HAL
open science

# A $\tilde{O}(n^2)$ Time-Space Trade-off for Undirected s-t Connectivity

Adrian Kosowski

▶ **To cite this version:**

Adrian Kosowski. A $\tilde{O}(n^2)$ Time-Space Trade-off for Undirected s-t Connectivity. SODA - 24th ACM-SIAM Symposium on Discrete Algorithms, Jan 2013, New Orleans, United States. pp.1873-1883. hal-00685373v3

**HAL Id: hal-00685373**

**https://inria.hal.science/hal-00685373v3**

Submitted on 10 Jul 2012

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Faster Walks in Graphs: A $\widetilde{O}(n^2)$ Time-Space Trade-off for Undirected $s$-$t$ Connectivity

Adrian Kosowski*

**Abstract**

In this paper, we make use of the Metropolis-type walks due to Nonaka *et al.* (2010) to provide a faster solution to the $\mathscr{S}$-$\mathscr{T}$-connectivity problem in undirected graphs (USTCON).

As our main result, we propose a family of randomized algorithms for USTCON which achieves a time-space product of $S{\cdot}T = \widetilde{O}(n^2)$ in graphs with $n$ nodes and $m$ edges (where the $\widetilde{O}$-notation disregards poly-logarithmic terms). This improves the previously best trade-off of $\widetilde{O}(nm)$, due to Feige (1995). Our algorithm consists in deploying several short Metropolis-type walks, starting from landmark nodes distributed using the scheme of Broder *et al.* (1994) on a modified input graph. In particular, we obtain an algorithm running in time $\widetilde{O}(n+m)$ which is, in general, more space-efficient than both BFS and DFS.

We close the paper by showing how to fine-tune the Metropolis-type walk so as to match the performance parameters (e.g., average hitting time) of the unbiased random walk for any graph, while preserving a worst-case bound of $\widetilde{O}(n^2)$ on cover time.

**Keywords:** undirected $\mathscr{S}$-$\mathscr{T}$ connectivity, time-space trade-off, graph exploration, Metropolis-Hastings walk, parallel random walks.

*Inria Bordeaux Sud-Ouest, 33400 Talence, France. E-mail: adrian.kosowski@inria.fr

# 1 Introduction

In the *undirected $\mathscr{S}$-$\mathscr{T}$ connectivity problem* (USTCON), the input to the algorithm is an undirected graph $G = (V, E)$ with $n$ vertices and $m$ edges. Two of the vertices of the graph, $\mathscr{S}, \mathscr{T} \in V$, are distinguished. The goal is to determine whether $\mathscr{S}$ and $\mathscr{T}$ belong to the same connected component of $G$. USTCON has a spectrum of applications in various areas of computer science, ranging from tasks of network discovery to computer-aided verification. The problem has also made its mark on complexity theory, most famously, playing a central part in the rise and eventual collapse of the complexity class SL.

The time complexity of algorithms for USTCON depends on the amount of space available to the algorithm. Given $\widetilde{\Theta}(n)$ space, USTCON can be solved deterministically in time $O(m)$ by fast algorithms such as BFS or DFS. Given $\Theta(\log n)$ space, the problem can still be solved deterministically [21] in polynomial time. However, in this case the fastest known solutions are randomized. Aleliunas *et al.* [2] proposed a log-space algorithm with bounded error probability, which consists in running a random walk, starting from node $\mathscr{S}$ for $O(nm)$ steps, and testing if node $\mathscr{T}$ has been reached.

The study of the interplay between the space complexity $S$ and the time complexity $T$ of randomized algorithms for USTCON was initiated by Broder *et al.* [10]. They observed that both BFS/DFS, and the random walk, admit the same time-space trade-off of $T = \widetilde{O}(\frac{mn}{S})$, and investigated whether there exist algorithms with such a trade-off for an arbitrary choice of $S$, $c \cdot \log n \leq S \leq n$, where $c > 0$ is some model-dependent constant. After a sequence of papers relying on the deployment of many short random walks, this question was eventually settled in the affirmative by Feige [14], who proposed a family of algorithms which achieve such a time-space trade-off in the whole of the considered range of space bounds.

The main result of this paper is an improved time-space trade-off for USTCON. Since the cover time of the random walk is precisely $\Theta(nm)$ for some graphs, any improvement with respect to Aleliunas *et al.* [2] or Feige [14] requires a refinement of the performed walk on graphs. Instead of the random walk, we make use of the Metropolis-Hastings walk on graphs, with weighting proposed by Nonaka *et al.* [20]. This walk covers any undirected graph in $\widetilde{O}(n^2)$ steps, but its transition probabilities rely on knowledge of the degrees of neighboring nodes at every step. We start the technical sections of this paper with an explicit implementation of the walk from [20] using the Metropolis sampling algorithm from [18]. This yields a solution to USTCON in $\widetilde{O}(n^2)$ time and logarithmic space. Our contribution lies in completing this quadratic time-space trade-off for larger bounds on the space complexity of the algorithm. The main technical difficulty concerns overcoming problems with short runs of the Metropolis-Hastings walk, which sometimes exhibits inferior behavior to the random walk in terms of the speed of discovering new nodes.

For the entire range of space bounds ($c \cdot \log n \leq S \leq n$), we propose algorithms running in time $T = \widetilde{O}(\max\{\frac{n^2}{S}, m\})$. In other words, we obtain $T = \widetilde{O}(\frac{n^2}{S})$ for $S \leq \frac{n^2}{m}$, and $T = \widetilde{O}(m)$ for $S > \frac{n^2}{m}$. (Note that $T = \Omega(m)$ is a lower bound on execution time for any algorithm for USTCON, regardless of the space bound.) In particular, we prove that USTCON can be solved in time $\widetilde{O}(m)$ using space $O(\frac{n^2}{m})$, which is, in general, less than the space requirement of BFS/DFS.

All of the considered algorithms for USTCON are randomized (in the Monte Carlo sense), with bounded probability of one-sided error. This means that the positive answer "connected" may only be reached by the algorithm when $\mathscr{S}$ and $\mathscr{T}$ belong the same connected component of $G$, whereas the negative answer "not connected" signifies that, with probability at least 2/3, $\mathscr{S}$ and $\mathscr{T}$ belong to different components of $G$.

### 1.1 Related work

Much of the work on undirected $\mathscr{S}$-$\mathscr{T}$ connectivity has focused around its role as the fundamental complete problem for the symmetric log-space complexity class (SL). A survey of other important problems identified as belonging to the class SL, such as simulating symmetric Turing machines, and testing if a graph is bipartite, is provided in AGCR. A major line of study concerned determining the minimum space required to solve USTCON deterministically. The bound on the required space was reduced, over several decades, from the $O(\log^2 n)$ bound given by Savitch's theorem [22], through $O(\log^{3/2} n)$ [23], and $O(\log^{4/3} n)$ [4]. Finally, in 2004, Reingold's [21] new construction of universal graph exploration sequences provided the first log-space algorithm for USTCON, showing that SL=L. Befor Reingold's paper, Nisan [19] had shown a deterministic algorithm for USTCON running in polynomial time and $O(\log^2 n)$ space. Borodin *et al.* [9] proposed a log-space Las-Vegas type algorithm for USTCON (with no-error) which runs in expected polynomial time.

When considering randomized algorithms with bounded one-sided error, the unbiased random walk was shown to solve USTCON in $O(\log n)$ space and $\widetilde{O}(mn)$ time by Aleliunas *et al.* [2]. Several years later, Broder *et al.* [10] proposed a family of algorithms based on short random walks starting from landmark nodes. Relying on landmarks chosen on the set of nodes according to the stationary distribution of the walk, they achieved a time-space trade-off of $T = \widetilde{O}(\frac{m^2}{S})$. Subsequent algorithms from the literature [5, 14] make use of different landmark distribution schemes. Barnes and Feige [5] achieve a trade-off of $T = \widetilde{O}(\frac{m^{1.5}n^{0.5}}{S})$ by using a mixed landmark distribution scheme, which places half of the landmarks according to the stationary distribution of the random walk, and half according to the uniform distribution on nodes. Feige [14] introduces the inverse distribution scheme, which likewise places half of the landmarks according to the stationary distribution of the random walk, and the other half according to the inverse of node degrees. He achieves a time-space trade-off of $T = \widetilde{O}(\frac{mn/\delta}{S})$ in general, where $\delta$ is the minimum degree of the graph. Thus, the trade-off of $T = \widetilde{O}(\frac{n^2}{S})$ is reached for the case of (nearly) regular graphs.

