Lower Bounds for Testing Digraph Connectivity with One-Pass Streaming Algorithms

Theresa Migler

Abstract—In this note, we show that three graph properties—strong connectivity, acyclicity, and reachability from a vertex s to all vertices—each require a working memory of $\Omega(\epsilon m)$ on a graph with m edges to be determined correctly with probability greater than $(1+\epsilon)/2$.

Index Terms—Lower bounds, streaming graph problems, strong connectivity, acyclicity

1 Introduction

In the streaming model of computation, the input is given as a sequence, or *stream*, of elements. There is no random access to the elements; the sequence must be scanned in order. The goal is to process the stream using a small amount of working memory. For an overview see [8]. There has been much research devoted to the study of streaming algorithms, most notably the Gödel-prize winning work of Alon, Matias, and Szegedy, on the space complexity of randomized algorithms to approximate frequency moments of a sequence when the sequence is given one element at a time and cannot be stored [2].

2 Related Work

For undirected graph problems, there are many lower bounds in the edge streaming model. Sun and Woodruff provide a tight $\Omega(n \log n)$ lower bound for the following graph problems: testing bipartiteness, connectivity, cycle-freeness, whether a graph is Eulerian, planarity, H-minor freeness, finding a minimum spanning tree of a connected graph, and testing if the diameter of a sparse graph is constant. They also give an $\Omega(nk \log n)$ lower bound for: deterministic algorithms for k-edge connectivity and k-vertex connectivity [9].

Guha, McGregor, and Tench give an algorithm for k-vertex connectivity that takes $O(kn \text{ polylog } n)$ space [5].

Feigenbaum, Kannan, and Zhang show that any exact, deterministic algorithm for computing the diameter of an undirected graph in the Euclidean plane requires $\Omega(n)$ bits of working memory [4]. Zelke shows that any algorithm that is able to find a minimum cut of an undirected graph requires $\Omega(m)$ bits of working memory, this remains true even if randomization is allowed [10].

For directed graphs problems, the ones most likely to come up in analyses of the internet, much less is known. Henzinger et al. showed that for any $0 < \epsilon < 1$, estimating the size of the transitive closure of a DAG with relative expected error ϵ requires $\Omega(m)$ bits of working memory [7]. Feigenbaum, Kannan, McGregor, Suri, and Zhang showed that testing reachability from a given vertex s to another given vertex t requires $\Omega(m)$ bits of space [3]. Guruswami and Onak showed that even with p passes, the problem requires $\Omega(n^{1+1/(2(p+1))}/p^{20} \log^{3/2} n)$ bits of space to be solvable with probability at least 9/10 [6].

As for upper bounds in undirected graphs, there are one-pass algorithms for connected components, k-edge and k-vertex connectivity ($k \leq 3$), and planarity testing that use $O(n \log n)$ bits of working memory [7]. There is an algorithm that approximates the diameter within $1+\epsilon$ using $O(\frac{1}{\epsilon})$ bits [4]. For upper bounds in directed graphs, there is an algorithm that computes the exact size of the transitive closure using $O(m \log n)$ bits of working memory [7].

3 Lower Bounds

In this short note, we consider three basic connectivity questions in directed graphs: determining if a graph is strongly connected, determining if a graph is acyclic, and determining if a vertex s reaches all other vertices. A directed graph $G = (V, E)$ is said to be *strongly connected* if for every pair of vertices $u, v \in V$ there is a path from u to v and a path from v to u. A directed graph $G = (V, E)$ is said to be *acyclic* if G contains no cycles. We say that a vertex s *reaches* a vertex v if there is a directed path from s to v.

We show that, even with randomization, these graph properties each require $\Omega(m)$ bits of working memory to be decided with probability greater than $(1+\epsilon)/2$ by a one-pass streaming algorithm on n vertices and m edges. For these lower bounds we will use simple reductions from the index problem (or the bit-vector problem) in communication complexity:

> *Alice has a bit-vector x of length m. Bob has an index $i \in \{1, 2, \ldots, m\}$ and wishes to know the ith bit of x. The only communication allowed is from Alice to Bob.*

The following is a rewording of Theorem 2 from Ablayev [1].

