SIAM J. COMPUT.
Vol. 28, No. 3, pp. 1051–1072

A TIME-SPACE TRADEOFF FOR UNDIRECTED GRAPH TRAVERSAL BY WALKING AUTOMATA*

PAUL BEAME†, ALLAN BORODIN‡, PRABHAKAR RAGHAVAN§, WALTER L. RUZZO†, AND MARTIN TOMPA†

Abstract. We prove a time-space tradeoff for traversing undirected graphs, using a structured model that is a nonjumping variant of Cook and Rackoff's "jumping automata for graphs."

Key words. graph connectivity, graph reachability, time-space tradeoff, walking automaton, jumping automaton, JAG

AMS subject classifications. 05C40, 68Q05, 68Q10, 68Q15, 68Q20, 68Q25

PII. S0097539793282947

1. The complexity of graph traversal.

Graph traversal is a fundamental problem in computing, since it is the natural abstraction of many search processes. In computational complexity theory, graph traversal (or, more precisely, *st*-connectivity) is a fundamental problem for an additional reason: understanding the complexity of directed versus undirected graph traversal seems to be the key to understanding the relationships among deterministic, probabilistic, and nondeterministic space-bounded algorithms. For instance, although directed graphs can be traversed nondeterministically in polynomial time and logarithmic space simultaneously, it is not widely believed that they can be traversed deterministically in polynomial time and small space simultaneously. (See Tompa [32] and Edmonds and Poon [22] for lower bounds and Barnes et al. [5] for an upper bound.) In contrast, *undirected* graphs can be traversed in polynomial time and logarithmic space *probabilistically* by using a random walk (Aleliunas et al. [2], Borodin et al. [15]); this implies similar resource bounds on (nonuniform) deterministic algorithms (Aleliunas et al. [2]). More recent work presents uniform deterministic polynomial time algorithms for the undirected case using sublinear space (Barnes and Ruzzo [8]), and even $O(\log^2 n)$ space (Nisan [28]), as well as a deterministic algorithm using $O(\log^{1.5} n)$ space, but more than polynomial time (Nisan, Szemerédi, and Wigderson [29]).

In this paper we concentrate on the undirected case. The simultaneous time and space requirements of the best-known algorithms for undirected graph traversal are as follows. Depth-first or breadth-first search can traverse any n vertex, m edge undirected graph in $O(m + n)$ time, but requires $\Omega(n)$ space. Alternatively, a random walk can traverse an undirected graph using only $O(\log n)$ space, but requires $\Theta(mn)$ expected time (Aleliunas et al. [2]). In fact, Feige [23], based on earlier work

*Received by the editors March 5, 1993; accepted for publication (in revised form) February 7, 1997; published electronically January 29, 1999. This material is based upon work supported in part by the Natural Sciences and Engineering Research Council of Canada, by the National Science Foundation under grants CCR-8703196, CCR-8858799, CCR-8907960, and CCR-9002891, and by IBM under Research Contract 16980043. A portion of this work was performed while the fourth author was visiting the University of Toronto, whose hospitality is gratefully acknowledged.

http://www.siam.org/journals/sicomp/28-3/28294.html

†Department of Computer Science and Engineering, University of Washington, Box 352350, Seattle, WA 98195 (beame@cs.washington.edu, ruzzo@cs.washington.edu, tompa@cs.washington.edu).

‡Department of Computer Science, University of Toronto, Toronto, ON, Canada M5S 1A4 (bor@cs.toronto.edu).

§IBM Research Division, Almaden Research Center, 650 Harry Road, San Jose, CA 95120 (pragh@almaden.ibm.com).

of Broder et al. [18] and Barnes and Feige [7], has shown that there is a spectrum of compromises between time and space for this problem: any graph can be traversed in space S and expected time T, where $ST \le mn(\log n)^{O(1)}$. This raises the intriguing prospect of proving that logarithmic space and linear time are not simultaneously achievable or, more generally, proving a time-space tradeoff that closely matches these upper bounds.

Although it would be desirable to show a tradeoff for a general model of computation such as a random access machine, obtaining such a tradeoff is beyond the reach of current techniques. Thus it is natural to consider a "structured" model (Borodin [14]), that is, one whose basic move is based on the adjacencies of the graph, as opposed to one whose basic move is based on the bits in the graph's encoding. An appropriate structured model for proving such a tradeoff is some variant of the JAG ("jumping automaton for graphs") of Cook and Rackoff [20]. Such an automaton has a set of states, and a limited supply of pebbles that it can move from vertex to adjacent vertex ("walk") or directly to a vertex containing another pebble ("jump"). The purpose of its pebbles is to mark certain vertices temporarily, so that they are recognizable when some other pebble reaches them. The pebbles represent vertex names that a structured algorithm might record in its workspace. Walking represents replacing a vertex name by some adjacent vertex found in the input. Jumping represents copying a previously recorded vertex name.

Rabin (see [20]), Savitch [31], Blum and Sakoda [13], Blum and Kozen [12], Hemmerling [24], and others have considered similar models; see Hemmerling's monograph for an extensive bibliography (going back over a century) emphasizing results for "labyrinths": graphs embedded in two- or three-dimensional Euclidean space.

The JAG is a structured model, but not a weak one. In particular, it is general enough to encompass in a natural way most known algorithms for graph traversal. For instance, a JAG can execute a depth-first or breadth-first search, provided it has one pebble for each vertex, by leaving a pebble on each visited vertex in order to avoid revisiting it, and keeping the stack or queue of pebble names in its state. Furthermore, as Savitch [31] shows, a JAG with the additional power to move a pebble from vertex i to vertex $i+1$ can simulate an arbitrary Turing machine on directed graphs. Even without this extra feature, we have shown [10] that JAGs are as powerful as Turing machines for the purposes of solving undirected graph problems (our main focus).

Cook and Rackoff define the time T used by a JAG to be the number of pebble moves, and the space to be $S = P \log_2 n + \log_2 Q$, where P is the number of pebbles and Q the number of states of the automaton. (Keeping track of the location of each pebble requires $\log_2 n$ bits of memory, and keeping track of the state requires $\log_2 Q$.) It is well known that *st*-connectivity for directed graphs can be solved by a deterministic Turing machine in $O(\log^2 n)$ space, by applying Savitch's theorem [30] to the obvious $O(\log n)$ space nondeterministic algorithm for the problem. Cook and Rackoff show that the same $O(\log^2 n)$ space upper bound holds for deterministic JAGs by direct construction of an $O(\log n)$ pebble, $n^{O(1)}$ state deterministic JAG for directed *st*-connectivity. More interestingly, they also prove a lower bound of $\Omega(\log^2 n/\log\log n)$ on the space required by JAGs solving this problem, nearly matching the upper bound. Standard techniques (Adleman [1], Aleliunas et al. [2]) extend this result to any randomized JAG whose time bound is at most exponential in its space bound. Berman and Simon [11] extend this space lower bound to probabilistic JAGs with even larger time bounds, namely, exponential in $\log^{O(1)} n$.

In this paper we use a variant of the JAG to study the tradeoff between time and space for the problem of *undirected* graph traversal. The JAG variant we consider is more restricted than the model introduced by Cook and Rackoff, because the pebbles are not permitted to jump. This nonjumping model is closer to the one studied by Blum and Sakoda [13], Blum and Kozen [12], and Hemmerling [24]. We will distinguish this nonjumping variant by referring to it as a WAG: "walking automaton for graphs."

Several authors have considered traversal of undirected regular graphs by a WAG with an unlimited number of states but only the minimum number (one) of pebbles, a model better known as a *universal traversal sequence* (Aleliunas et al. [2], Alon, Azar, and Ravid [3], Bar-Noy et al. [4], Borodin, Ruzzo, and Tompa [16], Bridgland [17], Buss and Tompa [19], Istrail [25], Karloff, Paturi, and Simon [26], Tompa [33]). A result of Borodin, Ruzzo, and Tompa [16] shows that such an automaton requires $\Omega(m^2)$ time (on regular graphs with $3n/2 \le m \le n^2/6-n$). Thus, for the particularly weak version of logarithmic space corresponding to the case $P=1$, a quadratic lower bound on time is known.

The known algorithms and the lower bounds for universal traversal sequences suggest that the true time-space product for undirected graph traversal is approximately quadratic, perhaps $\Theta(mn)$. The result of this paper is a lower bound that provides progress toward proving this conjecture. More specifically, we prove lower bounds on time that are nonlinear in m for a wide range of values of P. In particular, for any WAG M solving st-connectivity in logarithmic space, there is a family of regular graphs on which M requires time $m^{1+\Omega(1)}$. Near the other extreme, if M uses a number of pebbles that is sublinear in m, there is a family of regular graphs on which M requires time superlinear in m. Although these give the desired quadratic lower bound only at the extreme of linear time, they each at least establish that logarithmic space and linear time are not simultaneously achievable on the nonjumping model when $m=\omega(n)$. (They do not settle the question of simultaneous achievability of logarithmic space and linear time when $m=O(n)$ since the families of regular graphs mentioned above have degree $d=\omega(1)$ and hence $m=\omega(n)$; see sections 3 and 4.)

We prove upper and lower bounds for undirected graph problems on other variants of the JAG in a companion paper [10]. Following the preliminary appearance of these results, Edmonds [21] proved a much stronger result for traversing undirected graphs, and Barnes and Edmonds [6] and Edmonds and Poon [22] proved even more dramatic tradeoffs for traversing directed graphs.

2. Walking automata for graphs. The problem we will be considering is "undirected st-connectivity": given an undirected graph G and two distinguished vertices s and t, determine if there is a path from s to t.