Undirected $\mathscr{S}$-$\mathscr{T}$ connectivity is a special case of the more general reachability problem in directed graphs (STCON), which is a complete problem for the class NL. STCON can also be solved deterministically in $O(\log^2 n)$ space using Savitch's theorem [22]. So far, it has resisted fast solutions in small space. This problem was extensively studied in different variants of a model of computation based on Jumping Automata on Graphs (JAG-s). The memory of a JAG is organized in the form of $P$ pebbles placed in the graph and $Q$ states of the automaton, with space defined as $S = P \log n + \log Q$. Cook and Rackoff [12] show a way of solving STCON in the JAG model deterministically in $O(\log^2 n)$ space, and also prove an almost matching lower bound on space of $\Omega(\log^2 n/\log\log n)$. This lower bound is also known to apply to randomized JAG-s running in slightly super-polynomial time [8]. Gopalan *et al.* [15] propose a family of algorithms for STCON based on short random walks, whose runtime increases from $O(n^{\log n})$ to $O(n^n)$ as space decreases from $O(\log^2 n)$ to $O(\log n)$.

Finally, we remark on recent developments in the area of graph exploration with biased random walks. Ikeda *et al.* [17] and Nonaka *et al.* [20] studied possible adjustments to the transition matrix of the walk based on the availability of local topological information (otherwise known as "look-ahead"). In general, the idea of these approaches is to increase the probability of transition to a node of lower degree. The former paper introduces a new type of walk, called the $\beta$-walk, whose transition matrices are biased so that transition from a node to its neighbor of degree $d$ is proportional to $d^{-\beta}$. Such a walk was shown to visit all nodes of the graph in $O(n^2 \log n)$ steps in expectation for an optimal choice of parameter $\beta = 1/2$. Nonaka *et al.* [20] later used the key lemmas from this work to prove an analogous result for a walk with a modified transition matrix, which fits into the class of Metropolis-Hastings walks. This walk is

the starting point for considerations in our paper. A somewhat different approach was adopted by Berenbrink *et al.* [7], who show that a random walk with the additional capability of marking one unvisited node in its neighborhood as visited can be used to speed up exploration.

### 1.2 Overview of the paper

The organization of the technical parts of the paper is the following. In Section 2, we recall the definition of the Metropolis-Hastings walk and provide its efficient implementation using the Metropolis algorithm. In this way, given a representation of graph $G$, each step of the walk can be simulated by a procedure running in $\widetilde{O}(1)$ time and using $\Theta(\log n)$ bits of space.

We subsequently identify the key properties of the unit-potential Metropolis-Hastings walk, denoted $RW(G_1)$, which allow it to be used as a replacement for the (unbiased) random walk on $G$, denoted $RW(G)$, in algorithms solving USTCON. The major difference between these types of walks is that a short random walk $RW(G)$ has the desirable property of *low edge-return rate*, i.e., each edge of the graph is visited $O(\sqrt{t})$ times in expectation during $t$ steps of the walk (for sufficiently small $t$). However, no analogous property hold for the Metropolis-Hastings walk. In fact, on some graphs (e.g., the *glitter star* defined in [20]), the Metropolis-Hastings walk $RW(G_1)$, will in expectation discover only $O(1)$ edges during $t$ steps of the walk, visiting each of these edges $\Omega(t)$ times (for any choice of $t \leq n$). We overcome this problem in two stages:

- In Section 2 we prove that in a graph of maximum degree $\Delta$, the Metropolis-Hastings walk $RW(G_1)$ begins to achieve a *low node-return rate* starting from a threshold length of $\Delta^2$ steps: a Metropolis-Hastings walk of length $t$, $\Delta^2 < t < n^2$, visits each node of the graph $O(\sqrt{t})$ times in expectation. This property is formally stated as Lemma 2.

- In Section 3 we show how to obtain the trade-off $T = \widetilde{O}(\max\{\frac{n^2}{S}, m\})$ for an arbitrary choice of space bound $S$. Our initial approach makes use of a modification of a technique introduced by Broder *et al.* [10]. It consists in running $p \approx S$ walks of length $t \approx \frac{n^2}{S^2}$ each, which originate from an appropriately chosen subset of $p$ nodes of the graph called *landmarks*. In our formulation, the walks used are Metropolis-Hastings walks (rather than random walks on $G$), and the set of landmarks is sampled uniformly on $V$. By observing the visits of each of these walks to other landmarks from the set, it is possible to obtain information about paths connecting different landmarks. When the performed Metropolis-Hastings walks have a low node-return rate (i.e., when $t > \Delta^2$), the obtained information turns out to be w.h.p. sufficient to find an answer to $\mathscr{S}$-$\mathscr{T}$ connectivity with a low probability of error. Otherwise, when $t < \Delta^2$, we modify the approach, performing a logical transformation of graph $G$. We split each node of degree greater than $\sqrt{t}$, so that the maximum degree of the modified graph does not exceed $\sqrt{t}$. Then, all of the considerations are performed for this modified graph. In particular, the set of landmark nodes is chosen by uniform sampling on the set of nodes of this modified graph. The overhead associated with this transformation is just small enough for our algorithm to have the claimed time complexity of $T = \widetilde{O}(\max\{\frac{n^2}{S}, m\})$. An implementation of the complete algorithm is provided in Appendix A.

Finally, in the closing Section 4 we discuss the tightness of the obtained results. We also propose a modified weighting of the Metropolis-Hastings walk which performs faster than uniform-weighted Metropolis-Hastings for many classes of graphs, while still covering all the nodes of the graph in $\widetilde{O}(n^2)$ time. This walk satisfies the property that its commute time between any pair of nodes (and consequently also the average hitting time) is asymptotically upper-bounded by the values of the respective parameters for the unbiased random walk. In particular, it covers all the nodes of the previously mentioned glitter star, in expected $\widetilde{O}(n)$ steps.

### 1.3 Notation and model

The input graph $G = (V, E)$, with $|V| = n$ and $|E| = m$, is simple and not necessarily connected. In order to simplify notation for complexity bounds, we assume $m = \Omega(n)$. The degree of a node $v \in V$ is denoted by $\deg(v)$, the neighborhood of node $v$ by $\Gamma(v)$, and the closed neighborhood of $v$ by $\Gamma^+(v) = \Gamma(v) \cup \{v\}$. The maximum degree of the graph is denoted by $\Delta$. The arc set $\vec{E} \subseteq V \times V$ of undirected graph $G$ is understood as the set of arcs of all edges and self-loops of $G$: $\vec{E} = \{(v, u) : v \in V, u \in \Gamma^+(v)\}$. An arc $(v, u) \in \vec{E}$ is sometimes denoted as $e_{vu}$ for compactness of notation. Note that the symbols $V$, $E$, $\Delta$, $n$, $m$ always refer to the input graph $G$. When considering a different graph $X$, we will sometimes denote its vertex, edge, and arc sets by $V(X)$, $E(X)$, and $\vec{E}(X)$, respectively.

Our algorithms are designed for the classical RAM model of computation. No special assumptions are made on the representation of graph $G$, except that for any node $v \in V$, there should exist a local ordering on the set of its neighbors, given by the bijective function $PORT_v : \Gamma(v) \to \{0, 1, \ldots, \deg(v) - 1\}$. Each of the following operations should be possible to implement in $\widetilde{O}(1)$ time: computing $\deg(v)$, computing $PORT_v(u)$ for a node $u \in \Gamma(v)$, and "traversing an edge" by computing $PORT_v^{-1}(i)$, for port $i \in \{0, 1, \ldots, \deg(v) - 1\}$. An example of a permissible representation is a lexicographically sorted array of ordered pairs of identifiers of neighboring nodes $(u, v)$, taken over $\{u, v\} \in E$.

For most of the paper, we consider weighted reversible Markovian processes corresponding to a random walk $RW(X)$ on some weighted undirected graph $X$ with positive weights on arcs. The walk is located on the nodes of graph $X$, and the next state of the walk is reached by following an arc incident to the current node, chosen with probability proportional to the weight of this arc. By a slight abuse of notation, we denote the transition matrix of the walk in the same way as the weighted graph. Most other notation follows that of Aldous and Fill [1]. In particular, we consider the following random variables:

- $N_a(t)$ denotes the number of steps in the time interval $[0, t)$ during which the walk visits $a$, where the symbol $a$ may represent a node, edge, or arc of the graph.
- $T_a$ denotes the first moment of time $t > 0$ at which the walk first visits (or returns to) a node from $a$, where the symbol $a$ may represent a subset of nodes or a single node of the graph.

