[u, w]$) is encountered by a random walk starting from v. (We define $H_v(v)$ to be $1/\pi_v$, i.e. the return time to v.) For all of these random variables, a replacement of the subscript v with the subscript π (respectively $[v, w]$) denotes a random walk starting at the stationary distribution (respectively the directed edge $[v, w]$).

Lemma 1 *Let G be a connected, undirected graph on n vertices. Consider a random walk of length τ starting from the stationary distribution. Then for every directed edge $[v,w]$,*

$$\Pr\Big([v,w] \in E_\pi(\tau)\Big) \geq \frac{\mathbf{E}\Big(N_\pi([v,w],\tau)\Big)}{1+\mathbf{E}\Big(N_{[v,w]}([v,w],\tau)\Big)}$$

Proof: Clearly

$$\begin{aligned}
\mathbf{E}\Big(N_\pi([v,w],\tau)\Big) &= \sum_{1\leq t\leq \tau} \Pr\Big(H_\pi([v,w])=t\Big)\,\Big(1+\mathbf{E}\Big(N_{[v,w]}([v,w],\tau-t)\Big)\Big) \\
&\leq \Pr\Big(H_\pi([v,w])\leq \tau\Big)\,\Big(1+\mathbf{E}\Big(N_{[v,w]}([v,w],\tau)\Big)\Big)\,.
\end{aligned}$$

But $\Pr\Big(H_\pi([v,w])\leq\tau\Big) = \Pr\Big([v,w]\in E_\pi(\tau)\Big)$, yielding the lemma. □

Lemma 2 *Let G be a connected, undirected graph with n vertices and m edges. Then, for every directed edge $[v,w]$,*

$$\mathbf{E}\Big(N_{[v,w]}([v,w],\tau)\Big) \leq \frac{\tau}{2m} + \gamma\sqrt{\tau \ln n},$$

where γ is an absolute constant.

Proof: We consider the edge process Y_t. From standard results in renewal theory [9, Theorem 3.7.1] we obtain that

$$\mathbf{E}\Big(N_{[v,w]}([v,w],\tau)\Big) = \pi'_{[v,w]}\Big(\tau + \mathbf{E}\Big(H_{Y_{[v,w]}(\tau)}([v,w])\Big)\Big) - 1. \qquad (1)$$

Clearly

$$\mathbf{E}\Big(H_{Y_{[v,w]}(\tau)}([v,w])\Big) = \mathbf{E}\Big(H_{X_w(\tau)}(v)\Big) + \mathbf{E}\Big(H_v([v,w])\Big). \qquad (2)$$

Let $d(x,y)$ be the distance (the length of the shortest path) between two vertices x and y in G. Let c be a sufficiently large constant.

We first bound $\mathbf{E}\Big(H_{X_w(\tau)}(v)\Big)$ using the fact that $d(X_w(\tau), w)$ is not likely to be more than $c\sqrt{\tau \ln n}$, for some $c > 0$ such that $c\sqrt{\tau \ln n}$ is an integer. By the law of total probability

$$\begin{aligned}
&\mathbf{E}\Big(H_{X_w(\tau)}(v)\Big) = \\
&\quad \mathbf{E}\Big(H_{X_w(\tau)}(v) \mid d(X_w(\tau), v) \le c\sqrt{\tau \ln n}\Big) \Pr\Big(d(X_w(\tau), v) \le c\sqrt{\tau \ln n}\Big) \\
&\qquad + \mathbf{E}\Big(H_{X_w(\tau)}(v) \mid d(X_w(\tau), v) > c\sqrt{\tau \ln n}\Big) \\
&\qquad \times \Pr\Big(d(X_w(\tau), v) > c\sqrt{\tau \ln n}\Big).
\end{aligned} \tag{3}$$

Since $d(X_w(\tau), v) \le 1 + d(X_w(\tau), w)$, we obtain from the main result of [4] that

$$\begin{aligned}
\Pr\Big(d(X_w(\tau), v) > c\sqrt{\tau \ln n}\Big) &\le \Pr\Big(d(X_w(\tau), w) \ge c\sqrt{\tau \ln n}\Big) \\
&\le \sum_{x : d(w,x) \ge c\sqrt{\tau \ln n}} 2 \left(\frac{\pi_x}{\pi_w}\right)^{\frac{1}{2}} \exp\left(-\frac{d(w,x)^2}{2\tau}\right) \\
&\le 3n^{\frac{1}{2}} \exp\left(-\frac{c^2 \tau \ln n}{2\tau}\right) \le \frac{1}{n^3},
\end{aligned} \tag{4}$$

for a sufficiently large c.