Consider the set of all n-vertex, edge-labeled, undirected graphs $G=(V,E)$ with maximum degree d. For this definition, edges are labeled as follows. For every edge $\{u,v\}\in E$ there are two labels $\lambda_{u,v},\lambda_{v,u}\in\{0,1,\dots,d-1\}$ with the property that, for every pair of distinct edges $\{u,v\}$ and $\{u,w\}$, $\lambda_{u,v}\ne\lambda_{u,w}$. It will sometimes be convenient to treat an undirected edge as a pair of directed *half-edges*, each labeled by a single label. For example, the half-edge directed from u to v is labeled $\lambda_{u,v}$.

Following Cook and Rackoff [20], a WAG is an automaton with Q states and P distinguishable pebbles, where both P and Q may depend on n and d. For the st-connectivity problem, two vertices s and t of its input graph are distinguished. The P pebbles are initially placed on s. Each move of the WAG depends on the current state, which pebbles coincide on vertices, which pebbles are on t, and the edge labels emanating from the pebbled vertices. Based on this information, the automaton changes state and selects some pebble p and some $i\in\{0,1,\dots,d-1\}$.

The selected i must be an edge label emanating from the vertex currently pebbled by p, and p is moved to the other endpoint of the edge with label i. (The decision to make t "visible" to the WAG but s "invisible" was made simply to render one-pebble WAGs on regular graphs equivalent to universal traversal sequences.) A WAG that determines st-connectivity is required to enter an accepting state if and only if there is a path from s to t. Note that WAGs are nonuniform models.

We have defined WAGs running on arbitrary graphs, but our lower bounds apply even to WAGs that are only required to operate correctly on regular graphs. The restriction to regular graphs, in addition to strengthening the results, provides comparability to the known results about universal traversal sequences. A technicality that must be considered in the case of regular graphs is that they do not exist for all choices of degree d and number of vertices n, as is seen from the following proposition.

PROPOSITION 1. *d-regular, n-vertex graphs exist if and only if dn is even and $d \leq n-1$.*

(See [16, Proposition 1], for example, for a proof.) To allow use of Ω-notation in expressing our lower bounds, however, the "time" used by a WAG must be defined for *all* sufficiently large n. To this end, we consider the time used by a WAG on d-regular, n-vertex graphs where dn is odd to be the same as its running time on d-regular, $(n+1)$-vertex graphs.

3. The tradeoff. In this section we prove time lower bounds for WAGs with P pebbles. The proof generalizes an unpublished construction of Szemerédi (communicated to us by Sipser) that proved an $\Omega(n \log n)$ lower bound on the length of universal traversal sequences for 3-regular graphs.

THEOREM 2. *Let P and d be fixed functions of n with dn even, $P \geq 1$, $d \geq 6$, and $d^2 + Pd = o(n)$. Let $m = dn/2$, $\epsilon = 1/(3\ln(6e))$, and*

$$d_0 = (2P/e)^{3P/(3P+2)}\, n^{1/(3P+2)}.$$

Let M be any (deterministic) WAG with P pebbles that determines st-connectivity for all d-regular, n-vertex graphs. Then M requires time

(a) $\Omega\left(m(\log n)^{\frac{d/P}{\log(d/P)}}\right)$, *if $P \leq \epsilon \ln(n/d^2)$ and $6P \leq d \leq d_0$,*

(b) $\Omega\left(mP\left(\frac{n}{d^2}\right)^{\frac{1}{3P}}\right)$, *if $P \leq \epsilon \ln(n/d^2)$ and $d_0 < d$, and*

(c) $\Omega\left(m \min\left(d, \log\frac{n}{(d^2+Pd)}\right)\right)$, *otherwise.*

Before proving the theorem, we will make a few observations about it. Perhaps the most noteworthy is that these bounds are nonlinear whenever either $d = \omega(1)$ or $d \geq 6P$.

It is obvious that the regions (i.e., the sets of (P, d) pairs) where the three cases apply are pairwise disjoint. It is also true that all three regions are nonempty for all sufficiently large n, although we will not justify this statement.

Although they have very different forms, the three bounds meet "smoothly," except along the line segment $d = 6P$, $1 \leq P \leq \epsilon\ln(n/d^2)$. Specifically, we will show that where any pair of the three bounds meet along the curve $P = \epsilon\ln(n/d^2)$, $d \geq 6P$, both are $\Theta(m\log(n/d^2))$, and where bounds (a) and (b) meet along the curve $d = d_0$, $1 \leq P \leq \epsilon\ln(n/d^2)$, both are $\Theta(md_0)$.

All three bounds are increasing functions of d (recall $m = dn/2$). The ratio of the lower bounds to m is also an interesting quantity. Note that the ratio of bound (a) to m is an increasing function of d, while that of bound (b) is decreasing. Since they are equal (within constant factors) at $d = d_0$, the two could be combined into the single expression $\Omega(m \min((\log n)(d/P)/\log(d/P),\ P(n/d^2)^{1/(3P)}))$, as was done in bound (c).

It seems likely that the decrease in bound (b) is an artifact of the proof technique rather than an intrinsic reduction in the complexity of the problem, since intuitively higher degree would seem to make the search more difficult. On the other hand, higher degree reduces the graph's maximum possible diameter, which perhaps helps. It is known that the length of universal traversal sequences is not monotonic in d, although it may be monotonic up to some large threshold, perhaps $d = \lfloor n/2 \rfloor - 1$. (See Borodin, Ruzzo, and Tompa [16] for a discussion.) Similarly, the complexity of st-connectivity is not monotone in d, since regular graphs of degree $d > \lfloor n/2 \rfloor - 1$ are necessarily connected, but it is plausibly monotone for d up to cn, for some constant $0 < c < 1/2$.

Two special cases of the theorem are of particular interest. Namely, the following two corollaries show that logarithmic space implies time $m^{1+\Omega(1)}$ and that sublinear space implies superlinear time.

COROLLARY 3. *Let M be a WAG with P pebbles that determines st-connectivity for all regular n-vertex graphs. If $P = O(1)$, then there is a family of regular graphs on which M requires time $\Omega(m^{1+1/(3P+3)})$.*

Proof. Consider the family of regular graphs with degree $d = d_0 = \Theta(n^{1/(3P+2)})$. Theorem 2 applies, specifically case (a). This gives a time lower bound of $\Omega(md) = \Omega(m^{1+1/(3P+3)})$. ∎

For $P = 1$ the $\Omega(m^{7/6})$ bound given above is not as strong as the $\Omega(m^2)$ bound given by Borodin, Ruzzo, and Tompa [16] but is included for comparative purposes. Also, the $\Omega(m^2)$ lower bound for universal traversal sequences holds for degree up to $n/3 - 2$, so the decrease in the ratio of bound (b) to m noted above certainly *is* an artifact of our proof when $P = 1$.

COROLLARY 4. *Let M be a WAG with P pebbles that determines st-connectivity for all regular n-vertex graphs. If $P = o(n)$, then there is a family of regular graphs on which M requires time $\Omega(m \log(n/P)) = \omega(m)$.*

Proof. Suppose $P \geq n^{1/3}$. Consider the family of regular graphs with degree $d = \sqrt{n/P} = \omega(1)$. Then $d^2 + Pd \leq 2\sqrt{Pn} = o(n)$, so Theorem 2 applies, specifically case (c). This gives a lower bound of $\Omega(m \log d) = \Omega(m \log(n/P))$ on time. When $P < n^{1/3}$, a similar analysis suffices, choosing $d = \log n$. ∎

Corollary 4 is tight: time $O(m)$ is possible with $O(n)$ pebbles [10, Theorem 15]. Note also that, when $P = \Theta(n)$, the time is still $\Omega(m \log(n/P))$ [10, Theorem 3].

Various constants in the theorem can be improved by slight modification to the construction and/or its analysis, but in the interest of clarity we will not present these refinements.

Proof of Theorem 2. The idea underlying the proof is to build a graph with many copies of some fixed gadgets, each with many "entry points." Since M does not have enough pebbles to mark all the gadgets it has explored, it must spend time reexploring each gadget from different entry points, or it risks the possibility that one of them might never be fully explored. The crux of the argument is to choose the right gadgets and to interconnect them so that we can be sure this happens. We use an "adversary" argument to show this. We begin by giving an overview of the argument, followed by more detailed descriptions of the gadgets and adversary strategy, and finally the analysis.

Overview. Imagine the adversary "growing" the graph as follows. At a general point in the construction, the graph consists of some gadgets that are fully specified except for the interconnections among their "entry point" vertices. The adversary simulates M on this partial graph until M attempts to move some pebble p out of an entry point using a label for which no edge is yet defined. Our main freedom in the construction is the choice of the gadget at the other endpoint of this interconnecting edge f. The adversary will pick it so that M will spend a nonnegligible number of steps τ "exploring" the gadget reached through f. The adversary can achieve this for most of the $\Omega(m)$ interconnecting edges, yielding an $\Omega(m\tau)$ lower bound on time. The parameter τ will vary depending on n, P, and d, giving the three lower bounds quoted in the statement of the theorem.