By writing $\mathbb{E}_\alpha Y$ and $\Pr_\alpha[E]$, respectively, we mean the expectation of random variable $Y$, and the probability of event $E$ occurring, taken over walks starting from probability distribution $\alpha$ (which may be concentrated on a single node or arc). A walk starting from an arc is understood as one which starts from the head of the arc at time 0, and then moves to the tail of the arc at time 1.

Given a weighted graph $X$, we denote by $Com(i, j) \equiv \mathbb{E}_i T_j + \mathbb{E}_j T_i$ the commute time between nodes $i, j \in V(X)$. Throughout the paper, we consider only walks representing reversible Markovian processes, corresponding to symmetric weightings of the graph: $w(e_{vu}) = w(e_{uv})$, for all $(u, v) \in \vec{E}$. In some of the proofs, we rely on the resistor network representation of reversible walks: for each edge $e = \{u, v\} \in E(X)$ having weight $w(e)$ on each of its arc, a resistor with resistance $1/w(e)$ is placed between nodes $u$ and $v$ of the resistor network. The symbol $R(u, v)$ denotes the resistance of replacement between nodes $u$ and $v$ of the network. We recall that $Com(i, j) = R(i, j) \sum_{e \in \vec{E}(X)} w(e)$. [11]

## 2 Preliminaries: The Metropolis-Hastings Walk on Graphs

The *Metropolis-Hastings walk* with potential function $f : V \to \mathbb{R}^+$ is defined as a walk on the weighted graph $G_f = (V, E, w_f)$ with the following assignment of weights $w_f : \vec{E} \to \mathbb{R}^+$:

$$w_f(e_{vu}) = \min\left\{\frac{f(v)}{\deg(v)}, \frac{f(u)}{\deg(u)}\right\}, \quad \text{for all } \{v, u\} \in E.$$

$$w_f(e_{vv}) = f(v) - \sum_{u \in \Gamma(v)} w_f(\{v, u\}), \quad \text{for all } v \in V,$$

We recall that for a walk in state $v \in V$, the next state is chosen as $u \in \Gamma(v) \cup \{v\}$ with probability proportional to the weight $w_f(e_{vu})$. By a classical result due to Metropolis *et al.* [18], for a given representation of graph $G$, a single step of the Metropolis-Hastings walk $RW(G_f)$ can be simulated in $\widetilde{O}(1)$ time and space by means of the procedure shown in Algorithm 1. The algorithm takes advantage of the fact that $w_f(e_{vu})/\sum_{x \in \Gamma^+(v)} w_f(e_{vx}) \leq \frac{1}{\deg(v)}$, for all $u \in \Gamma(v)$. For a walk located at node $v$, it samples a node $u \in \Gamma(v)$ with uniform probability $\frac{1}{\deg(v)}$, and accepts $u$ as the new state with the appropriate probability. We remark that a step of $RW(G_f)$ can also be simulated by a log-space automaton which pushes a pebble along the arc $(v, u)$. The pebble remains at $u$ if state $u$ is accepted, and otherwise reverts to $v$ by traversing the arc $(u, v)$. Thus, one step of $RW(G_f)$ can be simulated by at most two moves of a pebble.

```
function next_state (v: node) {
    u ← neighbor of v in G chosen uniformly at random;    // pick a new state
    with probability min{deg(v)/deg(u) · f(u)/f(v), 1} do return u;  // accept: move to new state
    return v;    // do not accept: keep current state
}
```

**Algorithm 1:** State transition function on $V$ for the walk $RW(G_f)$.

**Definition 1.** *We denote by $G_1$ the weighted graph $G_f$ for the unit potential function $f(v) \equiv 1$.*

From the next two sections, we focus on the Metropolis-Hastings walk $RW(G_1)$. We note that the weights on the edges of $G$ are now simply given by $w(e_{vu}) = \min\{\frac{1}{\deg(v)}, \frac{1}{\deg(u)}\}$.

The bound on the time required by the Metropolis-Hastings walk to discover w.h.p. the entire connected component containing the starting node of the walk follows from the considerations of Nonaka *et al.* [20]. (All omitted proofs are provided in the Appendix.)

**Lemma 1** ([20])**.** *Let $i \in V$, let $H$ be the connected component of $G$ containing node $i$, and let $n_H = |V(H)|$. Then:*

- *a walk $RW(G_1)$ of length $12n_H^2$ starting from $i$ covers an arbitrary node $j \in V(H)$ with probability at least $\frac{1}{2}$.*
- *a walk $RW(G_1)$ of length $24n_H^2 \log n$ starting from $i$ covers all nodes from $V(H)$ with probability at least $1 - \frac{1}{n}$.*

By the above Lemma, a solution to USTCON, with probability $1 - \frac{1}{n}$, is obtained by running the walk $RW(G_1)$, starting from $\mathscr{S}$, for $24n^2 \log n$ steps. USTCON can therefore be solved in log-space by running Algorithm 1 in a loop for $O(n^2 \log n)$ iterations. (We are unaware of any previous reference in the literature for this observation.)

**Corollary 1.** *There is a log-space algorithm for USTCON which runs in time $O(n^2 \log n)$, with probability of one-sided error bounded by $\frac{1}{n}$.* □

For our purposes, we will need a more detailed analysis of the behavior of the Metropolis-Hastings walk. We start by recalling that the Metropolis-Hastings walk $RW(G_1)$ is a reversible Markovian process, since $w(e_{vu}) = w(e_{uv})$ for all arcs. Its stationary distribution is the uniform distribution $\pi : V \to \mathbb{R}^+$, with $\pi(v) = \frac{1}{n}$, for all $v \in V$. This allows us to show the following key lemma which captures the "low node-return rate" property of the Metropolis-Hastings walk, as highlighted in the Introduction. The first claim states that a Metropolis-Hastings walk starting within any subset of nodes $A \subsetneq V$ is likely to leave it within $O(|A|^2)$ steps, while its second claim shows that a Metropolis-Hastings walk of length $t$ is likely to return to its starting node not more than $O(\sqrt{t})$ times. However, both of the above statements hold only when considering walks of duration $\Omega(\Delta^2)$.

**Lemma 2.** *Suppose that $G$ is connected. Let $A \subsetneq V$, and let $i \in A$. For a weighted random walk $RW(G_1)$ starting from node $i$:*

*(i) the expected time to reach a node from $V \setminus A$ is bounded by:*

$$\mathbb{E}_i T_{V \setminus A} < (|A| + 1)(6|A| + 2\Delta),$$

*(ii) the expected number of visits to node $i$ before any time $t$, $0 < t < 6n^2$, is bounded by:*

$$\mathbb{E}_i N_i(t) < 5\sqrt{t} + 2\Delta.$$

The proof of the lemma follows by an analysis of resistances of replacement along shortest paths in the resistor network for the weighted graph $G_1$.

## 3 A time-space trade-off for USTCON

The time-space tradoffs for USTCON proposed by Broder *et al.* [10] make use of a number of short random walks, originating from a subset of nodes of the graph called *landmarks*. Herein, we design an algorithm which replaces these random walks by Metropolis-Hastings walks.

We start by a brief overview of the landmark-based approach. When considering an algorithm using space $S$, the size of the set of landmarks is defined by a parameter $p = \Theta(S/\log n)$. The algorithm first chooses a set of landmarks $L \subseteq V$ consisting of $p + 2$ nodes: node $\mathscr{S}$, node $\mathscr{T}$, and $p$ nodes picked (in the case of our work) uniformly at random from $V$. Then, a walk of suitably chosen length $t$ is released from each of the landmarks. Throughout this process, the algorithm maintains a disjoint-set data structure (also known as "Union-Find" [16]) on the set of landmarks, with each set corresponding to the landmarks identified as belonging to the same connected component of the graph.

Initially, each landmark belongs to a separate set. Whenever a walk starting from one landmark hits some other landmark, the algorithm updates the data structure, merging the classes corresponding to these two landmarks. At the end of the process, if landmarks $\mathscr{S}$ and $\mathscr{T}$ belong to the same class, then, with certainty, there exists an $\mathscr{S}$-$\mathscr{T}$ path in $G$, and the answer to USTCON is positive. Otherwise, the algorithm returns a negative result, and, in the rest of this Section, we focus on proving that this result is correct w.h.p.

The runtime of the algorithm of Broder *et al.* is determined by the time of running $p = \widetilde{\Theta}(S)$ random walks of length $t$ each, thus $T = \widetilde{O}(tp) = \widetilde{O}(\frac{tp^2}{S})$. To achieve the claimed trade-off of $T = \widetilde{O}(\frac{n^2}{S})$, we will therefore need to use walks of length roughly $t \approx \frac{n^2}{p^2}$.