For any two vertices x and y in the same component we can apply the bound implicitly proven in [1]

$$\mathbf{E}\Big(H_x(y)\Big) \le 2md(x,y) \le n^3. \tag{5}$$

Plugging equation (5) and equation (4) in equation (3) we obtain that

$$\mathbf{E}\Big(H_{X_w(\tau)}(v)\Big) \le 2cm\sqrt{\tau \ln n} + 2 \tag{6}$$

Turning to the second term of the right side of equation (2), we observe that

$$\mathbf{E}\Big(H_v([v,w])\Big) \le 2m + 1, \tag{7}$$

because the expected time to return to v given that v was left through an edge other than $[v,w]$ is at most $2m/(d_v - 1)$ and the expected number of returns to v before exiting through $[v,w]$ is $d_v - 1$. (The former fact follows from $2m/d_v = \mathbf{E}(H_v(v)) \geq (d_v - 1)/d_v \cdot \mathbf{E}(H_v \mid v \text{ not left via } [v,w])$.)

Combining equations (6), (7), and (2), we obtain that

$$\mathbf{E}\Big(H_{Y_{[v,w]}(\tau)}([v,w])\Big) \leq 2cm\sqrt{\tau \ln n} + 2m + 3.$$

Finally, from equation (1), because $\pi'_{[v,w]} = 1/(2m)$ for any edge $[v,w]$

$$\mathbf{E}\Big(N_{[v,w]}([v,w],\tau)\Big) \leq \frac{\tau}{2m} + c\sqrt{\tau \ln n} + O(1).$$

From here, the Lemma follows with an appropriate value for γ. $\Box$

Lemma 3 *Let G be a connected, undirected graph with n vertices and m edges. Let L be a set of p vertices (called leaders) in G chosen independently according to the stationary distribution. For every constant $c_1 > 0$ there exists a constant c_2 such that for every directed edge $[v,w]$ a set of p walks of length $c_2 m^2 \ln^3 n/p^2$, one from each of the leaders, satisfies*

$$\Pr\Big([v,w] \in \bigcup_{l \in L} E_l(c_2 m^2 \ln^3 n/p^2)\Big) \geq 1 - \frac{1}{n^{c_1}}\,.$$

Proof: For $p = O(\log n)$ the conclusion is obvious. For larger p we start from

$$\Pr\Big([v,w] \notin \bigcup_{l \in L} E_l(\tau)\Big) = \prod_{l \in L} \Pr\Big([v,w] \notin E_l(\tau)\Big),$$

and, since each vertex l is chosen independently according to the stationary distribution, Lemma 1 gives us a bound on $\Pr([v,w] \notin E_l(\tau))$. By Lemma 2 and because $\mathbf{E}\Big(N_\pi([v,w],\tau)\Big) = \tau/2m$, there exists a constant $c_3 > 0$ such that

$$\Pr\Big([v,w] \notin \bigcup_{l \in L} E_l(\tau)\Big) \leq \left(1 - \frac{c_3\sqrt{\tau}}{m\sqrt{\ln n}}\right)^p,$$

provided that $\tau = O(m^2 \log n)$. Now taking $\tau = c_2 m^2 \ln^3 n/p^2$ yields the result. $\Box$

Theorem 1 *Let $G = (V, E)$ be a connected, undirected graph with n vertices and m edges. Let L be a subset of p vertices chosen at random according to the stationary distribution. Let $S_v(t)$ denote the set of vertices seen in a random walk of length t starting at v. Define the random variable C_p by*

$$C_p = \inf\{t \ : \ \bigcup_{l \in L} S_l(t) = V\},$$

that is, C_p is the time needed for p parallel random walks to visit all the vertices in the graph. Then

$$\mathbf{E}(C_p) = O\left(\frac{m^2 \log^3 n}{p^2}\right).$$

Proof: Corollary of the previous lemma. □

In fact Lemma 3 implies the stronger result that the time needed for p parallel random walks to traverse every edge in the graph is $O(m^2 \log^3 n/p^2)$.