The interconnecting edge f is chosen as follows. Note that no single labeled gadget γ will suffice to keep p "busy" for τ steps. For example, M's very next move of p, say by label a, might be an exit from γ. On the other hand, if the adversary can learn that M's next move of p will be on label a, it can choose some gadget in which label a moves from an entry point *into* the gadget, rather than exiting from it. Similarly, if it can learn the next τ moves by p (and/or other pebbles following p across f), the adversary can choose a gadget in which this whole sequence of moves avoids exiting from the gadget. A key point is that M can sense only very limited facts about the gadget that p enters when it crosses f. Suppose p has just crossed f, arriving at a vertex v. M can sense (i) the degree of v, (ii) whether v is the target vertex t, and (iii) whether there are other pebbles on v. Thus, in general M has several possible next moves for p, based on which of these conditions hold. The adversary avoids having to consider these alternative futures by assuming, respectively, (1) that the graph is d-regular, (2) that M does not reach t (within $\Omega(m\tau)$ steps), and (3) that f connects to a gadget that contains no other pebbles when p enters it, and that remains free of other pebbles (except perhaps ones that follow p across f) for τ moves. Given these assumptions, the adversary will be able to deterministically simulate the next several moves by M so that it can decide which labeled gadget can host those moves without allowing a pebble to exit. Of course, the adversary must also ensure that assumptions (1)–(3) are ultimately justified. Building a d-regular graph requires some care but is not too difficult. Assumption (2) will follow easily if each connecting edge accounts for τ moves. Assumption (3) is slightly trickier; we will return to it below.

We view the overall adversary strategy as a two-phase process. A *local* phase determines the internal ("local") structure required of a gadget hosting the next several moves of p so that no pebble will exit this gadget until at least τ moves have been charged to it, starting after p's entry. The basic idea is to use a "lazy, greedy" definition: lazy in that the adversary will not define a labeled half-edge in the gadget until just before M needs to move a pebble across it, and greedy in that when such a half-edge is defined, it will be defined to stay within the gadget. Of course, this cannot continue indefinitely, but it will be possible for at least the first τ moves within the gadget. Thus, pebble motion across half-edges exiting the gadget is deferred for at least this long.

The adversary's simulation of M is now "rolled back" to the point at which p crossed f. The *global* phase of the adversary's strategy is to choose a gadget already present in the graph and to connect f to it. Recall that our goal is to reuse each gadget many times, so that the total time spent in it asymptotically exceeds its number of edges. (Occasionally, when all entry points of suitable gadgets have been used, a new copy of the needed gadget will be added. This process terminates when the number of vertices in the graph approaches n.) The gadget chosen for f must match the gadget determined by the local phase, must have an unused entry point to which to connect f, and (before f was connected to it) must have remained free of pebbles from the time when p crossed f until τ moves were charged to it. The "pebble-free" condition ensures assumption (3) above. Such a condition is necessary since, if it were violated, M might encounter "unexpected" pebbles in the chosen gadget, i.e., pebbles not encountered during the simulation in the local phase. This could cause M to deviate from the sequence of moves predicted by the local phase, and so possibly allow p or one of the pebbles that followed it across f to exit from the gadget in fewer than τ moves.

A point we slighted above is that the "τ steps" under discussion are not necessarily consecutive and are not necessarily all made by p or by pebbles that followed p across f. For example, p's moves after crossing f might be interleaved with moves by some other pebble p' after crossing another undefined edge f' and/or many previously defined connecting edges. In general, the adversary keeps track of these many interleaved activities by charging pebble moves to connecting edges, with the "local phase" for an undefined connecting edge f being the interval between its charge reaching 1 (at the first crossing of f by some pebble) and its charge reaching τ.

The final issue to address is that we want to avoid adding a new copy of a gadget until all entry points of most existing copies have been used. Specifically, we will have at most a fixed number ($P(\tau + 2) + d$, to be precise) of "open" copies of each gadget at any time. As noted above, many steps may occur between the first and τth steps charged to f. During this interval, other pebbles might touch *all* open copies of the gadget needed for f, leaving no *pebble-free* open gadget to which to connect f. Our solution to this problem is found in the adversary's method of charging pebble moves to edges. Moves in f's gadget are always charged to f. In addition, certain moves touching other gadgets are charged to f also. With this scheme, we can bound both the number of moves that occur in f's gadget and the number of other gadgets that are touched by pebbles during f's local phase. Thus, no gadget is expected to absorb too many moves, and there will be at least one suitable pebble-free open copy of the needed gadget when f accumulates charge τ.

The construction will "waste" (i.e., not fully utilize the connecting edges of) up to $P(\tau + 2) + d$ copies of each gadget. The main constraint that limits τ is that it must be small enough that this waste is small, i.e., $P(\tau + 2) + d$ times the number of distinct types of labeled gadgets times the number of connecting edges per gadget is small compared to the total number of connecting edges.

We will now present the construction in more detail. We actually define a sequence of graphs $G_{i,j}$, $0 \le i \le \mu$, $0 \le j$, representing successive phases of the construction. Like τ, the parameter μ varies slightly depending on n, p, and d, but will be $\Theta(m)$ in each case. (The maximum value of j is unimportant, but turns out to be about $P\tau$.) Each graph consists of the following:

1. A set of gadgets, each with the same size S and number L of entry vertices, and a fully defined internal structure and labeling. Each vertex that is not an entry vertex has degree d. There is a fixed $d' \ge 1$ such that each entry vertex has d' edges to neighbors in the same gadget, and up to $d - d'$ *connecting edges* joining it to the entry vertices of other gadgets or protogadgets (see below). We will show that $d - d' \ge d/2$ and that $L/S > 1/3$, ensuring at termination that the number of connecting edges is $\Theta(m)$.

2. A set of labeled *committed* connecting edges joining entry vertices of gadgets. $G_{i,j}$ will have exactly i committed connecting edges.

3. A set of up to P partially labeled *uncommitted* connecting edges, each joining an entry vertex u of some gadget to an entry vertex v of a protogadget (see below). The uncommitted half-edge from u to v is labeled, but the half-edge from v to u is unlabeled.

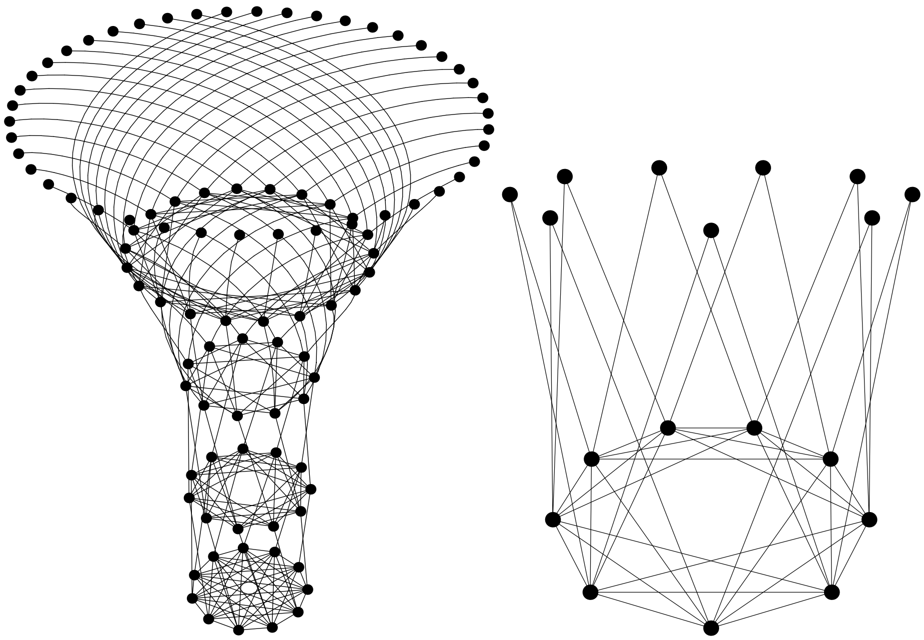
4. A set of up to P partially defined *protogadgets*. Like a gadget, a protogadget has S vertices, including L entry vertices, but unlike the gadgets, the internal structure of a protogadget is in general only partially defined; its vertices may have degree less than d, and its half-edges may not be labeled. In particular, only one entry vertex v of each protogadget will be incident to a connecting edge, say the uncommitted connecting edge $\{u, v\}$, and, as indicated above, the half-edge from v to u will be unlabeled. The protogadgets are the tools used in the local phases of the adversary's strategy.

In outline, the adversary's strategy is as follows. The initial graph $G_{0,0}$ consists of one arbitrarily chosen gadget. The start vertex s is an arbitrary vertex in this gadget. For any $G_{i,j}$, the *initial configuration* of M on $G_{i,j}$ consists of M in its start state and all P pebbles on $G_{i,j}$'s copy of s. Associate with each connecting edge of the graph $G_{i,j}$ an integer *charge*, initially zero. The adversary will charge each pebble motion to at most one connecting edge, according to a rule to be given later. It will simulate M starting from M's initial configuration on $G_{i,j}$ until one of the following two things happens. (It simulates M as if all vertices in $G_{i,j}$ were of degree d, even though some are of smaller degree.)

1. Suppose M attempts to move a pebble from a vertex u across a half-edge labeled a, where no such labeled half-edge exists. If u is an entry vertex in some gadget, add a new uncommitted half-edge from u labeled a to the entry point v of a new protogadget. More precisely, we define $G_{i,j+1}$ to be $G_{i,j}$ plus that half-edge and protogadget. If u is in some protogadget, choose some other vertex v in the *same* protogadget (according to a rule to be given later) and add a half-edge from u to v labeled a. More precisely, we define $G_{i,j+1}$ to be $G_{i,j}$ plus that half-edge (plus a few others, as we will see). The choice of v is not arbitrary; one point we must establish is that there will always be a suitable vertex v when needed. The thrust of this step in the adversary strategy is to keep pebbles "trapped" in protogadgets for as long as possible. This portion of the adversary's strategy is the "local" strategy introduced above, so-called because of its focus on the structure *within* a gadget.