### 3.1 An initial approach

We fix a value of the parameter $p = O(S)$, describing the number of landmark nodes. The landmark-based algorithms are built around the premise that landmarks belonging to the same connected component of $G$ quickly discover each other with the help of the short walks they release. In particular, it is desirable that the set of landmarks in each connected component of $G$ has the property that for any partition of the set of landmarks into two subsets, a short walk originating from a landmark in one of these subsets is likely to reach some landmark from the other subset. Broder *et al.* [10] observe (cf. also [14] for a high-level exposition of the argument) that this property is satisfied if the considered set of landmarks is *good*, i.e., it fulfills the following two assumptions. Firstly, the set of short walks originating from all of the landmarks should be likely to jointly cover all the arcs of the graph. Secondly, a short walk originating from an arbitrary starting node of the graph should be likely to reach at least one landmark from the set.

Most of the analysis and key lemmas in this subsection follow along the lines proposed by Broder *et al.* We confine ourselves to a summary of the approach, highlighting the subtle differences resulting from the use of Metropolis-Hastings walks. We start by re-setting the good landmark property in the context of Metropolis-Hastings walks $RW(G_1)$ of a specifically chosen length $\tau$.

**Property 1.** *Let $L \subseteq V$ be the set of $p = |L|$ landmark nodes, let $H$ be a connected component of $G$, and let $n_p = \max\{\gamma \frac{n}{p} \log n, \Delta\}$, where $\gamma = 60$ is a suitably chosen absolute constant (whose value follows from the proof of Lemma 5). We say that the set of landmarks $L$ is* good with respect to $H$ *if the following properties hold:*

- *With probability at least $1 - \frac{1}{n}$, a set of $p$ walks $RW(G_1)$ of length $\tau = n_p^2$ each, with one walk originating from each landmark from $L$, covers an arbitrarily chosen arc of $H$.*
- *With probability at least $1 - \frac{1}{n}$, a walk $RW(G_1)$ of length $\tau = n_p^2$, originating from an arbitrarily chosen node of $H$, hits some landmark from $L$.*

In the above property, the choice of the length $\tau$ of the walk takes into account that walks of length $\widetilde{O}(\frac{n^2}{p^2})$ lead us to the sought time complexity of $\widetilde{O}(\frac{n^2}{p})$ for the algorithm. However, in order to ensure that a uniformly sampled landmark set is likely to be good, we will make use of the low node-return rate of the Metropolis-Hastings walk from Lemma 2, and thus we need to have $\tau = \Omega(\Delta^2)$.

We will now show that that Property 1 holds w.h.p. for a set of landmarks, each of which is chosen according to the uniform distribution $\pi$ on the set of nodes $V$. To achieve this, we capture the "contribution" of a single Metropolis-Hastings walk to the probability of success of the events described in the Property. It turns out that a Metropolis-Hastings walk of the chosen length $\tau$, when starting from a landmark, has probability $\Omega(1/p)$ of reaching an arbitrary arc of the graph. When starting from an arbitrary node from $V$, such a walk has probability $\Omega(1/p)$ of reaching any specific landmark. These claims are formulated in a slightly more general way as the two lemmas below. Their proofs take into account the low node-return rate property from Lemma 2($ii$), and the properties of a walk starting from its stationary distribution $\pi$.

**Lemma 3.** *Suppose that $G$ is connected. For a weighted walk $RW(G_1)$ starting from a node chosen according to the uniform distribution $\pi$, the probability of traversing (a fixed) non-loop arc $e_{ij}$ before time $t$, where $\Delta^2 \leq t < 6n^2$, is:*

$$\Pr_\pi[T_{e_{ij}} < t] > 0.1\sqrt{t}/n.$$

**Lemma 4.** *Suppose that $G$ is connected. Let $j \in V$ be picked according to the uniform distribution $\pi$. For a weighted walk $RW(G_1)$ starting from some node $i \in V$, the probability of reaching $j$ before time $t$, where $\Delta^2 \leq t < 6n^2$, is:*

$$\Pr_i[T_j < t] > 0.1\sqrt{t}/n.$$

After combining the above lemmas and applying some elementary arguments about unions of independent events, we finally obtain that Property 1 is satisfied w.h.p. by landmarks uniformly chosen from $V$, provided that the considered connected component is sufficiently large.

**Lemma 5.** *If a connected component $H \subseteq G$ has $n_H \geq n_p/6$ nodes, then a (multi)set of $p$ nodes, picked with uniform probability from $V$, is a good set of landmarks with respect to $H$ with probability at least $1 - \frac{1}{2n}$.*

The results of Broder *et al.* imply directly that if a set of landmarks is good with respect to a connected component $H$, then all landmarks in $H$ can be identified as belonging to the same connected component by releasing a small number of walks from each landmark, and applying Union-Find type operations on a disjoint-set datastructure on the landmarks. Since the proof does not rely on any other assumptions beyond the properties of good landmarks, the result is directly applicable to our considerations of the Metropolis-Hastings walk.

**Lemma 6** ([10])**.** *Let $L$ be a set of good landmarks with respect to connected component $H \subseteq G$. Then, a set of walks of length $n_p^2$ each, with $\beta \log n$ walks originating from each of the landmarks, with probability at least $1 - \frac{1}{2n}$ discovers that all landmarks located within $H$ belong to the same connected component.*

In the above, the absolute constant $\beta$ can be chosen as $\beta = 72$.

Our algorithm for USTCON is now obtained as follows. We pick a set of landmarks $L$, consisting of $\mathscr{S}$, $\mathscr{T}$, and $p$ nodes picked uniformly at random from $V$, and then follow $\beta \log n$ Metropolis-Hastings walks from each landmark, updating the disjoint-set data structure. Finally, the algorithm decides whether $\mathscr{S}$ and $\mathscr{T}$ are connected based on whether these two landmarks have been identified as belonging to the same connected component.

The algorithm never provides a false-positive answer. The probability of identifying a pair of nodes $\mathscr{S}$ and $\mathscr{T}$ from the same component $H \subseteq G$ as not being connected, can be bounded using the following argument adapted from Broder *et al.* Let $H$ be the connected component of $G$ containing node $\mathscr{S}$. If $n_H \geq n_p/6$, then by Lemma 5, the set $L$ is a set of good landmarks with respect to $H$ with probability at least $1 - \frac{1}{2n}$ (note that adding nodes $\mathscr{S}$ and $\mathscr{T}$ to a good set of landmarks cannot make this set of landmarks a bad one). Conditioned on this, by Lemma 6, we obtain a correct answer to USTCON with probability $1 - \frac{1}{2n}$. Thus, the algorithm works correctly with probability at least $1 - \frac{1}{n}$. In the case when $n_H < n_p/6$, we consider only the walks originating from landmark $\mathscr{S}$. There are $\beta \log n$ such (independent) walks, each of length $n_p^2 > 36n_H^2$. It follows from Lemma 3, putting $i = \mathscr{S}$ and $j = \mathscr{T}$, that in this case, node $\mathscr{T}$ will be reached with probability at least $1 - \frac{1}{n}$. This completes the proof of correctness.

**Proposition 1.** *For all $p \geq 1$, there is an algorithm solving USTCON using space $S = \widetilde{O}(p)$ and time $T = \widetilde{O}(n_p^2 p)$, where $n_p = \max\{\gamma \frac{n}{p} \log n, \Delta\}$, with probability of one-sided error bounded by $\frac{1}{n}$.* □

For the case when $p = \widetilde{O}(\frac{n}{\Delta})$, we have obtained the trade-off $T = \widetilde{O}(\frac{n^2}{S})$. We now show how to obtain the claimed trade-off in the general case.

### 3.2 Removing the dependence on maximum degree

We now remove the dependence of length of the used walks on the value of $\Delta$. We design a graph $G^* = (V^*, E^*)$ by subdividing the nodes of $G$, so that each node from $V$ turns into a path of nodes in $G^*$ with maximum degree bounded by $D + 2$, where $D \geq 1$ is an integer parameter, whose value is specified later. Formally, graph $G^*$ is defined as follows:

- For each node $v \in V$, $V^*$ contains $\lceil \frac{\deg(v)}{D} \rceil$ copies of $v$, labeled $(v, 0), (v, 1), \ldots, (v, \lceil \frac{\deg(v)}{D} \rceil - 1)$.
- Nodes $(u, i)$ and $(v, j)$, $u \neq v$, are connected by an edge in $E^*$ if and only if $\{u, v\} \in E$, $iD \leq PORT_v(u) < (i + 1)D$, and $jD \leq PORT_u(v) < (j + 1)D$.
- Nodes $(u, i)$ and $(u, i + 1)$, for all $0 \leq i < \lceil \frac{\deg(v)}{D} \rceil - 1$, are connected by an edge of $E^*$, with special port labels '$prev$' and '$next$' at its endpoints.