3 An algorithm for USTCON in $O(p)$ space

We now present the algorithm for USTCON using $O(p)$ space. As a subroutine, we use a standard Union/Find algorithm.

We use three constants, k_1, k_2, and k_3, in the description of the algorithm, which must be chosen sufficiently large. The choice of these constants determines the error probability of the algorithm. For ease of reference, we note here that k_1 is the constant c_2 of Lemma 3, Theorem 3, and Corollary 1, k_2 is the constant c_2 of Lemma 7, and k_3 is bound in Theorem 4.

algorithm STConn;
begin

do $k_3 \ln n$ times **begin**

Let L be a set of p elements of V, chosen independently at random according to the stationary distribution;
$L := L \bigcup \{s, t\}$;
(* $Set(l)$ are the leaders known to be connected to leader l *)
for every l in L **do** $Set(l) := l$;
do $k_2 \ln n$ times **begin**
for every l in L **do begin**
Take a random walk $X_l(t)$ of length $k_1 m^2 \ln^3 n/p^2$ from l;
At each step, **if** $X_l(t) \in L$ **then**
$Union(Find(X_l(t)), Find(l))$;
end ;
end ;
if $Find(s) = Find(t)$
then return ("YES: s and t are connected")

end ;

return ("NO: s and t don't seem to be connected")
end .

Theorem 2 *Given an undirected graph G with n vertices and m edges, and given two vertices s and t of G, the algorithm* **STConn** *decides* USTCON *with one-sided error using space $O(p)$ and time $O(m^2 \log^5 n/p)$.*

Proof: Choosing a random set of p vertices according to the stationary distribution can be done in $O(m)$ steps using $O(p \log n)$ random bits and $O(p)$ space. Since, only $O(p)$ space can be used to store L, and do lookups on, a binary search tree, or perfect hash function must be used. (Constructing a perfect hash function for storing L requires expected time $O(p)$ [6].) If the unions are weighted and each union causes path compression on all elements of the set, then each find has cost $O(1)$. Since at most $O(n)$ non-trivial unions are performed, the cost of all the unions is $O(n \log n)$. Performing all

$O(\log n)$ random walks of length $O(m^2 \log^3 n/p^2)$ takes time $O(m^2 \log^4 n/p^2)$ per leader for a total of $O(m^2 \log^4 n/p)$ time. Since this is also the total number of finds and lookups performed, this is the running time of each execution of the outermost loop. $\square$

Remark: Note that this algorithm is easily parallelizable using p processors and $O(p)$ space. The parallel hashing scheme described in [7] can be used to implement a parallel version of this algorithm that runs on p processors, $n^\epsilon \leq p \leq n^{1-\epsilon}$, $\epsilon > 0$, that are connected by a bounded degree network. Briefly, storing the leader set using parallel hashing allows for the p processors to execute parallel unions and parallel finds in time $O(p^{\epsilon'})$ for any $\epsilon' > 0$, and consequently the random walks from each of the leaders can be executed in parallel. The resulting parallel implementation of the **STConn** algorithm runs in time $O(m^{2+\epsilon''}/p^2)$ $(\epsilon'' > \epsilon')$.

4 The correctness of STConn

Because our algorithm has one-sided error, it suffices to analyze its correctness in the case when s and t are in the same component of G. If G is actually connected, the results of section 2 show that, in one pass through the inner loop of **STConn**, every edge is traversed with high probability. From this, it is possible to deduce that every leader is discovered by some leader. As mentioned earlier, however, this is not enough to prove that s and t become linked. The rest of this section shows that s and t will be "linked up" with high probability after $O(\log n)$ passes through the inner loop.