Theorem 1. *For Bob to correctly determine x_i with probability better than $\frac{(1+\epsilon)}{2}$, $\Omega(\epsilon m)$ bits of communication are required.*

We will now state and prove our main Lemma:

Lemma 2. *Any algorithm that correctly determines the following graph properties with probability better than $\frac{(1+\epsilon)}{2}$ requires $\Omega(\epsilon m)$ bits of working memory:*

- *acyclicity*
- *strong connectivity*
- *reachability of all from s.*

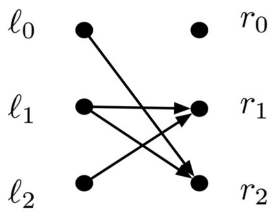
Proof. We reduce from the index problem and use Theorem 1. Let x denote the m-bit vector owned by Alice. We define the stream using two sets of edges E_1, E_2. The edge stream first has the edges of E_1 in arbitrary order, followed by the edges of E_2, also in arbitrary order. The set E_1 is entirely determined by the m-bit vector x owned by Alice, and the set E_2 is entirely determined by the index i owned by Bob. The graph defined by $E_1 \cup E_2$ has $\Omega(\sqrt{m})$ vertices, and E_1 has $O(m)$ edges. To solve the index problem, Alice constructs E_1 and simulates the streaming algorithm up to the point when E_1 has arrived, then sends to Bob the current state of the memory. Upon reception of the message, Bob constructs E_2 and continues the simulation up to the point when E_2 has finished arriving. Bob's final decision is then determined by the outcome of the streaming algorithm. Thus, the lemma will be proved.

Acyclicity. Let $n = \lceil \sqrt{m} \rceil$ and let $V = L \cup R$, where L and R both have size n and have vertices labeled 0 through $n-1$. Let E_1 be the bipartite graph that has an edge from vertex $\ell_j \in L$ to vertex $r_k \in R$ iff x has a 1 in position $jn + k$. Let E_2 consist of a single edge determined by Bob's bit i: let $k = i \bmod n$ and $j = \frac{i-k}{n}$. Then E_2 consists of the edge from vertex $r_k \in R$ to vertex $\ell_j \in L$. See Figs. 1a and 1b for an illustration of an example E_1 and E_2.

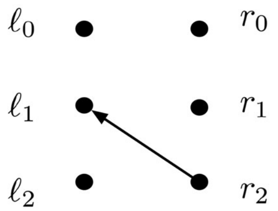
Observe that $E_1 \cup E_2$ is acyclic iff $x_i = 0$, thus the reduction is complete.

- The author is with California Polytechnic State University, San Luis Obispo, CA 93407. E-mail: Tmigler@gmail.com.

Manuscript received 12 June 2018; revised 2 July 2018; accepted 5 July 2018. Date of publication 23 July 2018; date of current version 10 Sept. 2018.
(Corresponding author: Theresa Migler.)
Recommended for acceptance by J. Hughes.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/LCOS.2018.2854761



(a) An example of E_1 corresponding to the bit-vector 001011010.

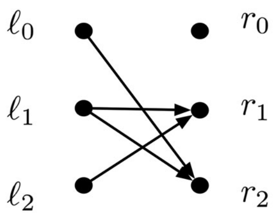


(b) An example of E_2 corresponding to B's index being 5, $j = 1$ and $k = 2$.

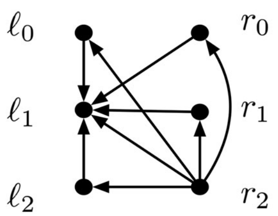
Fig. 1. An example acyclicity reduction.

Strong Connectivity. The construction for E_1 is the same as in the acyclic case. Let E_2 consist of $4n - 2$ edges determined by Bob's bit i: let $k = i \bmod n$ and $j = \frac{i-k}{n}$. Then E_2 consists of all edges from $r_k \in R$ to each vertex $V - \{r_k\}$, and from $V - \{\ell_j\}$ to $\ell_j \in L$. See Figs. 2a and 2b for an illustration of an example E_1 and E_2.

We claim that G is strongly connected iff $x_i = 1$. Indeed, if G is strongly connected, there must be a path from ℓ_j to r_k. The only edges leaving ℓ_j are to vertices in R and the only edges entering r_k are from vertices in L. And the only edges extending from R to L are either entering ℓ_j or leaving r_k. Thus, the only possible path from ℓ_j to r_k is the single edge from ℓ_j to r_k which is present only when $x_i = 1$. Now suppose that $x_i = 1$. k can certainly reach every vertex and every vertex can reach ℓ_j. Since the edge from ℓ_j to r_k is present, we know that every vertex can reach r_k and r_k can reach every vertex. Therefore, G is strongly connected. Thus the reduction is complete.

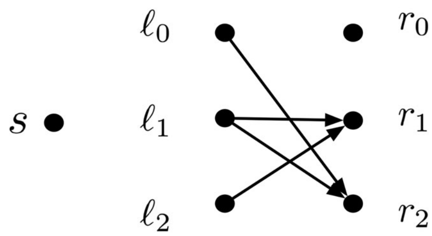


(a) An example of E_1 corresponding to the bit-vector 001011010.