Let $n^*$ and $\Delta^*$ be the number of nodes and the maximum degree of $G^*$, respectively. We have $\Delta^* = D + 2$, and the following bound on $n^*$ holds:

$$n^* = \sum_{v \in V} \lceil \frac{\deg(v)}{D} \rceil < n + \sum_{v \in V} \frac{\deg(v)}{D} = n + \frac{2m}{D}.$$

Solving USTCON on $G$ between nodes $\mathscr{S}$ and $\mathscr{T}$ can be reduced to solving USTCON on $G^*$ between nodes $(\mathscr{S}, 0)$ and $(\mathscr{T}, 0)$, since the transformation of $G$ into $G^*$ does not affect connectivity. In order to apply the algorithm for USTCON to $G^*$, rather than to $G$, we introduce the following modifications:

- Landmarks need to be distributed following the uniform distribution on $V^*$. This can be achieved by picking $p$ integers uniformly at random from the range $[1, n^*]$, then enumerating all the nodes of $V^*$ in order, and associating the landmarks with the corresponding nodes from $V^*$. This operation can be performed in $O(n^* + p \log n^*)$ time, which is always $\widetilde{O}(m)$.
- The performed walks need to follow $RW(G_1^*)$, rather than $RW(G_1)$. A simulation of one step of the walk $RW(G_1^*)$ can be performed in $\widetilde{O}(1)$ time.
- The duration of each of the performed walks is given as $n_p^{*2}$, where:

$$n_p^* = \max\left\{\gamma \frac{n^*}{p} \log n^*, \Delta^*\right\} < \max\left\{\gamma \frac{n + 2m/D}{p} \cdot 2 \log n, D + 2\right\} \tag{1}$$

It follows that the time complexity of the entire algorithm is bounded by the $\widetilde{O}(m)$ complexity of landmark distribution and the $\widetilde{O}(n_p^{*2} p)$ complexity of simulating the Metropolis-Hastings walks on $G^*$. Substituting the expression from (1) for $n_p^*$, we have:

$$T = \widetilde{O}(m + n_p^{*2} p) = \widetilde{O}\left(m + \frac{n^2}{p} + \frac{m^2}{D^2 p} + D^2 p\right)$$

Now, putting $D = \lceil \sqrt{m/p} \rceil$ gives $D^2 p = \Theta(m)$, and we obtain the required time bound $T = \widetilde{O}(m + \frac{n^2}{p} + \frac{m^2}{m} + m) = \widetilde{O}(\max\{\frac{n^2}{p}, m\})$. Since the proposed solution can be implemented with a space bound of $S = \widetilde{O}(p)$, we have proven the main theorem of the paper.

**Theorem 1.** *For all $S \geq c \log n$, where $c > 0$ is some model-dependent constant, there is an algorithm solving USTCON using space $S$ and time $T = \widetilde{O}(\max\{\frac{n^2}{S}, m\})$, with probability of one-sided error bounded by $\frac{1}{n}$.* □

# 4 Remarks

**Tightness of the trade-off.** For a space bound $S \geq \frac{n^2}{m}$, we cannot hope for an algorithm with smaller run-time than $T = \widetilde{O}(m)$, achieved in Theorem 1. In fact, the lower bound of $T = \Omega(m)$ holds for the RAM model under most reasonable representations of $G$ in the memory (cf. Proposition 3 in Appendix B for a standard proof).

For smaller values of $S$, the optimality of the achieved trade-off $S \cdot T = \widetilde{O}(n^2)$ is open. For the extremal case of $S = O(\log n)$, the results of [6] imply that $T = \widetilde{\Omega}(n^2)$ for any deterministic algorithm using a jumping automaton (JAG) with at most one movable pebble. There is also little hope of improving the time complexity using randomized algorithms similar to the Metropolis-Hastings walk, since Nonaka *et al.* [20] showed that any walk, having a stationary distribution which is (almost) uniform on the nodes of the graph, has $\Omega(n^2)$ cover time for some graphs.

Even more strongly, one can ask whether there exists an algorithm for USTCON which runs in $\widetilde{O}(1)$ space and $\widetilde{O}(m)$ time. This appears unlikely in view of the negative result of Edmonds [13], who showed that a randomized JAG using $\widetilde{O}(1)$ space and $o(\log n/\log\log n)$ pebbles requires in expectation $n^{1+\Omega(1)/\log\log n}$ time to explore certain 3-regular graphs.

**Fine-tuning the Metropolis-Hastings walk.** In view of Lemma 1, the Metropolis-Hastings walk visits all the nodes of a graph within $\widetilde{O}(n^2)$ steps. This is an improvement with respect to the bound of $O(nm)$ on the cover time of an unbiased random walk. Nevertheless, the Metropolis-Hastings walk may perform worse than the random walk for specific graph classes. A generic example of such a graph, called the *glitter star*, was defined by [20] as a tree on $n = 2l+1$ nodes, with one central node of degree $l$ connected to $l$ nodes of degree 2, which are in turn connected to $l$ leaves. On the glitter star, the cover time of the random walk is $\Theta(n \log n)$, and the cover time of the Metropolis-Hastings walk is $\Theta(n^2)$.

Below we propose a walk $RW(G_f)$ with a different potential function which combines some of the advantages of the random walk and the Metropolis-Hastings walk.

**Proposition 2.** *For a graph $G$, let the node potential function $f : V \to \mathbb{R}^+$ be given as $f(u) = \frac{\deg(u)}{d} + 1$, where $d = \frac{2m}{n}$ is the average degree of the graph. Then, for any pair of nodes $u, v \in V$, the walk $RW(G_f)$ achieves a commute time of:*

$$Com_{G_f}(u,v) = O(\min\{Com_G(u,v), Com_{G_1}(u,v)\}),$$

*where $Com_G(u,v)$ and $Com_{G_1}(u,v)$ denote the commute times for the random walk on $G$ and the Metropolis-Hastings walk, respectively. A step of the walk $RW(G_f)$ can be simulated using $\widetilde{O}(1)$ space and time.*

The above Proposition implies that for any graph, the walk $RW(G_f)$ with $f(u) = \frac{\deg(u)}{d} + 1$, is asymptotically not slower than the unbiased random walk in terms of parameters such as maximum hitting time and (arbitrarily weighted) average hitting time. At the same time, this walk preserves the upper bound of $\widetilde{O}(n^2)$ on the cover time in the graph, making it an interesting alternative to the unbiased random walk in practical applications, e.g., for different random graph models.

We remark that there exist different ways of combining the unbiased random walk and the Metropolis-Hastings walk. For example, one may consider an automaton which iteratively performs a phase of the walk $RW(G)$, followed by a phase of the walk $RW(G_1)$ of the same length, doubling the lengths of both walks in each subsequent iteration. Such a walk visits all the nodes of the graph in expected time asymptotically equal to the cover time of the faster of the two walks.

## References


[1] D. Aldous, J. Fill. *Reversible Markov Chains and Random Walks on Graphs*. Book draft available at `http://stat-www.berkeley.edu/users/aldous/RWG/book.html`, 2001.

[2] R. Aleliunas, R.M. Karp, R.J. Lipton, L. Lovasz, C. Rackoff. Random walks, universal traversal sequences, and the complexity of maze problems. *Proc. 20th Annual Symposium on Foundations of Computer Science (FOCS)*, 1979, pp. 218–223.

[3] C. Alvarez, R. Greenlaw. A compendium of problems complete for symmetric logarithmic space. *Computational Complexity* 9:2 (2000), 123–145.

[4] R. Armoni, A. Ta-Shma, A. Wigderson, S. Zhou. An $O(\log(n)^{4/3})$ space algorithm for $(s,t)$ connectivity in undirected graphs. *Journal of the ACM* 47:2 (2000), 294-311.

[5] G. Barnes, U. Feige. Short random walks on graphs. *Proc. 25th Annual ACM Symposium of the Theory of Computing (STOC)*, 1993, pp. 728–737.

[6] P. Beame, A. Borodin, P. Raghavan, W.L. Ruzzo, M. Tompa. Time-Space Tradeoffs for Undirected Graph Traversal by Graph Automata. *Information and Computation* 130:2 (1996), 101–129.