2. Suppose an uncommitted edge f in $G_{i,j}$ accumulates a charge of τ. In this case, we will convert f into a committed edge. More precisely, we will form $G_{i+1,0}$ from $G_{i,j}$ by choosing an existing gadget "similar" to f's protogadget and committing f to enter the chosen gadget. (This is described more fully below.) Again, f cannot be committed arbitrarily; a second point that we must establish is that an appropriate gadget (usually) exists when needed, and that M's behaviors on $G_{i,j}$ and $G_{i+1,0}$ are similar. The thrust of this step is that the size of $G_{i+1,0}$ is growing slowly with i, since we are (usually) able to reuse existing gadgets, but the time M spends in $G_{i+1,0}$ is rising rapidly with i, since a lower bound on the total running time of M is τ times the number of committed edges (i, which is ultimately $\mu = \Theta(m)$). This portion of the adversary's strategy is the "global" strategy, so-called because of its focus on the interconnections among gadgets.

The adversary continues the simulation on $G_{i,j+1}$ or $G_{i+1,0}$ as appropriate, and repeats this process until $G_{\mu,0}$ is constructed.



(a)
$d = 10,\ k = 4,\ q = 2,\ r = 1$
For clarity, only half of the forward edges are shown.

(b)
$d = 8,\ k = 1,\ q = 2,\ r = 1$

FIG. 1. *Examples of the funnel gadgets.*

Gadgets. Before describing the adversary strategy in more detail, we will describe the gadgets and protogadgets. The gadgets are called "funnels." An example is shown in Fig. 1a. The entry vertices are those on the "rim" of the funnel. Intuitively, the adversary will try to "trap" pebbles in a funnel for a while by assigning edge labels so that the moves taken by pebbles in the gadget in the near future (i.e., the next τ moves in the gadget) either stay on the same layer or drop to the next deeper layer. The "cone" portion of the funnel (near the top of Fig. 1a) allows many entry vertices to share vertices in the narrower portion near the bottom of Fig. 1a. An example of a two-layer funnel is shown in Fig. 1b.

Four interrelated parameters $k, q, g,$ and r, which in turn depend on $n, P,$ and d, characterize the gadgets. All four are positive integers. Each gadget has $k+1$ *layers,* numbered 0 through k. Layer $l,\ 0 \le l \le k$, has

$$n_l = (d+1) \cdot \max(1, 2^{\lceil \log_2 k \rceil - l})$$

vertices, designated v_i^l, $0 \le i \le n_l - 1$. The entry vertices are those on layer 0. Hence, the number of entry vertices is

$$L = (d+1) \cdot 2^{\lceil \log_2 k \rceil},$$

and the total number of vertices per gadget is

$$S = (d+1)(2^{\lceil \log_2 k \rceil + 1} - 1 + k - \lceil \log_2 k \rceil).$$

Note that

$$\frac{L}{S} = \frac{(d+1)\cdot 2^{\lceil \log_2 k\rceil}}{(d+1)(2^{\lceil \log_2 k\rceil+1}-1+k-\lceil \log_2 k\rceil)} > \frac{2^{\lceil \log_2 k\rceil}}{2^{\lceil \log_2 k\rceil+1}+k} \geq \frac{1}{3}, \tag{1}$$

as promised, and that

$$S \leq (1+1/d)\, d\,(2\cdot 2^{\lceil \log_2 k\rceil}+k) \leq (7/6)\, d\,(5k) < 6dk. \tag{2}$$

The parameter q is an even integer, $2 \leq q$. The d edge labels $\{0,1,\ldots,d-1\}$ are partitioned into $g = \lfloor d/q \rfloor$ "full" blocks, each of size q, plus perhaps one "partial" block of size $d \bmod q$ in case q does not evenly divide d. The same fixed partition is used for all gadgets and is arbitrary, except that for each $a \in \{0,1,\ldots,d-1\}$, we place both a and its mate in the same block, where the *mate* of label a is $d-1-a$. Note that if d is odd, then $(d-1)/2$ is its own mate and will be in the partial block.

The remaining gadget parameter r is an integer satisfying $1 \leq r \leq g/3$. Note that the existence of such an r implies that $g \geq 3$, and hence

$$q \leq d/3. \tag{3}$$

Intuitively, r denotes an upper bound on the number of pebbles that we attempt to trap in a given gadget.

The edges within a gadget always connect vertices on the same or adjacent layers. A half-edge is called a "forward" half-edge if it goes from layer l to layer $l+1$, "backward" if it goes to layer $l-1$, and "cross" if it goes to layer l. For each layer l and each block B of labels, there is a $t \in \{\text{forward}, \text{backward}, \text{cross}\}$ such that all half-edges with labels in B leaving vertices on layer l will be of type t. Thus it is natural to refer to the labels and the blocks of labels at a layer as forward, backward, or cross, as well as the half-edges. For $i \in \mathbb{N}$, $a \in \{0,1,\ldots,d-1\}$, and $0 \leq l \leq k$, define

$$\chi(i,a,l) = \begin{cases} (i+a+1) \bmod n_l & \text{if } a < (d-1)/2, \\ (i+n_l/2) \bmod n_l & \text{if } a = (d-1)/2, \\ (i-(d-1-a)-1) \bmod n_l & \text{if } a > (d-1)/2. \end{cases}$$

If $a \in \{0,1,\ldots,d-1\}$ is a forward label at layer l, then for $0 \leq i \leq n_l - 1$, a will label the half-edge from vertex v_i^l to vertex $v_{\chi(i,a,l+1)}^{l+1}$. Similarly, if a is a cross label it will go to $v_{\chi(i,a,l)}^{l}$. Notice that a cross edge labeled a will be labeled by a's mate in the reverse direction. No parallel edges arise since $n_l \geq d+1$. As an example, if all edges are cross edges (a case that does not arise in our constructions) and if $n_l = d+1$, then layer l would be a $(d+1)$-clique. As another example, whenever label 0 is a cross label at layer l, the half-edges labeled 0 will form a Hamiltonian cycle through the layer l vertices, and those edges will be labeled $d-1$ (0's mate) in the other direction. Note that the backward labels are not constrained by χ.

The set of gadgets is defined as follows. For $0 \leq l \leq k$ let

$$b_l = \begin{cases} g - r & \text{if } l = 0, \\ r(n_{l-1}/n_l) & \text{otherwise}, \end{cases}$$

$$f_l = \begin{cases} 0 & \text{if } l = k, \\ r & \text{otherwise}. \end{cases}$$

See Table 1. If q does not evenly divide d, then the labels in the partial block will be cross labels at each layer $0 \leq l \leq k$. For each layer $0 \leq l \leq k$, choose f_l of

TABLE 1
Number of edge blocks of each type per layer.

Layer	0	1	$\cdots$	$\lceil \log_2 k \rceil$	$\lceil \log_2 k \rceil + 1$	$\cdots$	$k-1$	k
f_l	r	r	$\cdots$	r	r	$\cdots$	r	0
b_l	$g-r$	$2r$	$\cdots$	$2r$	r	$\cdots$	r	r
Cross	0	$g-3r$	$\cdots$	$g-3r$	$g-2r$	$\cdots$	$g-2r$	$g-r$

the remaining g blocks as forward labels and b_l as backward labels (connecting edge labels, if $l = 0$). All blocks not selected above will be cross labels. Note that the rules in the previous paragraph define the forward and cross half-edges, given their labels, but not the backward half-edges. The chosen backward labels are assigned to these half-edges in an arbitrary but fixed way. Note that there are just enough backward labels: each of the n_{l-1} vertices on level $0 \le l-1 < k$ has exactly qr forward labels, with destinations evenly distributed over the n_l vertices on layer l, so each vertex on layer l is incident to exactly $qr(n_{l-1}/nl) = q \cdot b_l$ edges from layer $l-1$.

For layer 0, the b_0 blocks selected above will label connecting edges. Thus, each entry vertex will be adjacent to exactly $d' = rq + (d \bmod q)$ other vertices in the same gadget, and to $d - d'$ connecting edges. Note, since $r \le g/3$ and $q \le d/3$ (from inequality (3)), that

$$d - d' = gq - rq \ge (2/3)gq = (2/3)\lfloor d/q \rfloor q \ge (2/3)((3/4)(d/q))q = d/2, \tag{4}$$

as claimed earlier. Also note that at most $3r$ blocks are chosen as forward and backward at each layer, and that this is always possible since $g \ge 3r$.

The number of distinct gadget types created by this process is

$$\begin{aligned} &\binom{g}{r}^k \binom{g-r}{2r}^{\lceil \log_2 k \rceil} \binom{g-r}{r}^{k-\lceil \log_2 k \rceil - 1} \binom{g}{r}^1 \\ &\quad \le \binom{g}{r}^{2k-\lceil \log_2 k \rceil} \binom{g}{2r}^{\lceil \log_2 k \rceil} \\ &\quad \le \left(\frac{eg}{r}\right)^{r(2k-\lceil \log_2 k \rceil)} \left(\frac{eg}{2r}\right)^{2r\lceil \log_2 k \rceil} \\ &\quad \le \left(\frac{eg}{r}\right)^{r(2k+\lceil \log_2 k \rceil)}. \end{aligned} \tag{5}$$

Figure 1b fully shows a gadget with $d = 8$, $k = 1$, $q = 2$, $g = 4$, and $r = 1$, with forward edges labeled 0 and 7 from layer 0 and backward edges labeled 3 and 4 from layer 1. Figure 1a shows a gadget with $d = 10$, $k = 4$, $q = 2$, $g = 5$, and $r = 1$, with forward edges labeled 0 from layers 0 through 3. In the interest of clarity, the forward edges labeled 9 (0's mate) are not shown in the figure.