Theorem 3 *Let G be a connected, undirected graph with n vertices and m edges. Let L be a set of p leaders, each chosen at random according to the stationary distribution. Then for any $c_1 > 0$ there is a constant $c_2 > 0$ such that*

$$\Pr(L \cap S_{[v,w]}(c_2 m^2 \ln^3 n/p^2) \neq \emptyset) \geq 1 - \frac{1}{n^{c_1}},$$

where $S_{[v,w]}(T)$ denotes the set of distinct vertices visited in a T step random walk starting at $[v,w]$.

Proof: The proof is very similar to that of Lemma 3. As before the case $p = O(\log n)$ is trivial.

Let e be a directed edge chosen uniformly at random. By a proof virtually identical to that of Lemma 1,

$$\Pr(e \in E_{[v,w]}(\tau)) \geq \frac{\mathbf{E}(N_{[v,w]}(e,\tau))}{1 + \mathbf{E}(N_e(e,\tau))}.$$

Obviously, if e is chosen uniformly at random then

$$\mathbf{E}(N_{[v,w]}(e,\tau)) = \frac{\tau}{2m}.$$

By Lemma 2

$$\mathbf{E}(N_e(e,\tau)) \leq \frac{\tau}{2m} + \gamma\sqrt{\tau \ln n}.$$

Hence, for e chosen uniformly at random, there exists a constant c_3 such that

$$\Pr(e \in E_{[v,w]}(c_2 m^2 \ln^3 n/p^2)) \geq c_3 \frac{\ln n}{p},$$

provided that $P = \Omega(\log n)$.

In order to choose a leader according to the stationary distribution, one can choose a directed edge e uniformly at random and let the leader be the head of e. Since the probability of reaching a leader is greater than or equal to the probability of traversing the edge chosen to determine it, we obtain that

$$\Pr(L \cap S_{[v,w]}(c_2 m^2 \ln^3 n/p^2) = \emptyset) = (1 - \Pr(e \in E_{[v,w]}(c_2 m^2 \ln^3 n/p^2)))^p \leq \frac{1}{n^{c_1}},$$

for a sufficiently large c_2. □

Corollary 1 *Let G be a connected, undirected graph with n vertices and m edges. Let L be a set of p leaders chosen at random according to the stationary distribution. Then for any $c_1 > 0$ there is a constant $c_2 > 0$ such that*

$$\Pr(L \cap S_s(c_2 m^2 \ln^3 n/p^2) \neq \emptyset) \geq 1 - \frac{1}{n^{c_1}}.$$

and

$$\Pr(L \cap S_t(c_2 m^2 \ln^3 n/p^2) \neq \emptyset) \geq 1 - \frac{1}{n^{c_1}}.$$

□

Let L be any set of p leaders. We say the set L is *good* if for an absolute constant k_1 (determined in Lemma 4 below) the following two properties hold:

1. The probability that a set of p independent random walks of length $\tau = \frac{1}{2}k_1 m^2 \ln^3 n/p^2$, one from each leader in L, traverses every edge in G is at least $1 - 1/n^3$.

2. For every edge $[v, w] \in G$, the probability that a random walk of length τ starting from $[v, w]$ visits some leader in L is at least $1 - 1/n^3$.

Lemma 4 *Let G be a connected, undirected graph with n vertices and m edges. Let L be a set of p leaders chosen uniformly at random according to the stationary distribution. Then* $\Pr(L \text{ is good}) \geq 1 - 2/n$.

