1 Further results

1.1 Graph Problems

For all of the problems listed the model is streaming with edge arrivals and n is the number of vertices.

1. For the Maximal Matching Problem Esfandiari et al. [6] gives an algorithm that, with high probability, approximates the size of a maximum matching within a constant factor using $\tilde{O}(n^{2/3})$ space.

2. For the Weighted Maximal Matching Problem Crouch & Stubbs [4] gives a $(4+\epsilon)$ approximation algorithm which applies in semistreaming, sliding window, and MapReduce models. Chen et al. [2] studied this problem in the 1-pass model.

3. The Parameterized Vertex Cover with parameter k was studied by Chitnis et al. [3]. They proved a tight lower bound on the space of $\Omega(k^2)$ for randomized streaming algorithm.

4. Minimum Spanning Tree estimation: Given a weighted undirected graph and ϵ, find a spanning tree that has weight $\leq (1+\epsilon)$OPT where OPT is the weight of the minimal spanning tree. Assadi & N [1] proved that any algorithm that use $n^{o(1)}$ space requires $\Omega(1/\epsilon)$ passes. The result still holds if the weights are constant.

5. ϵ-Connectivity: If at least $\epsilon \cdot n$ edges need to be inserted into G to make it connected, G is said to be *ϵ-far from being connected*. Assadi & N [1] proved that any algorithm that use $n^{o(1)}$ space requires $\Omega(1/\epsilon)$ passes.

6. Cycle-freeness: If at least $\epsilon \cdot n$ edges need to be deleted from G to remove all its cycles, then G is said to be *ϵ-far from being cycle-free*. The problem is to determine if a graph is cycle-free or ϵ-far from being cycle-free. Assadi & N [1] proved that any algorithm that use $n^{o(1)}$ space requires $\Omega(1/\epsilon)$ passes.

1.2 Non-Graph Problems

1. The Longest Increasing Subsequence: Given an ordered sequence of numbers $\vec{x} = (x_1, ..., x_n)$, find an increasing subsequence that is of

maximal length. This is a streaming problem if, as the x_i's arrive, you decide if they will be in the increasing subsequence or not. Saks & Seshadhri [8] showed that, for all $\delta > 0$, a deterministic, single-pass streaming algorithm for additively approximating this problem to within an additive δn requires $O(\log^2 n/\delta)$ space. They also considered the LONGEST COMMON SUBSEQUENCE problem (Given $\vec{x}$ and $\vec{y}$ find a maximal sequence that is a subsequence of both strings.) and gave an analogous result for that one as well.

2. MAXIMUM COVERAGE: Given n, k and a set of m sets $S_i \subseteq \{1, \ldots, n\}$, find the k subsets that maximize the size of their union. There is a straightforward greedy $(1 - e^{-1})$-approximation algorithm that runs in polynomial time. McGregor & Tu [7] give two single-pass streaming algorithms and one multi-pass streaming algorithm for approximations to this problem. For the multi-pass case they also have a lower bound.

   (a) They have a single-pass streaming algorithm that for a $(1-e^{-1}-\epsilon)$-approximation that takes $\tilde{O}(\epsilon^{-2}m)$ space.

   (b) They have a single-pass streaming algorithm that for a $(1 - \epsilon)$-approximation that takes $\tilde{O}(\epsilon^{-2}m \min(k, \epsilon^{-1}))$ space.

   (c) They have an algorithm that for a $(1 - e^{-1} - \epsilon)$-approximation that takes $O(\epsilon^{-1)}$ passes and $\tilde{O}(\epsilon^{-2}k)$ space. They show that any $O(1)$ pass streaming algorithm for an $(1 - (1 - (1/k)^k) \sim 1 - \frac{1}{e}$ requires $\Omega(m)$ space.

3. BASIC COUNTING: Given a stream of bits, maintain a count of the number of 1's in the last N elements seen from the stream. Datar et al. [5] showed this problem requires $\Omega(\epsilon^{-1} \log^2 N)$ space for any randomized algorithms.