Protogadgets and local strategy. The protogadgets are built incrementally by the adversary. Initially, each consists of S vertices, denoted as in the gadgets, together with the cross edges defined by the partial block of labels (if any) at each level. As discussed previously, the adversary proceeds by simulating M from its initial configuration on $G_{i,j}$. Suppose during the tth step of this simulation that M attempts to move some pebble p along the half-edge labeled a from some vertex u but no such half-edge exists. As sketched earlier, if u is an entry vertex of some gadget, we create a new protogadget into which p will move. If u is a vertex v_i^l in some protogadget

π, the adversary decides whether to make the block of labels containing a all forward half-edges or all cross half-edges (see below). The graph $G_{i,j+1}$ is then defined to be the same as $G_{i,j}$, except that at layer l in π, a's block of half-edges is added. The adversary restarts the simulation of M, starting from M's initial configuration on $G_{i,j+1}$. It should be clear that during the first $t-1$ steps of the simulation, M will behave on $G_{i,j+1}$ exactly as it did on $G_{i,j}$, since $G_{i,j}$ is a subgraph of $G_{i,j+1}$. The tth step, of course, was impossible in $G_{i,j}$, but is possible in $G_{i,j+1}$. Note that p can exit from π only at an entry vertex but is no nearer to such a vertex in $G_{i,j+1}$ after the tth step than before. Thus we can view M as running on a dynamically growing graph, one being built by the adversary so as to trap pebbles in protogadgets for some number of moves. We will adopt this view when no confusion will arise and let $G_{i,*}$ denote the last $G_{i,j}$ built before $G_{i+1,0}$.

Let z be the number of free blocks at level l, i.e., blocks whose half-edges have not yet been defined. The adversary chooses a's block to label cross edges provided $z > b_l + f_l$ and forward edges provided $b_l < z \le b_l + f_l$. If $z \le b_l$, the adversary *fails* (but see Claim 1 below).

Let

$$\tau = \begin{cases} (k - \lceil \log_2 k \rceil) \lfloor g/3 \rfloor & \text{if } P \le r, \\ r & \text{if } P > r. \end{cases}$$

Note that $(k - \lceil \log_2 k \rceil) \lfloor g/3 \rfloor \ge r$ since $k \ge 1$ and $g/3 \ge r$, so in either case we have

$$(k - \lceil \log_2 k \rceil) \lfloor g/3 \rfloor \ge \tau. \tag{6}$$

We prove three claims about the protogadgets. We will see later that the global strategy prevents M from making more than τ moves in any protogadget, so Claim 1 below shows that the adversary will never fail.

CLAIM 1. *The adversary will never fail, provided M makes at most τ moves in any protogadget.*

Proof. First, clearly at most $\min(P, \tau)$ pebbles can enter a protogadget in τ steps, and for the particular definition of τ chosen above, $\min(P, \tau) \le r$. Now, suppose the claim is false. Suppose the adversary first fails during an attempted move at level l in some protogadget π. Then at least $g - b_l$ moves were previously made by pebbles at layer l. As noted, at most r pebbles can enter π in τ moves. It cannot be the case that $l < k$, since for all such layers $g - b_l \ge r = f_l$, so during the last r of the $g - b_l$ moves, all r pebbles moved past layer l, leaving none to cause failure there. Thus, the failure occurred in layer k. For a pebble to reach layer k, it must be that the maximum number of cross edges, plus at least one forward edge, have been previously defined at each layer less than k. Thus, the number of moves completed in this protogadget prior to failure is at least

$$\begin{aligned}
&(g - b_k) + \sum_{l=0}^{k-1} (g - b_l - f_l + 1) \\
&= (g - r) + \lceil \log_2 k \rceil (g - 3r + 1) + (k - \lceil \log_2 k \rceil - 1)(g - 2r + 1) \\
&\ge (k - \lceil \log_2 k \rceil)(g - 2r + 1) \\
&> (k - \lceil \log_2 k \rceil) \lfloor g/3 \rfloor \\
&\ge \tau.
\end{aligned}$$

The second inequality uses the assertion that $r \le g/3$, and the third uses inequality (6). ∎

CLAIM 2. *Each protogadget is a subgraph of some gadget.*

Proof. The adversary chooses at most f_l forward blocks and at most $g - (f_l + b_l)$ cross blocks at each layer. Thus there are enough unchosen blocks to select a total of exactly f_l forward and b_l backward blocks, which precisely defines a gadget. ∎

CLAIM 3. *All entry points of a protogadget are equivalent in the sense that if M makes at most τ moves in a protogadget entered through vertex v_h^0, then the resulting configuration will be exactly the same as if it had entered through vertex v_0^0, except that positions of all pebbles in it on layer l will be shifted by $h \bmod n_l$ for $0 \le l \le k$.*

Proof. Intuitively, this reflects the rotational symmetry of the funnel. To make this precise, we claim that for any $h \in \mathbb{N}$ and any protogadget π, the mapping $\phi_h(v_i^l) = v_{i'}^l$, where $i' = (i + h) \bmod n_l$, is an automorphism on π, i.e., a surjection on the vertices of π preserving labeled half-edges. Consider a forward edge labeled a at level l in π, say (v_i^l, v_j^{l+1}), where $j = \chi(i, a, l + 1)$. Note that for each fixed a and l, there is a constant c (depending on a and l but independent of i) such that $\chi(i, a, l + 1) = (i + c) \bmod n_{l+1}$. Now $\phi_h(v_i^l) = v_{i'}^l$, $\phi_h(v_j^{l+1}) = v_{j'}^{l+1}$, with $i' = (i + h) \bmod n_l$, and $j' = (j + h) \bmod n_{l+1}$, so since n_{l+1} divides n_l we have

$$\begin{aligned}
\chi(i', a, l + 1) &= (i' + c) \bmod n_{l+1} \\
&= (((i + h) \bmod n_l) + c) \bmod n_{l+1} \\
&= (i + h + c) \bmod n_{l+1} \\
&= (((i + c) \bmod n_{l+1}) + h) \bmod n_{l+1} \\
&= (j + h) \bmod n_{l+1} \\
&= j'.
\end{aligned}$$

A similar argument applies to cross edges. ∎

The analog of Claim 3 also holds for gadgets, provided the τ moves use only forward and/or cross edges. The same may not be true if backward edge labels are used.

Global strategy. We have now described the gadgets and protogadgets and the adversary's strategy for building them. We turn to the remaining part of its strategy: charging and committing edges. Recall that the adversary associates a charge with each connecting edge, in which it counts moves in $G_{i,*}$. In addition, it associates with each connecting edge a second integer, called a *birthdate*, recording the time at which a pebble first crosses the edge.

The construction of $G_{i+1,0}$ from $G_{i,*}$ proceeds as follows. The adversary begins with M in its *initial* configuration in the current graph $G_{i,*}$. The adversary simulates successive moves of M on $G_{i,*}$ until some uncommitted connecting edge accumulates charge τ, where edge charges are determined by the following rules. During a move, suppose M moves pebble p along

1. an edge internal to a gadget or protogadget. Let f be the connecting edge most recently crossed by p. If f has charge less than τ, then charge the move to f; otherwise there is no charge.
2. a connecting edge f (committed or not). Charge the move to the oldest (i.e., least birthdate) connecting edge having charge less than τ. If this is the first step in which a pebble has crossed edge f in either direction, define the birthdate of f to be the current time.

As sketched in the overview, the second charging rule ensures that when an uncommitted connecting edge f, even one whose associated pebbles have moved infrequently, accumulates charge τ, only a few of the gadgets of the appropriate type can have been touched by pebbles since the birth of f.

When some uncommitted connecting edge $f = \{u, v\}$ with label $\lambda_{u,v} = a$ accumulates charge τ we stop the simulation and construct from $G_{i,*}$ a new graph $G_{i+1,0}$ defined as follows. Let π_v be the protogadget entered through f, with v in π_v. Note that by the charging rules above, each move in π_v has been charged to f, so there have been at most τ such moves. Thus by Claim 1 the adversary did not fail while building π_v. By Claim 2, the protogadget π_v is a subgraph of some gadget γ_v. We say an entry vertex of a gadget is *open* if it has degree less than d. If possible, choose an entry vertex x of a gadget in $G_{i,*}$ such that

(7)
- x is open,
- x's gadget is of the same type as γ_v,
- x and u are not adjacent, and
- x's gadget has remained pebble free since the birthdate of the uncommitted edge f.

$G_{i+1,0}$ is identical to $G_{i,*}$, except that the protogadget π_v is removed and the uncommitted edge $f = \{u, v\}$ is replaced by the committed edge $\{u, x\}$ with labels $\lambda_{u,x} = a$ and $\lambda_{x,u} = b$, where b is any label not already present on an outgoing half-edge at x. If there is no such x, or if using the only such x would result in $G_{i+1,0}$ having neither uncommitted edges nor open entry vertices, we instead add one additional gadget of type γ_v, choose as x any of the new gadget's entry vertices, then proceed as described above. The latter contingency avoids premature termination of the construction. The requirement that x and u be nonadjacent avoids construction of parallel edges.

The behavior of M on $G_{i+1,0}$ is similar to its behavior on $G_{i,*}$. Suppose in $G_{i,*}$ that the uncommitted edge f was first crossed during the simulation of the bth move of M (i.e., has birthdate b) and accumulates charge τ during move b'. When M is simulated on $G_{i+1,0}$, it will behave exactly as on $G_{i,j}$ for the first $b - 1$ moves, since the portion of $G_{i+1,0}$ visited during that period is exactly the same as the portion visited in $G_{i,*}$. In particular, the charges and birthdates attached to edges will be the same. (Thus, one can view the adversary as rolling back the simulation to step b, committing f, and resuming.) Between steps b and b' those pebbles that crossed edge f in $G_{i,*}$ will be in x's gadget γ_x in $G_{i+1,0}$ instead of in the protogadget π_v entered through f as they were in $G_{i,*}$, but since γ_x contains π_v as a subgraph, their motions in $G_{i+1,0}$ will exactly reflect their motions in $G_{i,*}$. Note that by Claim 3 this is true regardless of which entry vertex x of γ_x was chosen. It is crucial that the chosen gadget γ_x was pebble free between steps b and b', so there is no possibility that these pebbles will meet pebbles in γ_x in $G_{i+1,0}$ that they did not meet in π_v in $G_{i,*}$. Again, the charges and birthdates attached to edges will be the same in $G_{i+1,0}$ as in $G_{i,*}$ through step b'. In particular, each of the $i + 1$ committed edges in $G_{i+1,0}$ will have a charge of τ, and hence M will run for at least $(i + 1)\tau$ steps on $G_{i+1,0}$.