Proof: Say that a set of random walks, one from each of the leaders, is unsuccessful for $[v, w]$ if $[v, w]$ is not visited by any of them. Letting $c_1 = 6$ in Lemma 3 , we see that at most $1/n^3$ of the possible leader sets can have probability greater than $1/n^3$ of yielding unsuccessful random walks for any fixed $[v, w]$. Similarly, letting $c_1 = 6$ in Theorem 3, we see that at most $1/n^3$ of the possible leader sets have probability greater than $1/n^3$ of remaining undiscovered in a random walk of length τ from any fixed edge $[v, w]$. The probability that a leader set is not good is bounded by the sum of the probabilities that it isn't good because it violates properties 1 or 2. Since there are less than $n^2/2$ edges, the probability that a leader set is bad is bounded by $1/n$. The constant k_1 is determined by the requirements of Lemma 3 and Theorem 3. $\Box$

Lemma 5 *Let G be a connected, undirected graph with n vertices and m edges. Let L be a set of p leaders chosen uniformly at random according to the stationary distribution. Suppose that L is a good set of leaders. Let A and B be a partition of L into two nonempty subsets. Consider a random walk of length 2τ from each of the leaders in L. Then the probability that some leader in A is visited from some leader in B or vice versa is greater than 1/18.*

Proof: (Unless stated otherwise, all edges referred to in this proof are directed.) We assign to each edge in the graph two labels: a "To" label T and a "From" label F. These labels are subsets of the set $\{A, B\}$. By definition, $A \in T(e)$ (respectively $B \in T(e)$) if the probability that e is visited by a walk of length τ emanating from one of the leaders in A (respectively a walk from one of the leaders in B) is at least $1/3$. Analogously, $A \in F(e)$ (respectively $B \in F(e)$) if the probability that some leader in A (respectively B) is visited in a random walk of length τ starting from e is at least $1/3$.

Properties 1 and 2 of good leader sets imply that for each edge neither label is empty. We now consider four cases:

1. There is some edge $[v, w]$ with $A \in F([v, w])$ and $B \in T([v, w])$ or vice versa.

   Then with probability $\geq 1/3$ edge $[v, w]$ is visited by one of the random walks of length τ originating in A and with probability $\geq 1/3$ a leader in B is visited in the remaining at least τ steps. Hence, with probability $\geq 1/9$ a leader in B is visited from a leader in A.

   After eliminating this case the only remaining possibility is that for every edge $F([v, w]) = T([v, w]) = \{A\}$ or $F([v, w]) = T([v, w]) = \{B\}$.

2. There is some undirected edge $\{v, w\}$ such that $F([v, w]) = T([v, w]) = \{A\}$, and $F([w, v]) = T([w, v]) = \{B\}$.

   Then with probability $\geq 1/3$, $[v, w]$ is visited by one of the walks of length τ originating in A and hence the vertex v is visited by one of these walks with probability $\geq 1/3$. Since a leader in B is visited from $[w, v]$ in τ steps with probability $\geq 1/3$, a leader in B is visited from v in τ steps with probability $\geq 1/3$. Hence with probability $\geq 1/9$ a leader in B is visited from a leader in A.

3. No label in the graph contains A or no label in the graph contains B.

   Without loss of generality, consider the first of the two conditions. Then every edge directed towards leaders in A, has a "To" label of B. Therefore, with probability $\geq 1/3$, each such edge is visited by one of the random walks of length τ originating in B and a leader in A is immediately visited. Hence, with probability $\geq 1/3$, a leader in A is visited from a leader in B.

4. For every undirected edge $\{v, w\}$, we have either $T([v, w]) = F([v, w]) = T([w, v]) = F([w, v]) = \{A\}$ or we have $T([v, w]) = F([v, w]) = T([w, v]) = F([w, v]) = \{B\}$

   Since case 3 does not hold and the graph is connected, there must be a vertex v that is simultaneously the endpoint of some all-A labeled edge and some all-B labeled edge. Assume without loss of generality that at least $1/2$ of the undirected edges with one endpoint at v have all their labels equal to B. Then since some edge $[w, v]$ has an A T-label, with probability $\geq 1/3$ v is visited in the first τ steps of the random walks originating at A. Since the majority of edges leaving v have a B F-label, with probability $\geq 1/2$ one of these edges will be traversed and then with probability $\geq 1/3$, a leader in B will be reached during the remaining at least τ steps. Hence with probability $\geq 1/18$ a leader in B is visited from a leader in A.