4. SUM: Given a stream of integers in $\{1, \ldots, R\}$, maintain the sum of the last N integers. Data et al. [5] showed that any streaming algorithm for this problem requires space $\Omega(\epsilon^{-1}(\log^N + \log R \log N))$. This and the previous problem relate to computing the L_P norm with an underlying vector that has a single dimension.

5. SORTING BY REVERSAL ON SIGNED PERMUTATIONS: Given a data stream of a permutation S on $\{1, \ldots, n\}$, a reversal $r(i, j)$ will transfer

$x = (x_1, \ldots, x_n)$ to $(x_1, \ldots, x_{i-1}, -x_j, \ldots, -x_i, x_{j+1}, \ldots x_n)$. Find the minimum number of reversals needed to sort S. Verbin & Yu [9] showed that this problem requires space $\Omega((n/8)^{1-1/t})$ for approximation factor $1 + 1/(4t-2)$.

References

[1] Sepehr Assadi and Vishvajeet N. Graph streaming lower bounds for parameter estimation and property testing via a streaming XOR lemma. In Samir Khuller and Virginia Vassilevska Williams, editors, *STOC '21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, pages 612–625. ACM, 2021. `https://doi.org/10.1145/3406325.3451110`.

[2] Jianer Chen, Qin Huang, Iyad Kanj, Qian Li, and Ge Xia. Streaming algorithms for graph k-matching with optimal or near-optimal update time. In Hee-Kap Ahn and Kunihiko Sadakane, editors, *32nd International Symposium on Algorithms and Computation, ISAAC 2021, December 6-8, 2021, Fukuoka, Japan*, volume 212 of *LIPIcs*, pages 48:1–48:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. `https://doi.org/10.4230/LIPIcs.ISAAC.2021.48`.

[3] Rajesh Hemant Chitnis, Graham Cormode, Mohammad Taghi Hajiaghayi, and Morteza Monemizadeh. Parameterized streaming: Maximal matching and vertex cover. In Piotr Indyk, editor, *Proceedings of the Twenty-Sixth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2015, San Diego, CA, USA, January 4-6, 2015*, pages 1234–1251. SIAM, 2015. `https://doi.org/10.1137/1.9781611973730.82`.

[4] Michael S. Crouch and Daniel M. Stubbs. Improved streaming algorithms for weighted matching, via unweighted matching. In Klaus Jansen, José D. P. Rolim, Nikhil R. Devanur, and Cristopher Moore, editors, *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM 2014, September 4-6, 2014, Barcelona, Spain*, volume 28 of *LIPIcs*, pages 96–104. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2014. `https://doi.org/10.4230/LIPIcs.APPROX-RANDOM.2014.96`.

[5] Mayur Datar, Aristides Gionis, Piotr Indyk, and Rajeev Motwani. Maintaining stream statistics over sliding windows. *SIAM J. Comput.*, 31(6):1794–1813, 2002.
https://doi.org/10.1137/S0097539701398363.

[6] Hossein Esfandiari, MohammadTaghi Hajiaghayi, Vahid Liaghat, Morteza Monemizadeh, and Krzysztof Onak. Streaming algorithms for estimating the matching size in planar graphs and beyond. *ACM Trans. Algorithms*, 14(4):48:1–48:23, 2018.
https://doi.org/10.1145/3230819.

[7] Andrew McGregor and Hoa T. Vu. Better streaming algorithms for the maximum coverage problem. *Theory Comput. Syst.*, 63(7):1595–1619, 2019.
https://doi.org/10.1007/s00224-018-9878-x.

[8] Michael E. Saks and C. Seshadhri. Space efficient streaming algorithms for the distance to monotonicity and asymmetric edit distance. In Sanjeev Khanna, editor, *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2013, New Orleans, Louisiana, USA, January 6-8, 2013*, pages 1698–1709. SIAM, 2013.
https://doi.org/10.1137/1.9781611973105.122.

[9] Elad Verbin and Wei Yu. The streaming complexity of cycle counting, sorting by reversals, and other problems. In Dana Randall, editor, *Proceedings of the Twenty-Second Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2011, San Francisco, California, USA, January 23-25, 2011*, pages 11–25. SIAM, 2011.
https://doi.org/10.1137/1.9781611973082.2.