Final Construction. After $G_{i+1,0}$ is built, we restart the simulation from the beginning on $G_{i+1,0}$ to build $G_{i+2,0}$, etc. Continue this process until $G_{\mu,0}$ is constructed. Finally, from $G_{\mu,0}$ we build a pair of similar graphs G and G', one connected and the other not, on which M will have identical behavior. In particular, if M runs for fewer than $\mu \cdot \tau$ steps, then M cannot be correct on both. The connected graph G is built by

1. committing all uncommitted edges, as described above;
2. joining the remaining open entry vertices with some number, Δ, of extra vertices so as to make G have n vertices and be d-regular; and
3. designating one of these extra vertices as t.

One way to accomplish the second step is the following. First, pick any two nonadjacent open vertices and connect them. Repeat this as often as possible. Let u be the number of "missing" half-edges, i.e., the total over all open vertices of d minus their degrees, and let i be the number of remaining open vertices. Since the pairing process could not be applied to reduce i further, it must be the case that the i open entry vertices form a clique. Recalling that each entry vertex is incident to at most $d - d'$ connecting edges, the number u of missing half-edges can be at most

$$i((d - d') - (i - 1)) \leq i(d - i) \leq d^2/4,$$

since $d' \geq 1$ and since $i(d-i)$ is maximized when $i = d/2$. Thus $u \leq d^2/4$. Furthermore, u will necessarily be even, since each entry vertex starts with $d-d'$ missing half edges; since from (4) $d - d'$ is a multiple of q, hence even; and since each committed edge replaces a pair of missing half-edges. Notice that this implies that $d(n - \Delta)$ is even, since the gadgets together contain $n - \Delta$ vertices and $d(n - \Delta) - u$ half-edges, which naturally occur in pairs. Complete the construction by adding a Δ-vertex, d-regular graph that contains a $u/2$-matching, removing the edges of this matching, and connecting each of the u missing half-edges to a distinct endpoint of the matching. Such a regular graph exists by Proposition 1, since dn, $d(n - \Delta)$, and hence $d\Delta$ are even; since, as shown below, $d < \Delta$ and $u \leq d^2/4 < \Delta$; and since the proof of Proposition 1 given in Borodin et al. [16] constructs a regular graph that is Hamiltonian and hence has a $u/2$-matching. (That construction is similar to the construction of cross edges in one layer of our gadgets, where the 0-labels form a Hamiltonian cycle.)

The nonconnected graph G' is built similarly, except that $d + 1$ of the Δ extra vertices, including t, are connected in a clique and hence are disconnected from the rest of the graph.

By an argument similar to the one above, M's behavior on both G and G' is essentially the same as on $G_{\mu,0}$. In particular, the edge charges will be the same, so M will run for at least $\mu{\cdot}\tau$ steps without reaching any of the Δ extra vertices, including t. One point to be shown in the analysis below is that $\Delta \geq d+1+\max(d+1, d^2/4) = d^2/4+d+1$, i.e., large enough to allow completion of the construction of G and G' as described above. Since $d \leq \sqrt{n}-2$ (in fact, $d^2 + Pd = o(n)$), it suffices that $\Delta \geq n/4$.

Analysis. All that remains to show our $\Omega(m\tau)$ lower bound is to give values for the various parameters so as to satisfy the constraints listed above (and to maximize τ). For convenience, we summarize the relevant parameters and constraints here.

C1. Number of committed connecting edges: $\mu = \Omega(m)$.
C2. Number of vertices added in the final step of the construction: $\Delta \geq n/4$.
C3. Number of layers per gadget: $k \geq 1$.
C4. Size of full blocks in the label partition: $q \geq 2$, even.
C5. Number of full label blocks: $g = \lfloor d/q \rfloor$.
C6. Upper bound on the number of pebbles entering a protogadget: $1 \leq r \leq g/3$.
C7. Time per committed edge: τ; if $P \leq r$ then $\tau = (k - \lceil \log_2 k \rceil) \lfloor g/3 \rfloor$; else $\tau = r$.

To satisfy constraint C1, choose

$$\mu = \lfloor dLn/(8S) \rfloor . \tag{8}$$

Since we have seen in inequality (1) that $L/S = \Omega(1)$, we have $\mu = \Omega(m)$ as claimed above.

We now turn to constraint C2. We say a gadget is *closed* if each of its L entry vertices is connected to the maximum number $d-d'$ of committed half-edges; otherwise the gadget is *open.* $G_{\mu,0}$ has exactly μ committed edges, or 2μ committed half-edges. From (4), each closed gadget contributes $(d-d')L \geq dL/2$ committed half-edges, so by (8) there can be no more than $2\mu/(dL/2) \leq n/(2S)$ closed gadgets in $G_{\mu,0}$, each of size S, and so closed gadgets contribute no more than $n/2$ vertices to G. Thus, to ensure constraint C2, i.e., that Δ is at least $n/4$, it suffices to ensure that the following additional constraint holds:

C8. Number of vertices in open gadgets: must be at most $n/4$.

When building $G_{i+1,0}$ from $G_{i,*}$, the adversary replaces a protogadget π by a copy of a fixed gadget γ. There might be many copies of the gadget γ with which π can be replaced. A key claim in establishing constraint C8 is that there are never more than $P(\tau+2)+d$ open copies of such a gadget.

CLAIM 4. *When $G_{i+1,0}$ is defined, if there are $P(\tau+2)+d$ open copies of the gadget γ_v, then at least one of them will have an entry vertex x satisfying the conditions* (7), *so a new (open) gadget will* not *be introduced into $G_{i+1,0}$.*

Proof. We show an upper bound on the number of open gadgets that are disqualified from containing x. It is easy to see that at most $d-d' \leq d-1$ entry vertices are adjacent to u in $G_{i,*}$. A more subtle problem is to bound the number of gadgets that can be touched by pebbles between the birth of π's uncommitted connecting edge f and the time at which f has accumulated charge τ. At most P gadgets contain pebbles at the time of f's birth. At most $P-1$ edges older than f can have charge less than τ, because, by Claim 1, for each such edge f' there is at least one pebble that does not leave its gadget or protogadget until f' has accumulated charge τ. Each gadget touched by some pebble after the birth of f necessitates the crossing of some connecting edge. Thus after at most $(P-1)\tau$ such crossings, f will be the oldest uncommitted edge, and after at most τ more crossings, f will have charge τ. Thus, at most $P(\tau+1)$ gadgets can be touched by pebbles during the relevant interval. Finally, at all times, at most P open gadgets are incident to uncommitted half-edges, hence at most P lack open entry vertices. Thus, the number of vertices x not disqualified is at least $P(\tau+2)+d-(d-1)-P(\tau+1)-P=1$, which establishes the claim. ∎

Inequality (5) bounds the number of distinct gadget types, Claim 4 bounds the number of open copies of each, and inequality (2) bounds the size of each copy. Thus, the total number of vertices in open gadgets is at most

$$\left(\frac{eg}{r}\right)^{r(2k+\lceil\log_2 k\rceil)} (P(\tau+2)+d)\, 6dk. \tag{9}$$

We divide the remainder of the analysis into two cases. The second applies when P is small. The first applies to either small or large P but gives a weaker bound than the second for small P.

Case 1. Let $\delta = 3\epsilon/2 = 1/(2\ln(6e))$, let

$$\beta = 72\left(d^2 + Pd\ln\frac{n}{d^2+Pd}\right),$$

and suppose n, P, and d are such that $d^2+Pd \leq n/e$ and $\beta \leq n/e^{6/\delta}$, both of which are true for all sufficiently large n, since $d^2+Pd = o(n)$. Then we claim that the following parameter values satisfy constraints C3–C8:

$$\begin{aligned}
k &= 1,\\
q &= 2\left\lceil \frac{d-5}{\delta \ln(n/\beta)} \right\rceil,\\
g &= \lfloor d/q \rfloor,\\
r &= \lfloor g/3 \rfloor,\\
\tau &= r.
\end{aligned}$$

Note that constraints C3 and C5 are immediately satisfied, as is constraint C7 since $k = 1$. It is also immediate that q is even and is positive, since $d \geq 6$, $\delta > 0$, and $n/\beta > 1$; hence constraint C4 is satisfied.