[7] P. Berenbrink, C. Cooper, R. Elsässer, T. Radzik, T. Sauerwald. Speeding Up Random Walks with Neighborhood Exploration. *Proc. 21st Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 2010, pp. 1422–1435.

[8] P. Berman, J. Simon. Lower bounds on graph threading by probabilistic machines. *Proc. 24th Annual Symposium on Foundations of Computer Science (FOCS)*, 1983, pp. 304–311.

[9] A. Borodin, S.A. Cook, P.W. Dymond, W.L. Ruzzo, M. Tompa. Two applications of inductive counting for complementation problems. *SIAM Journal on Computing* 18:3 (1989), 559-578.

[10] A.Z. Broder, A.R. Karlin, P. Raghavan, E. Upfal. Trading space for time in undirected $s$-$t$ connectivity. *SIAM Journal on Computing* 23 (1994), 324–334.

[11] A. Chandra, P. Raghavan, W.L. Ruzzo, R. Smolensky, P. Tiwari. The electrical resistance of a graph captures its commute and cover times. *Proc. 21st Annual ACM Symposium of the Theory of Computing (STOC)*, 1989, pp. 574–586.

[12] S.A. Cook, C.W. Rackoff Space lower bounds for maze threadability on restricted machines. *SIAM Journal on Computing* 9:3 (1980), 630–652.

[13] J. Edmonds. Time-space trade-offs for undirected st-connectivity on a JAG. *Proc. 25th Annual ACM Symposium of the Theory of Computing (STOC)*, 1993, pp. 718–727.

[14] U. Feige. A Randomized Time-Space Trade-off of $\widetilde{O}(m\widetilde{R})$ for USTCON. *Proc. 34th Annual Symposium on Foundations of Computer Science (FOCS)*, 1993, pp. 238–246.

[15] P. Gopalan, R.J. Lipton, A. Mehta. Randomized Time-Space Tradeoffs for Directed Graph Connectivity. *Proc. Foundations of Software Technology and Theoretical Computer Science (FSTTCS)*, LNCS 2914, 2003, pp. 208–216.

[16] J.E. Hopcroft, J.D. Ullman. Set Merging Algorithms. *SIAM Journal on Computing* 2:4 (1973), 294-303.

[17] S. Ikeda, I. Kubo, M. Yamashita. The hitting and cover times of random walks on finite graphs using local degree information. *Theoretical Computer Science* 410:1 (2009), 94–100.

[18] N. Metropolis, A.W. Rosenbluth, M.N. Rosenbluth, A.H. Teller, and E. Teller, Equation of State Calculations by Fast Computing Machines, *Journal of Chemical Physics* 21 (1953), 1087–1092.

[19] N. Nisan. *$RL \subseteq SC$. Proc. 24th Annual ACM Symposium of the Theory of Computing (STOC)*, 1992, pp. 619–623.

[20] Y. Nonaka, H. Ono, K. Sadakane, M. Yamashita. The hitting and cover times of Metropolis walks. *Theoretical Computer Science* 411:16–18 (2010), 1889–1894.

[21] O. Reingold. Undirected connectivity in log-space. *Journal of the ACM* 55:4 (2008), 1–24.

[22] W.J. Savitch. Relationships between nondeterministic and deterministic tape complexities. *Journal of Computer and System Sciences* 4 (1970), 177–192.

[23] M. Saks, S. Zhou. $BP_HSPACE(S) \subseteq DSPACE(S^{3/2})$. *Journal of Computer and System Sciences* 58:2 (1999), 376–403.

## Appendix A: Implementation

For the sake of completeness, below we provide the pseudocode of the algorithm for USTCON announced in Theorem 1. The implementation is self-contained, except for the following subroutines. The disjoint-set data structure is implemented by the procedures: $SET(x)$ which adds a new set containing only element $x$ to the data structure, $FIND(x)$ which returns (the identifier of) the set containing element $x$, and $UNION(S_1, S_2)$ which replaces sets $S_1$ and $S_2$ by set $S_1 \cup S_2$ in the data structure. Each of these operations is performed in amortized $\widetilde{O}(1)$ time.

The routine $TRAVERSE_EDGE_v(port)$, for a node $v \in V$, returns a pair $(u, inport)$, such that $u \in \Gamma(v)$ with $PORT_v(u) = port$, and $inport = PORT_u(v)$. This routine can be performed in $\widetilde{O}(1)$ time in the RAM model, as well as in most JAG-based models.

We recall the values of the absolute constants: $\gamma = 60$ and $\beta = 72$.

```
// Solution to USTCON using p auxiliary landmarks
procedure test_connectivity (𝒮, 𝒯: nodes from V) {
    D ← ⌈√(m/p)⌉;
    n* ← Σ_{v∈V} ⌈deg(v)/D⌉;
    L ← {(𝒮, 0), (𝒯, 0)};

    // Distribute p landmarks uniformly on V*
    ℓ ← multi-set of p integers chosen uniformly at random from the range {1, 2, ..., n*};
    i ← 0;
    for v ∈ V do
        for j ← 0, 1, ..., ⌈deg(v)/D⌉ − 1 do {
            i ← i + 1;
            if i ∈ ℓ then L ← L ∪ {(v, j)};
        }
    for l ∈ L do SET(l) ← l;

    // From each landmark, run β log n* Metropolis walks RW(G*_1) of length n*_p^2 each
    repeat β log n* times {
        for l ∈ L do {
            s ← l;
            repeat ⌈max{γ (n*/p) log n*, D + 2}⌉^2 times {
                s ← next_state* (s);
                UNION(FIND(s), FIND(l));
            }
        }
    }
    if FIND((𝒮, 0)) = FIND((𝒯, 0)) then return "connected";
    return "probably not connected";
}

// Simulate one step of the walk RW(G*_1) from state (v, i) ∈ V*
function next_state* (v: node, v_i: integer) {
    v_deg* ←get_degree*(v, v_i);
    port ← get_random_port*(v, v_i);
    if port = 'prev' then {
        u ← v;
        u_i ← v_i − 1;
    } else if port = 'next' then {
```

```
        u ← v;
        u_i ← v_i + 1;
    } else { // port ∈ [left, right] is an integer corresponding to a port at v in G
        (u, inport) ← TRAVERSE_EDGE_v(port);
        u_i ← ⌊inport/D⌋;
    }
    u_deg* ← get_degree*(u, u_i);
    with probability min{v_deg*/u_deg*, 1} do return (u, u_i);
    return (v, v_i);
}

// Return the degree of (v, i) in G*
function get_degree* (v: node, i: integer) {
    left ← i · D;
    right ← min{(i + 1) · D − 1, deg(v)};
    deg* ← right − left + 1;
    if left > 0 then deg* ← deg* + 1;
    if right < deg(v) then deg* ← deg* + 1;
    return deg*;
}

// Return a port at node (v, i) in G* chosen uniformly at random
function get_random_port* (v: node, i: integer) {
    left ← i · D;
    right ← min{(i + 1) · D − 1, deg(v)};
    deg* ← get_degree* (v, i);
    with probability (right − left + 1)/deg* do
        return integer from range [left, right] chosen uniformly at random;
    neighbors ← ∅;
    if left > 0 then neighbors ← neighbors ∪ {'prev'};
    if right < deg(v) then neighbors ← neighbors ∪ {'next'};
    return element of neighbors chosen uniformly at random;
}
```

## Appendix B: Auxiliary claims

**Lemma 7.** *For all $i, j \in V$, $\mathbb{E}_i N_j(t) = \mathbb{E}_j N_i(t)$.*

*Proof.* Let $X_u(\tau)$, $u \in \{i, j\}$, denote the random variable equal to 1 if a walk of length $\tau$ is located at $u$ after $\tau$ steps, and 0 otherwise. Since $RW(G_1)$ is a reversible Markovian process, by the properties of the $\tau$-th power of the transition matrix of the walk (cf. [1], Chapter 3.1), we have:

$$\pi(i)\mathbb{E}_i X_j(\tau) = \pi(j)\mathbb{E}_j X_i(\tau).$$

Since $\pi(i) = \pi(j) = \frac{1}{n}$, it follows that $\mathbb{E}_i X_j(\tau) = \mathbb{E}_j X_i(\tau)$. Taking into account that $N_u(t) = \sum_{\tau=0}^{t-1} X_u(\tau)$, $u \in \{i, j\}$, by linearity of expectation we obtain the claim. □