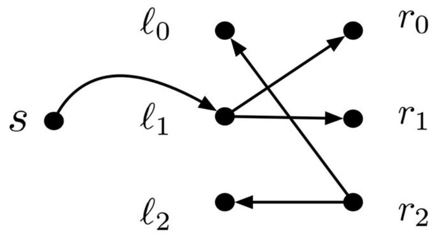


(b) An example of E_2 corresponding to B's index being 5, $j = 1$ and $k = 2$.

Fig. 2. An example strong connectivity reduction.



(a) An example of E_1 corresponding to the bit-vector 001011010.



(b) An example of E_2 corresponding to B's index being 5, $j = 1$ and $k = 2$.

Fig. 3. An example reachability reduction.

Reachability from s. Let E_1 be as above with additional vertex s with in and out degree 0.

Let E_2 consist of $2n - 1$ edges determined by Bob's bit i: let $k = i \bmod n$ and $j = \frac{i-k}{n}$. Then E_2 consists of one edge from s to $\ell_j \in L$, $n - 1$ edges from $\ell_j \in L$ to $R - \{r_k\}$, and $n - 1$ edges from $r_k \in R$ to $L - \{\ell_j\}$. See Figs. 3a and 3b for an illustration of an example E_1 and E_2.

We claim that in G s reaches everything iff $x_i = 1$. Indeed, if s can reach all vertices in G, and the only edge from s is to ℓ_j, ℓ_j must be able to reach all vertices in $G - \{s\}$. In particular ℓ_j must reach r_k. The only edges extending from R to L are from r_k, so the only way for ℓ_j to reach r_k is by the edge from ℓ_j to r_k which is present only when $x_i = 1$. Now suppose $x_i = 1$. We know s reaches ℓ_j and therefore all of R, including r_k, and r_k reaches all of $L - \{j\}$. Therefore, s reaches all vertices of G. Thus the reduction is complete. □

4 Conclusion

We have shown that any one-pass algorithm that correctly determines the acyclicity, strong connectivity, or reachability of a graph with probability better than $\frac{(1+\epsilon)}{2}$ requires $\Omega(\epsilon m)$ bits of working memory. Future work includes determining if the same result holds for an multiple-pass streaming algorithm and determining if any other graph properties exhibit a similar bound.

Acknowledgments

The author would like to express her sincere gratitude to Claire Mathieu for her support, inspiration, and guidance during this project while the author was visiting her at Brown University.

References

[1] F. Ablayev, "Lower bounds for one-way probabilistic communication complexity and their application to space complexity," *Theoretical Comput. Sci.*, vol. 157, pp. 139–159, 1996.

[2] N. Alon, Y. Matias, and M. Szegedy, “The space complexity of approximating the frequency moments,” *J. Comput. Syst. Sci.*, vol. 58, no. 1, pp. 137–147, 1999.

[3] J. Feigenbaum, S. Kannan, A. McGregor, S. Suri, and J. Zhang, “On graph problems in a semi-streaming model,” *Theoretical Comput. Sci.*, vol. 348, no. 2, pp. 207–216, Dec. 2005.

[4] J. Feigenbaum, S. Kannan, and J. Zhang, “Computing diameter in the streaming and sliding-window models,” *Algorithmica*, vol. 41, pp. 25–41, 2004.

[5] S. Guha, A. McGregor, and D. Tench, “Vertex and hyperedge connectivity in dynamic graph streams,” in *Proc. 34th ACM SIGMOD-SIGACT-SIGAI Symp. Principles Database Syst.*, 2015, pp. 241–247.

[6] V. Guruswami and K. Onak, “Superlinear lower bounds for multipass graph processing,” *Algorithmica*, vol. 76, no. 3, pp. 654–683, Nov. 2016.

[7] M. R. Henzinger, P. Raghavan, and S. Rajagopalan, “Computing on data streams,” in *Proc. External Memory Algorithms*, 1999, pp. 107–118.

[8] S. Muthukrishnan, “Data streams: Algorithms and applications,” *Found. Trends Theoretical Comput. Sci.*, vol. 1, no. 2, pp. 117–236, Aug. 2005.

[9] X. Sun and D. P. Woodruff, “Tight bounds for graph problems in insertion streams,” in *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques.*, Dagstuhl, Germany: Schloss Dagstuhl–Leibniz-Zentrum für Informatik, 2015, pp. 435–448.

[10] M. Zelke, “Intractability of min- and max-cut in streaming graphs,” *Inf. Process. Lett.*, vol. 111, no. 3, pp. 145–150, Jan. 2011.