□

We say that a subset of leaders forms a component, if during some prior phase of the algorithm, they have all been connected up with one another. During a particular phase, we say that a component C is *successful* if it discovers some other component or some other component discovers it. The previous lemma proves, that if the leader set is good, every component has probability at least $1/18$ of being successful. The next lemma shows that the number of separate components decreases exponentially with the number of phases.

Lemma 6 *Let G be a connected, undirected graph with n vertices and m edges. Let L be a set of p leaders chosen uniformly at random according to the stationary distribution. Suppose that L is a good leader set. Let N_i be the number of components after the ith phase. Then there exist constants α and β, with $0 < \alpha, \beta < 1$, such that if $N_i > 1$ then*

$$\Pr(N_{i+1} > \beta N_i) \leq \alpha.$$

Proof: Plainly, N_{i+1} equals N_i minus the number of nonredundant links formed in phase i. Since the number of such links formed in phase i exceeds one half the number of successful components, and the previous lemma shows that the probability that a component is successful is at least 1/18,

$$\mathbf{E}(\text{number of links formed in phase } i) \geq \frac{1}{2 \cdot 18} N_i.$$

Hence,

$$\mathbf{E}(N_{i+1}) \leq (1 - \frac{1}{36}) N_i$$

and so there is a positive constant $\beta < 1$ such that

$$\Pr(N_{i+1} > \beta N_i) \leq \alpha.$$

□

Lemma 7 *Let G be a connected, undirected graph with n vertices and m edges. Let L be a set of p leaders chosen uniformly at random according to the stationary distribution. Suppose that L is a good leader set. Let N_i be the number of components after the ith phase. Then for any constant $c_1 > 0$, there is a constant $c_2 > 0$ such that*

$$\Pr(N_{c_2 \ln n} > 1) \leq \frac{1}{n^{c_1}}.$$

Proof: We say that a phase is successful if $N_{i+1} \leq \beta N_i$. Since the leader set is fixed and good, successive phases are independent (the random walks are independent), and by the previous lemma, phase i has probability greater than $1 - \alpha$ of being successful for each i. But the probability that $N_{c_2 \ln n}$ is greater than one is bounded by the probability that there are fewer than $\ln_{1/\beta} n$ successful phases out of $c_2 \ln n$ phases. This in turn is bound by the probability that there are fewer than $\ln_{1/\beta} n$ successes in $c_2 \ln n$ Bernoulli trials with probability greater than $1 - \alpha$ of success, which by Chernoff's bound is less than $1/n^{c_1}$, for appropriately chosen c_2. □

Theorem 4 *The algorithm* **STConn** *decides* USTCON *using space $O(p)$ and time $O((m^2 \log^5 n)/p)$ with one-sided error. If s and t are in the same connected component, the algorithm fails to output YES with probability $O(n^{-1})$; if s and t are in different components, it outputs NO.*

Proof: If the graph consists of a single connected component, then we need only consider one execution of the outer loop of the algorithm, wherein the algorithm can fail to output YES when it should if either the leader set is not good, or the leader set is good, but the number of components did not reduce to 1. By Lemma 4, the former has probability at most $1/n$ and by Lemma 7 the latter, when choosing the constant k_3 appropriately, has probability at most $1/n$ and so the theorem follows in this case.

The other case is when s and t are in a single component C containing $\tilde{n}$ vertices and $\tilde{m}$ edges. If $m^2/p^2 > \tilde{m}\tilde{n}$ or $\tilde{n} < \ln^{3/2} n$, then in $k_3 \ln n$ random walks of length $k_1 m^2 \ln^3 n/p^2$ starting from s, the vertex t will be seen with overwhelming probability, since the expected cover time of the component is bounded by $2\tilde{m}\tilde{n}$ [1].