For constraint C6, it is immediate that $r \leq g/3$. To show $r \geq 1$ it suffices to show $q \leq d/3$:

$$q = 2\left\lceil \frac{d-5}{\delta \ln(n/\beta)} \right\rceil \leq 2\left\lceil \frac{d-5}{6} \right\rceil = 2\left\lfloor \frac{d}{6} \right\rfloor \leq \frac{d}{3}.$$

To satisfy constraint C8 above, we first note (making frequent use of the inequalities $x/2 \leq \lfloor x \rfloor$ and $\lceil x \rceil \leq 2x$, valid for all $x \geq 1$) that

$$\begin{aligned}
g/r &= g/\lfloor g/3 \rfloor \leq g/(g/6) = 6,\\
r &= \lfloor g/3 \rfloor \leq g/3 = \lfloor d/q \rfloor/3 \leq d/(3q)\\
&= \frac{d}{6\left\lceil \frac{d-5}{\delta \ln(n/\beta)} \right\rceil} \leq \frac{1}{6}\frac{d}{d-5}\delta \ln(n/\beta) \leq \delta \ln(n/\beta),\\
r + 2 &\leq 3r,\\
d^2 + Pd &\leq d^2 + Pd \ln \frac{n}{d^2 + Pd} < \beta, \text{ and}\\
\delta &= 1/(2\ln(6e)) < 1.
\end{aligned}$$

Returning to constraint C8, we must show that (9) is at most $n/4$:

$$\begin{aligned}
&\left(\frac{eg}{r}\right)^{r(2k+\lceil \log_2 k \rceil)} (P(\tau + 2) + d)\, 6dk\\
&\quad = 6\left(\frac{eg}{r}\right)^{2r} (d^2 + Pd(r+2))\\
&\quad \leq 18(6e)^{2r}(d^2 + Pdr)\\
&\quad \leq 18(6e)^{2\delta \ln(n/\beta)}(d^2 + \delta Pd \ln(n/\beta))\\
&\quad = 18(n/\beta)(d^2 + \delta Pd \ln(n/\beta))\\
&\quad = \frac{18n(d^2 + \delta Pd \ln(n/\beta)}{72\left(d^2 + Pd \ln \frac{n}{d^2+Pd}\right)}\\
&\quad < n/4,
\end{aligned}$$

as desired.

To complete the analysis of Case 1, we show that τ is large enough to imply the bound in the statement of the theorem:

$$\begin{aligned}
\tau = r &= \lfloor g/3 \rfloor \geq g/6 = \lfloor d/q \rfloor/6 \geq d/(12q)\\
&= \frac{d}{24\left\lceil \frac{d-5}{\delta \ln(n/\beta)} \right\rceil}.
\end{aligned}$$

The latter quantity equals $d/24$, if $d \le \delta \ln(n/\beta) + 5$. If $d > \delta \ln(n/\beta) + 5$, then

$$\begin{aligned}
\frac{d}{24\left\lceil \frac{d-5}{\delta \ln(n/\beta)} \right\rceil} &\ge \frac{d}{48\left(\frac{d-5}{\delta \ln(n/\beta)}\right)} \\
&= \frac{1}{48}\frac{d}{d-5}\delta \ln(n/\beta) \\
&\ge \frac{\delta \ln(n/\beta)}{48} \\
&= \frac{\delta}{48}\ln \frac{n}{72\left(d^2 + Pd \ln \frac{n}{d^2+Pd}\right)} \\
&\ge \frac{\delta}{48}\ln \frac{n}{72\,(d^2 + Pd) \ln \frac{n}{d^2+Pd}} \\
&= \Omega\left(\ln \frac{n}{d^2+Pd}\right).
\end{aligned}$$

The penultimate inequality holds since, by assumption, $\ln(n/(d^2+Pd)) \ge 1$. The final lower bound follows since $\ln(x/(72 \ln x)) = \Omega(\ln x)$. Thus, as claimed in the statement of the theorem, $\tau = \Omega(\min(d, \ln(n/(d^2 + Pd))))$.

Case 2. Recall $\epsilon = 1/(3 \ln(6e))$ and suppose $6P \le d \le \sqrt{n}/6^9$ and $1 \le P \le \epsilon \ln(n/d^2)$. (Note that $d \le \sqrt{n}/6^9$ must be true for all sufficiently large n, since $d^2+Pd = o(n)$.) Then we claim that the following parameter values satisfy constraints C3–C8:

$$\begin{aligned}
\hat{q} &= \frac{ed}{P}\left(\frac{d^2}{n}\right)^{1/(3P)}, \\
q &= 2\lceil \hat{q}/2 \rceil, \\
g &= \lfloor d/q \rfloor, \\
r &= P, \\
k &= \left\lfloor \frac{\ln(n/d^2)}{3P \ln(ed/(qP))} \right\rfloor, \\
\tau &= (k - \lceil \log_2 k \rceil)\lfloor g/3 \rfloor = \Theta(gk).
\end{aligned}$$

Note that constraint C5 is immediately satisfied, as is constraint C7 since $r \ge P$. Since $\hat{q}$ is positive, it is also immediate that q is even and is positive, hence constraint C4 is satisfied.

Note for future use that

$$(n/d^2)^{1/(3P)} \ge (n/d^2)^{1/(3\epsilon \ln(n/d^2))} = e^{1/(3\epsilon)} = 6e. \tag{10}$$

For constraint C3, note that $q \ge \hat{q}$. Thus,

$$k = \left\lfloor \frac{\ln((n/d^2)^{1/(3P)})}{\ln(ed/(qP))} \right\rfloor \ge \left\lfloor \frac{\ln((n/d^2)^{1/(3P)})}{\ln(ed/(\hat{q}P))} \right\rfloor = \left\lfloor \frac{\ln((n/d^2)^{1/(3P)})}{\ln((n/d^2)^{1/(3P)})} \right\rfloor = 1.$$

Thus $k \ge 1$. Using a similar analysis, we note for future use that $k = 1$ whenever $q > 2$. This holds since $q > 2$ implies $\hat{q}/2 > 1$, which implies $q \le 2\hat{q}$. Thus,

$$k = \left\lfloor \frac{\ln((n/d^2)^{1/(3P)})}{\ln(ed/(qP))} \right\rfloor \le \left\lfloor \frac{\ln((n/d^2)^{1/(3P)})}{\ln(ed/(2\hat{q}P))} \right\rfloor = \left\lfloor \frac{\ln((n/d^2)^{1/(3P)})}{\ln((n/d^2)^{1/(3P)}/2)} \right\rfloor = 1. \tag{11}$$

The last equality follows from the fact that $1 < (\ln x)/(\ln(x/2)) < 2$ whenever $x > 4$, and from (10).

For constraint C6, it is immediate that $r = P \geq 1$. To show $r \leq g/3$ it suffices to show $3qP \leq d$. If $q = 2$, this holds, since by assumption $6P \leq d$. If $q > 2$, then $\hat{q}/2 > 1$, so

$$3qP = 6\lceil \hat{q}/2 \rceil P \leq 6\hat{q}P = 6(ed/P)(d^2/n)^{1/(3P)}P \leq 6ed/(6e) = d. \tag{12}$$

The last inequality follows from (10).

For constraint C8, we first note for integer $k \geq 1$ that

$$2k + \lceil \log_2 k \rceil \leq 8k/3.$$

(The bound is tight at $k = 3$.) Also,

$$\tau + 2 = (k - \lceil \log_2 k \rceil) \lfloor g/3 \rfloor + 2 \leq gk,$$

since $g \geq 3$. Using (9), we bound the number of vertices in open gadgets as follows.

$$\begin{aligned}
&\left(\frac{eg}{r}\right)^{r(2k+\lceil \log_2 k \rceil)} (P(\tau + 2) + d)\, 6dk \\
&\qquad \leq 6\left(\frac{eg}{P}\right)^{8Pk/3} dk(Pgk + d) \\
&\qquad \leq 6\left(\frac{ed}{qP}\right)^{(8/3)P\left\lfloor \frac{\ln(n/d^2)}{3P\ln(ed/(qP))} \right\rfloor} dk(Pdk/q + d) \\
&\qquad \leq 6\left(\frac{n}{d^2}\right)^{8/9} d^2(Pk^2/q + k) \\
&\qquad \leq 6n^{8/9}d^{2/9}(Pk^2/q + k).
\end{aligned}$$

We break the rest of the derivation of constraint C8 into two subcases based on d. Note that, since $3qP \leq d$ from (12),

$$k = \left\lfloor \frac{\ln(n/d^2)}{3P\ln(ed/(qP))} \right\rfloor \leq \frac{\ln n}{3P\ln(3e)} \leq \ln n.$$

When $d < n^{1/4}$, since k and P are both $O(\log n)$, we have

$$6n^{8/9}d^{2/9}(Pk^2/q + k) = O(n^{8/9}(n^{1/4})^{2/9}\log^3 n) = O(n^{17/18}\log^3 n) = o(n).$$

When $d \geq n^{1/4}$, we will show that $q > 2P$ and $k = 1$, so we have

$$6n^{8/9}d^{2/9}(Pk^2/q + k) \leq 6n^{8/9}(n^{1/2}/6^9)^{2/9}(1/2 + 1) = n/4.$$

We show that $q > 2P$ as follows.

$$\frac{q}{P} \geq \frac{\hat{q}}{P} = \frac{ed}{P^2}\left(\frac{d^2}{n}\right)^{1/(3P)} \geq \frac{n^{1/4}}{P^2}\left(\frac{n^{2/4}}{n}\right)^{1/3} = \frac{n^{1/12}}{P^2} = \omega(1).$$

(Recall $P = O(\log n)$.) Since $q > 2P$ and $P \geq 1$, we have $q > 2$, so $k = 1$ by (11).

To complete the analysis of Case 2, we show that $\tau = (k - \lceil \log_2 k \rceil) \lfloor g/3 \rfloor$ is large enough to imply the bound in the statement of the theorem. Again we split the analysis into two subcases based on d. We have $q > 2$ if and only if $\hat{q} > 2$, which holds exactly when

$$d > d_0 = (2P/e)^{3P/(3P+2)}\, n^{1/(3P+2)}.$$

In this case we have $k = 1$ by (11) and

$$\begin{aligned} \tau = \lfloor g/3 \rfloor \geq g/6 \geq d/(12q) \geq d/(24\hat{q}) &= \frac{d}{24\left(\frac{ed}{P}\left(\frac{d^2}{n}\right)^{1/(3P)}\right)} \\ &= \frac{P}{24e}\left(\frac{n}{d^2}\right)^{1/(3P)} \qquad (13) \\ &= \Omega\left(P\left(\frac{n}{d^2}\right)^{1/(3P)}\right), \end{aligned}$$

as claimed. We remark that, by (10), when P is maximal, (13) is $P/4 = \Theta(\log(n/d^2))$, so the transition to the bound given in Case 1 is "smooth."