**Lemma 8.** *For any node $i \in V$:*

$$\mathbb{E}_\pi N_i(t) = \frac{t}{n} \tag{2}$$

*and for any arc $e_{ij}$ of $G$ corresponding to an edge $\{i, j\} \in E$:*

$$\mathbb{E}_\pi N_{e_{ij}}(t) = \frac{t}{n} \min\left\{\frac{1}{\deg(i)}, \frac{1}{\deg(j)}\right\}. \tag{3}$$

*Proof.* Follows directly from the stationary distribution of the Metropolis-Hastings walk on nodes and edges. □

**Proposition 3.** *Any algorithm for USTCON requires time $\Omega(m)$.*

*Proof.* Consider a generic instance of USTCON defined as follows. Take two disjoint copies of some 2-edge-connected graph $H$ on $n/2$ nodes, with one distinguished node $x$. The two copies of $H$ are assigned the subscripts 1 and 2, respectively. Now, as the considered instance of USTCON we use, with probability 1/2, the disconnected graph $H_1 \cup H_2$ with $\mathscr{S} = x_1$ and $\mathscr{T} = x_2$. Otherwise, we pick an edge $\{u, v\}$ of $H$ uniformly at random, and use as the instance the connected graph $H_1 \cup H_2 \cup \{\{u_2, v_1\}, \{u_1, v_2\}\} \setminus \{\{u_1, v_1\}, \{u_2, v_2\}\}$, likewise with $\mathscr{S} = x_1$ and $\mathscr{T} = x_2$. Subject to a choice of node identifiers in the representations, the connected and disconnected instances differ on precisely 4 memory cells in the adjacency lists of the graph (for nodes $u_1$, $u_2$, $v_1$, and $v_2$), and these cells, taken over the choices of edge $\{u, v\}$, form a partition of the memory representation of the graph. Consequently, the expected number of memory reads for an algorithm deciding connectivity with probability $1/2 + p$ is lower-bounded by $p \cdot m/4$, and is thus $\Omega(m)$, within the range $n \leq m \leq n^2/8 - O(n)$. □

## Appendix B: Proofs of technical lemmas

### Proof of Lemma 2

The interested reader may see this proof as an analogue of the discussion for short random walks in regular graphs, cf. Aldous and Fill, Chapter 6, Proposition 16.

*Claim* $(i)$: Consider a shortest path $P$ in graph $G$ from $i$ to a nearest vertex $j \in V \setminus A$. Let $P = (i_0, i_1, \ldots, i_a, j)$, where $i_0 = i$, and $i_l \in A$, for $0 \leq l \leq a$. Let $G^\circ$ be the subgraph of $G$ induced by nodes from set $A$, their neighbors in $G$, and node $j$: $G^\circ = G[A \cup N(A) \cup \{j\}]$. Since any random walk in $G$ which starts from $i$ and does not enter $V \setminus A$ is confined to nodes and edges of graph $G^\circ$, we have the following relation between the walks $RW(G_1)$ and $RW(G_1^\circ)$:

$$\mathbb{E}_i T_{V\setminus A} \leq \mathbb{E}_i T^\circ_{V\setminus A} = \mathbb{E}_i T^\circ_j < Com^\circ(i, j) = R^\circ(i, j) \sum_{e \in \vec{E}(G^\circ)} w(e), \tag{4}$$

where the latter equality follows from the electrical network representation of random walks. The resistance $R^\circ(i,j)$ is upper-bounded by the resistance of the series connection going through the nodes of path $P$ in $G$:

$$R^\circ(i,j) \leq \tfrac{1}{w(e_{i_0 i_1})} + \tfrac{1}{w(e_{i_1 i_2})} + \ldots + \tfrac{1}{w(e_{i_{a-1} i_a})} + \tfrac{1}{w(e_{i_a j})} =$$

$$= \max\{\deg(i_0), \deg(i_1)\} + \ldots + \max\{\deg(i_{a-1}), \deg(i_a)\} + \max\{\deg(i_a), \deg(i_j)\} < 2\sum_{l=0}^{a-1} \deg(i_l) + 2\Delta.$$

Since the path $P_s = (i_0, i_1, \ldots, i_{a-1})$ is a shortest path in graph $G$ between nodes $i_0$ and $i_{a-1}$, such that $P_s \subseteq A$ and $\Gamma(P_s) \subseteq A$, it follows that (cf. [1]):

$$\sum_{l=0}^{a-1} \deg(i_l) \leq 3|A|,$$

and:

$$R^\circ(i,j) < 6|A| + 2\Delta. \tag{5}$$

Since the total weight of edges and self-loops of $G$ incident to a vertex in $V$ is equal to 1, we have:

$$\sum_{e \in \vec{E}(G^\circ)} w(e) \leq \sum_{v \in A} \left( \sum_{u \in \Gamma(v) \cup \{v\}} w(e_{vu}) \right) + \sum_{u \in \Gamma(j) \cup \{j\}} w(e_{ju}) \leq |A| + 1. \tag{6}$$

Claim $(i)$ follows from inequalities (4), (5), and (6).

*Claim* $(ii)$: Suppose that $s = \sqrt{6t} \geq \frac{t}{n}$, and let:

$$A = \{j \in V : \mathbb{E}_i N_j(t) > s\}.$$

Since the considered walk hits nodes from $V$ a total of (at most) $t$ times, we have $|A| < \frac{t}{s} \leq n$, and the considerations performed in the proof of Lemma 2$(i)$ can be applied for the above-defined set $A$.

First, we bound the expected number of returns to node $i$ for a walk starting at $i$ before reaching $V \setminus A$ for the first time:

$$\mathbb{E}_i N_i(T_{V \setminus A}) = 1 + (1 - \Pr_i[T_{V \setminus A} < T_i]) \cdot \mathbb{E}_i N_i(T_{V \setminus A}) \implies \mathbb{E}_i N_i(T_{V \setminus A}) = \frac{1}{\Pr_i[T_{V \setminus A} < T_i]}.$$

Taking into account [1] (Chapter 3, eq. (28) and Corollary 11) and bound (5), we have:

$$\mathbb{E}_i N_i(T_{V \setminus A}) = \frac{1}{\Pr_i[T_{V \setminus A} < T_i]} = \pi(i) \cdot R(i,j) \cdot \sum_{e \in \vec{E}(G)} w(e) \ \leq \ \pi(i) \cdot R^\circ(i,j) \cdot \sum_{e \in \vec{E}(G)} w(e) <$$

$$< \frac{1}{n} \cdot (6|A| + 2\Delta)n = 6|A| + 2\Delta < 6\frac{t}{s} + 2\Delta. \tag{7}$$

It follows from Lemma 7 that the definition of set $A$ may be rewritten as:

$$A = \{j \in V : \mathbb{E}_j N_i(t) > s\}$$

Thus, $V \setminus A = \{j \in V : \mathbb{E}_j N_i(t) \leq s\}$, which means that if a walk starting from $i$ reaches $V \setminus A$, it will return to $i$ at most $s$ times in expectation before time $t$. So, using (7), we obtain the claim:

$$\mathbb{E}_i N_i(t) \leq \mathbb{E}_i N_i(T_{V \setminus A}) + s < 6\frac{t}{s} + s + 2\Delta = 2\sqrt{6t} + 2\Delta < 5\sqrt{t} + 2\Delta.$$

□

## Proof of Lemma 3

Fix an arbitrary arc $e_{ij}$, with $\{i,j\} \in E$. We will bound the sought probability from the inequality:

$$\mathbb{E}_\pi N_{e_{ij}}(t) \leq \Pr_\pi[T_{e_{ij}} < t]\ \mathbb{E}_{e_{ij}} N_{e_{ij}}(t) \implies \Pr_\pi[T_{e_{ij}} < t] \geq \frac{\mathbb{E}_\pi N_{e_{ij}}(t)}{\mathbb{E}_{e_{ij}} N_{e_{ij}}(t)}. \tag{8}$$

The expected number of traversals of $e_{ij}$ for a walk of even length starting from the stationary distribution on $V$ is given by equation (3).