Otherwise, if $m^2/p^2 < \tilde{m}\tilde{n}$, the algorithm can fail to output YES when it should if either none of the $c_0 \ln n$ selections of leaders include enough leaders that are in the component C or if some selection of leaders includes enough leaders in C, but the associated random walks do not succeed in connecting s to t. For the latter case, we observe that, in each of the $c_0 \ln n$ executions of the outer loop of the algorithm, the expected number of leaders that are chosen from C is $\tilde{p} = p\tilde{m}/m$. If $\tilde{p}/2$ leaders are indeed chosen from C, then since

$$\tau = \frac{c_3 m^2 \ln^3 n}{p^2} = \frac{c_3 \tilde{m}^2 \ln^3 n}{\tilde{p}^2},$$

the analysis given for a single connected graph on $\tilde{n}$ vertices and $\tilde{m}$ edges with $\tilde{p}$ leaders yields a failure probability of $O(\tilde{n}^{-1}) = o(1)$. To bound the probability that a leader selection is not sufficiently dense, we note that the probability that fewer than $\tilde{p}/2$ leaders are chosen from C is bounded by the probability of fewer then $\tilde{p}/2$ successes in p trials with probability $\tilde{m}/m$ of success. By standard bounds, this probability is at most

$$e^{-\frac{1}{8}p\frac{\tilde{m}}{m}} \leq e^{-\frac{1}{8}\sqrt{\frac{\tilde{m}}{\tilde{n}}}} \leq c,$$

for some constant $c < 1$ (since $m/p < \sqrt{\tilde{m}\tilde{n}}$). Therefore, the probability that a single execution of the outermost loop fails is bounded by $c + o(1)$, and hence the overall probability of failure is bounded by $O(n^{-1})$, for a sufficiently large constant k_3. $\square$

5 Open problems

Can the bound on the parallel cover time given in Theorem 1 be improved? Note that we bound the cover time for all vertices by bounding the cover time for all edges. It is not clear that this is necessary.

Theorem 2 shows that for p slightly larger than the average degree m/n, our algorithm runs faster than the random walk. Devising an algorithm that runs in time $O(mn \log^k n/p)$ is perhaps the most interesting open problem.

There is no fundamental reason why our upper bound is the best possible. We thus hope that this work will spark interest in proving a time-space tradeoff for USTCON, even in a restricted model of space-bounded computation such as the JAGs of Cook and Rackoff [5]. Beame *et al.* [2] give a number of time-space tradeoffs for structured models based on automata that traverse graphs. For one natural variant that admits implementations of our algorithms, they show that the product of time and space is $\Omega(n^2)$ for d-regular graphs ($d \geq 3$), and is $\Omega(mn)$ for non-regular graphs.

Acknowledgement

We are grateful to Lyle Ramshaw for a thorough reading of the manuscript and many useful comments and corrections. We also thank Larry Ruzzo and Martin Tompa for their many comments and suggestions, and also the anonymous referees.

References

[1] 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 *20th Annual Symposium on Foundations of Computer Science*, pages 218–223, San Juan, Puerto Rico, October 1979.

[2] P. Beame, A. Borodin, P. Raghavan, W.L. Ruzzo, and M. Tompa. Time-space tradeoffs for undirected graph traversal. Revised version submitted for publication, 1992.

[3] A. Borodin, S. A. Cook, P. W. Dymond, W. L. Ruzzo, and M. Tompa. Two applications of complementation via inductive counting. In *Third Annual Conference on Structure in Complexity Theory*, pages 116–125, Washington, D.C., June 1988. To appear in *SIAM Journal on Computing*.

[4] T.K. Carne. A transmutation formula for markov chains. *Bull. Sc. Math., 2e série*, 109:399–405, 1985.

[5] S. A. Cook and C. W. Rackoff. Space lower bounds for maze threadability on restricted machines. *SIAM Journal on Computing*, 9(3):636–652, 1980.

[6] M. L. Fredman, J. Komlos, and E. Szemeredi. Storing a sparse table with $O(1)$ worst case access time. *Journal of the ACM*, 31(3):538–544, July 1984.

[7] A.R. Karlin and E. Upfal. Parallel hashing: An efficient implementation of shared memory. *Journal of the ACM*, 35(4):876–892, October 1988.

[8] H. R. Lewis and C. H. Papadimitriou. Symmetric space-bounded computation. *Theoretical Computer Science*, 19:161–187, 1982.

[9] S. Ross *Stochastic Processes.* Wiley, 1983.