In the second subcase we have $d \leq d_0$. First, note that

$$d_0 = (2P/e)^{3P/(3P+2)}\, n^{1/(3P+2)} \leq Pn^{1/(3P+2)} \leq Pn^{1/5} = o(n^{1/4}). \qquad (14)$$

Second, since $d \leq d_0$, we have $q = 2$ and $k \geq 1$. Also, note that $(k - \lceil \log_2 k \rceil)/k \geq 1/3$, (attaining the minimum at $k = 3$) and that $g \geq 3$. Hence $\tau = \Omega(gk)$ and

$$\begin{aligned} gk &= \left\lfloor \frac{d}{q} \right\rfloor \left\lfloor \frac{\ln(n/d^2)}{3P\ln(ed/(qP))} \right\rfloor \\ &\geq \frac{d\ln(n/d^2)}{24P\ln(ed/(2P))} \\ &= \frac{d\ln(n/o(n^{1/2}))}{24P\ln(ed/(2P))} \\ &= \Omega\left(\frac{d/P}{\ln(d/P)}\ln n\right), \end{aligned}$$

as claimed. We remark that when P is maximal and $d \geq 6P$, the estimate in (14) can be refined, allowing one to show $d = \Theta(P) = \Theta(\log n)$. Thus, τ again matches the bound in Case 1 (up to constant factors).

Finally, when $d = d_0$ we have $\hat{q}$ exactly equal to 2; similarly, when $d = d_0$, the expression of which k is the floor is precisely 1. Furthermore, both expressions vary slowly with d, so both are $\Theta(1)$ when d is near d_0. Thus, again $\tau = (k - \lceil \log_2 k \rceil) \lfloor \lfloor d/q \rfloor /3 \rfloor$ is "smooth" as d crosses d_0, the threshold between the lower bounds quoted in (a) and (b) in the statement of the theorem, and in fact both lower bounds are $\Theta(md_0)$ for d near d_0.

This completes the proof. $\square$

It is interesting to note why the proof would fail if M were allowed to jump pebbles. In the local phase, the adversary was able to pick an existing gadget in which p must invest τ steps. In the presence of jumping, this fails, since p can always jump out of the new gadget. As a particular foil to the proof above, imagine an automaton that stations one pebble p on an entry vertex of some gadget, and successively moves a second pebble q to each neighbor, jumping q back to p to find the next neighbor. In time $\Theta(d)$, this has touched all $\Theta(d)$ connecting edges incident to that entry vertex, which was impossible in the construction above.

4. Open problem. The obvious important problem is to strengthen and generalize these lower bounds. Following an earlier version of this paper [9], Edmonds [21] proved a much stronger time-space tradeoff on general JAGs: for every $z \geq 2$, a JAG with at most $\frac{1}{28z}\frac{\log n}{\log\log n}$ pebbles and at most $2^{\log^z n}$ states requires time $n \cdot 2^{\Omega((\log n)/(\log\log n))}$ to traverse 3-regular graphs. The ultimate goal might be to prove that $ST = \Omega(mn)$ for JAGs or even for general models of computation.

Acknowledgment. We thank Michael Sipser for showing us the construction generalized in section 3.

REFERENCES

[1] L. M. Adleman, *Two theorems on random polynomial time*, in 19th Annual IEEE Symposium on Foundations of Computer Science, Ann Arbor, MI, 1978, pp. 75–83.

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

[3] N. Alon, Y. Azar, and Y. Ravid, *Universal sequences for complete graphs*, Discrete Appl. Math., 27 (1990), pp. 25–28.

[4] A. Bar-Noy, A. Borodin, M. Karchmer, N. Linial, and M. Werman, *Bounds on universal sequences*, SIAM J. Comput., 18 (1989), pp. 268–277.

[5] G. Barnes, J. F. Buss, W. L. Ruzzo, and B. Schieber, *A sublinear space, polynomial time algorithm for directed s-t connectivity*, SIAM J. Comput., 27 (1998), pp. 1273–1282.

[6] G. Barnes and J. A. Edmonds, *Time-space lower bounds for directed s-t connectivity on JAG models*, in Proceedings 34th Annual IEEE Symposium on Foundations of Computer Science, Palo Alto, CA, 1993, pp. 228–237.

[7] G. Barnes and U. Feige, *Short random walks on graphs*, SIAM J. Disc. Math., 9 (1996), pp. 19–28.

[8] G. Barnes and W. L. Ruzzo, *Undirected s-t connectivity in polynomial time and sublinear space*, Comput. Complexity, 6 (1996/1997), pp. 1–28.

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

[10] P. W. Beame, A. Borodin, P. Raghavan, W. L. Ruzzo, and M. Tompa, *Time-space tradeoffs for undirected graph traversal by graph automata*, Inform. and Comput., 130 (1996), pp. 101–129.

[11] P. Berman and J. Simon, *Lower bounds on graph threading by probabilistic machines*, in 24th Annual IEEE Symposium on Foundations of Computer Science, Tucson, AZ, 1983, pp. 304–311.

[12] M. Blum and D. C. Kozen, *On the power of the compass (or, why mazes are easier to search than graphs)*, in 19th Annual IEEE Symposium on Foundations of Computer Science, Ann Arbor, MI, 1978, pp. 132–142.

[13] M. Blum and W. J. Sakoda, *On the capability of finite automata in* 2 *and* 3 *dimensional space*, in 18th Annual IEEE Symposium on Foundations of Computer Science, Providence, RI, 1977, pp. 147–161.

[14] A. Borodin, *Structured vs. general models in computational complexity*, L'Enseignement Mathématique, XXVIII (1982), pp. 171–190. Also in [27, pp. 47–65].

[15] A. Borodin, S. A. Cook, P. W. Dymond, W. L. Ruzzo, and M. Tompa, *Two applications of inductive counting for complementation problems*, SIAM J. Comput., 18 (1989), pp. 559–578. See also 18 (1989), p. 1283.

[16] A. Borodin, W. L. Ruzzo, and M. Tompa, *Lower bounds on the length of universal traversal sequences*, J. Comput. System Sci., 45 (1992), pp. 180–203.

[17] M. F. Bridgland, *Universal traversal sequences for paths and cycles*, J. Algorithms, 8 (1987), pp. 395–404.

[18] A. Z. Broder, A. R. Karlin, P. Raghavan, and E. Upfal, *Trading space for time in undirected s-t connectivity*, SIAM J. Comput., 23 (1994), pp. 324–334.

[19] J. Buss and M. Tompa, *Lower bounds on universal traversal sequences based on chains of length five*, Inform. Comput., 120 (1995), pp. 326–329.

[20] S. A. Cook and C. W. Rackoff, *Space lower bounds for maze threadability on restricted machines*, SIAM J. Comput., 9 (1980), pp. 636–652.

[21] J. A. Edmonds, *Time-space trade-offs for undirected ST-connectivity on a JAG*, in Proceedings of the 25th Annual ACM Symposium on Theory of Computing, San Diego, CA, 1993, pp. 718–727.

[22] J. A. Edmonds and C. K. Poon, *A nearly optimal time-space lower bound for directed st-connectivity on the NNJAG model*, in Proceedings of the 27th Annual ACM Symposium on Theory of Computing, Las Vegas, NV, 1995, pp. 147–156.

[23] U. Feige, *A spectrum of time-space trade-offs for undirected s-t connectivity*, J. Comput. System Sci., 54 (1997), pp. 305–316.

[24] A. Hemmerling, *Labyrinth Problems: Labyrinth-Searching Abilities of Automata*, Teubner-Texte Math. 114, Teubner, Leipzig, 1989.

[25] S. Istrail, *Polynomial universal traversing sequences for cycles are constructible*, in Proceedings of the 20th Annual ACM Symposium on Theory of Computing, Chicago, IL, 1988, pp. 491–503.

[26] H. J. Karloff, R. Paturi, and J. Simon, *Universal traversal sequences of length $n^{O(\log n)}$ for cliques*, Inform. Proc. Lett., 28 (1988), pp. 241–243.

[27] *Logic and Algorithmic*, An International Symposium Held in Honor of Ernst Specker, Zürich, Feb. 5–11, 1980, Monographie No. 30 de L'Enseignement Mathématique, Université de Genève, Switzerland, 1982.

[28] N. Nisan, $RL \subseteq SC$, Comput. Complexity, 4 (1994), pp. 1–11.

[29] N. Nisan, E. Szemerédi, and A. Wigderson, *Undirected connectivity in $O(\log^{1.5} n)$ space*, in Proceedings 33rd Annual IEEE Symposium on Foundations of Computer Science, Pittsburgh, PA, 1992, pp. 24–29.

[30] W. J. Savitch, *Relationships between nondeterministic and deterministic tape complexities*, J. Comput. System Sci., 4 (1970), pp. 177–192.

[31] W. J. Savitch, *Maze recognizing automata and nondeterministic tape complexity*, J. Comput. System Sci., 7 (1973), pp. 389–403.

[32] M. Tompa, *Two familiar transitive closure algorithms which admit no polynomial time, sublinear space implementations*, SIAM J. Comput., 11 (1982), pp. 130–137.

[33] M. Tompa, *Lower bounds on universal traversal sequences for cycles and other low degree graphs*, SIAM J. Comput., 21 (1992), pp. 1153–1160.