To bound the expectation from the denominator of (8), we note that by Lemma 2($ii$), $\mathbb{E}_i N_i(t) < 5\sqrt{t} + 2\Delta$, and that arc $e_{ij}$ is chosen with probability $\min\left\{\frac{1}{\deg(i)}, \frac{1}{\deg(j)}\right\}$ during each visit to $i$:

$$\mathbb{E}_i N_{e_{ij}}(t) < (5\sqrt{t} + 2\Delta) \min\left\{\frac{1}{\deg(i)}, \frac{1}{\deg(j)}\right\}.$$

Considering a walk starting from a traversal of arc $e_{ij}$, we observe that after its traversal of $e_{ij}$ the walk must return to node $i$ before traversing $e_{ij}$ again:

$$\mathbb{E}_{e_{ij}} N_{e_{ij}}(t) \leq 1 + \mathbb{E}_j N_{e_{ij}}(t) < 1 + \mathbb{E}_i N_{e_{ij}}(t) < 1 + (5\sqrt{t} + 2\Delta) \min\left\{\frac{1}{\deg(i)}, \frac{1}{\deg(j)}\right\} \leq$$

$$\leq (5\sqrt{t} + 3\Delta) \min\left\{\frac{1}{\deg(i)}, \frac{1}{\deg(j)}\right\}. \tag{9}$$

By combining inequalities (3), (8), (9), and taking into account that $t > \Delta^2$, we obtain the claim:

$$\Pr_\pi[T_{e_{ij}} < t] > \frac{t}{n(5\sqrt{t} + 3\Delta)} > \frac{t}{8n\sqrt{t}} > 0.1\frac{\sqrt{t}}{n}.$$

□

## Proof of Lemma 4

Pick a node $j \in V$ according to the uniform probability distribution $\pi$. We will bound the sought probability from the inequality:

$$\mathbb{E}_i N_j(t) \leq \Pr_i[T_j < t]\ \mathbb{E}_j N_j(t) \implies \Pr_i[T_j < t] \geq \frac{\mathbb{E}_i N_j(t)}{\mathbb{E}_j N_j(t)}. \tag{10}$$

Taking into account Lemma 7 and condition (2), and noting that $j$ is chosen according to the uniform distribution $\pi$ on $V$, we have:

$$\mathbb{E}_i N_j(t) = \mathbb{E}_j N_i(t) = \mathbb{E}_\pi N_i(t) = \frac{t}{n}. \tag{11}$$

The expectation from the denominator of (10) is bounded by Lemma 2($ii$), $\mathbb{E}_j N_j(t) < 5\sqrt{t} + 2\Delta$. By combining the above relations, and taking into account that $t > \Delta^2$, we obtain:

$$\Pr_i[T_j < t] > \frac{t}{n(5\sqrt{t} + 2\Delta)} > \frac{t}{7n\sqrt{t}} > 0.1\frac{\sqrt{t}}{n}.$$

□

## Proof of Lemma 5

Fixing a connected component $H \subseteq G$ with $n_H \geq n_p/6$, we introduce the following notation for a set of landmarks $L$:

- let $L_H = L \cap V(H)$,
- let $X(L)$ denote the event that $|L_H| \geq \frac{1}{2}p\frac{n_H}{n}$,
- let $F_1(L)$ be the random variable over $L$ describing the maximum, over all non-loops arcs $e$ belonging to $H$, of the probability that a set of $p$ random walks $RW(G_1)$ of length $\tau = n_p^2$ each, with one random walk originating from each landmark from $L$, does not cover arc $e$.
- let $F_2(L)$ be the random variable over $L$ describing the maximum, over all nodes $u \in V(H)$, of the probability that a random walk $RW(G_1)$ of length $\tau = n_p^2$, originating from $u$, does not hit any landmark of $L$.

Suppose that $L$ is a set of $p$ nodes picked according to the uniform distribution $\pi^p$ on $V^p$. To prove the claim of the Lemma, we need to show the following bound:

$$\Pr_{L\sim\pi^p}[F_1 > n^{-1} \wedge F_2 > n^{-1}] < \frac{1}{2n}. \tag{12}$$

We observe that each landmark from $L$ belongs to $V(H)$ with probability $n_H/n$. Let $L_H = L \cap V(H)$. A w.h.p. lower bound on the size of $L_H$ follows from the Chernoff bound applied to $p$ Bernoulli trials with success probability $n_H/n$:

$$\Pr_{L\sim\pi^p}[X] \ \geq 1 - e^{-\frac{1}{8}p\frac{n_H}{n}} \geq 1 - e^{-\frac{1}{48}p\frac{n_p}{n}} \geq 1 - e^{-\frac{\gamma}{48}\log n} > 1 - \frac{1}{4n}, \tag{13}$$

where we took into account that $n_p \geq \gamma\frac{n}{p}\log n$, and that $\gamma = 60 > 48$. In the following, we only need to show that, conditioned on the event $X(L)$ holding, $L$ is a good set of landmarks with probability $1 - \frac{1}{4n}$. Note that all the landmarks from $L_H$ are distributed uniformly at random on $V(H)$, also when conditioned on $X(L)$.

To bound $F_1(L)$, fix a non-loop arc $e$ of $H$ as the arc maximizing the failure probability in the definition of $F_1(L)$. By applying Lemma 3 to graph $H$, the probability that a walk $RW(H_1)$ of length $\tau = n_p^2$, starting from the uniform distribution on $V(H)$, does not cover arc $e$, is at most $1 - \frac{0.1n_p}{n_H}$. Thus, considering that:

$$|L_H| \geq \frac{1}{2}p\frac{n_H}{n} = \frac{n_H \cdot 3\log n}{6\frac{n}{p}\log n} \geq \frac{n_H \cdot 3\log n}{0.1n_p},$$

the probability $F_{1,e}(L)$ that no random walk starting from a landmark hits arc $e$ is bounded by:

$$\mathbb{E}_{L\sim\pi^p}\left[F_1 \mid X\right] < \left(1 - \frac{0.1n_p}{n_H}\right)^{\frac{n_H}{0.1n_p}3\log n} < 2^{-3\log n} < n^{-3}.$$

Likewise, to bound $F_2(L)$, fix a node $u \in V(H)$ maximizing the probability that a walk $RW(G_1)$ of length $\tau = n_p^2$, originating from $u$, does not hit any landmark of $L$. By Lemma 4, the probability that the considered walk of length $\tau$ does not cover a node chosen according to the uniform distribution on $V(H)$, is at most $1 - \frac{0.1\sqrt{\tau}}{n_H}$. Thus, taking into account that $|L_H| > \frac{n_H}{0.1\sqrt{\tau}}3\log n$, the probability that the walk does not hit any landmark can once again be bounded as less than $n^{-3}$:

$$\mathbb{E}_{L\sim\pi^p}\left[F_2 \mid X\right] < n^{-3}.$$

It follows that:

$$\mathbb{E}_{L\sim\pi^p}\left[F_1 + F_2 \;\middle|\; X\right] < 2n^{-3},$$

and by the Markov bound:

$$\Pr_{L\sim\pi^p}[F_1 + F_2 > n^{-1} \mid X] < \frac{2}{n^2} < \frac{1}{4n}. \tag{14}$$

Now, inequalities (13) and (14) imply that inequality (12) holds, which completes the proof. □

## Proof of Proposition 2

We begin by observing that the unbiased random walk on $G$ can be described as a weighted Metropolis-Hastings walk $RW(G_{f_c})$, where, for all $u \in V$, the potential function on nodes is given as $f_c(u) = c\deg(u)$, where $c > 0$ is an arbitrarily chosen constant of proportionality ($w(e) = c$ for all edges). Now, looking at the electrical networks analogy, by identifying with each other the corresponding nodes of the electrical networks describing the walks $RW(G_{f_c})$ and $RW(G_1)$, and leaving the edges of both these networks in parallel connection, we obtain a new network on $G$ with edge weights $w_f$ given by:

$$w_f(e) = w_{f_c}(e) + w(e),$$

corresponding to the potential function on nodes:

$$f(u) = f_c(u) + 1 = c\deg(u) + 1.$$

It follows that the resistance of replacement of the network of $RW(G_f)$ for any two nodes $u, v \in V$ can be bounded as:

$$R_{G_f}(u,v) \leq R_{G_{f_c}}(u,v) \quad \text{and} \quad R_{G_f}(u,v) \leq R_{G_1}(u,v).$$

Moreover, the following relations hold between resistances and commute times:

$$Com_{G_{f_c}}(u,v) = 2cmR_{G_{f_c}}(u,v)$$

$$Com_{G_1}(u,v) = nR_{G_1}(u,v)$$

$$Com_{G_f}(u,v) = (2cm + n)R_{G_f}(u,v)$$

Fixing $c = \frac{1}{d} = \frac{n}{2m}$, i.e., $2cm = n$, we obtain from all of the above relations:

$$Com_{G_f}(u,v) = O(\min\{Com_{G_{f_c}}(u,v), Com_{G_1}(u,v)\}). \tag{15}$$

□