Adversarially Robust Coloring in Graph Streams

Amit Chakrabarti   Prantar Ghosh   Manuel Stoeckl

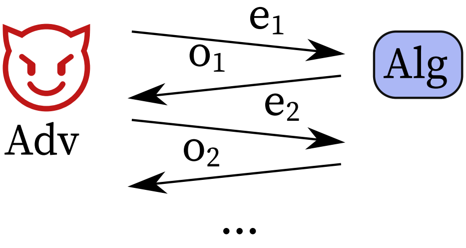
Department of Computer Science
Dartmouth College

Innovations in Theoretical Computer Science 2022

Outline

Adversarially robust streaming[1]

- A two player game between Adversary providing the stream $e_1, e_2, \ldots$, and Algorithm producing output $o_1, o_2, \ldots$
- Algorithm has limited space to remember stream inputs



- Alg is **adversarially robust** if low error probability with any adaptive adversary
- Standard streaming model is like having "oblivious adversary"
- All deterministic algorithms are also adversarially robust

[1] Adversarial setting is from [Ben-EliezerJayaramWoodruffYogev2020]; see full paper for more history

Problem: streaming graph coloring

- Stream: sequence of edges of an n vertex graph, no repetitions, forming a graph of max degree Δ
- Algorithm must output, at each point, a K-coloring of the graph.
  (As some graphs need $\Delta + 1$ colors, $K \geq \Delta + 1$)
- [AssadiChenKhanna19] give an oblivious $\Delta + 1$ coloring algorithm in $\tilde{O}(n)$ space, but adaptive adversaries can break it

Do robust streaming algorithms require significantly more space for this problem?

Lower and upper bounds

- Theorem: any robust K-coloring sketch needs $\Omega\left(\frac{n\Delta^2}{K}\right)$ space
  - Thus, $O(\Delta)$ coloring needs $\Omega(n\Delta)$ space; with $O(n)$ space, must use $\Omega(\Delta^2)$ colors
  - Compare: [AssadiChenSun21]: with $\tilde{O}(n)$ space, deterministic algos must use $\Omega\left(\exp\left(\Delta^{O(1)}\right)\right)$colors
- We find adversarially robust algorithms to:
  - $O(\Delta^2)$-color an edge insertion stream using $\tilde{O}\left(n\sqrt{\Delta}\right)$ space
    - $O(\Delta^2)$-color an edge insert/delete stream of length m using $\tilde{O}(\sqrt{nm})$ space
    - $O(\Delta^k)$-color an edge insert/delete stream of length m using $\tilde{O}\left(n^{1-1/k}m^{1/k}\right)$ space
  - $O(\Delta^3)$-color an edge insertion stream using $\tilde{O}(n)$ space, $\tilde{O}(n\Delta)$ random bits

Lower and upper bounds

- Theorem: any robust K-coloring sketch needs $\Omega\left(\frac{n\Delta^2}{K}\right)$ space
  - Thus, $O(\Delta)$ coloring needs $\Omega(n\Delta)$ space; with $O(n)$ space, must use $\Omega(\Delta^2)$ colors
  - Compare: [AssadiChenSun21]: with $\tilde{O}(n)$ space, deterministic algos must use $\Omega\left(\exp\left(\Delta^{O(1)}\right)\right)$colors
- We find adversarially robust algorithms to:
  - $O(\Delta^2)$-color an edge insertion stream using $\tilde{O}\left(n\sqrt{\Delta}\right)$ space
    - $O(\Delta^2)$-color an edge insert/delete stream of length m using $\tilde{O}(\sqrt{nm})$ space
    - $O(\Delta^k)$-color an edge insert/delete stream of length m using $\tilde{O}\left(n^{1-1/k}m^{1/k}\right)$ space
  - $O(\Delta^3)$-color an edge insertion stream using $\tilde{O}(n)$ space, $\tilde{O}(n\Delta)$ random bits

Lower bound: The subset-avoidance problem

$$\mathrm{AVOID}(t, a, b)$$

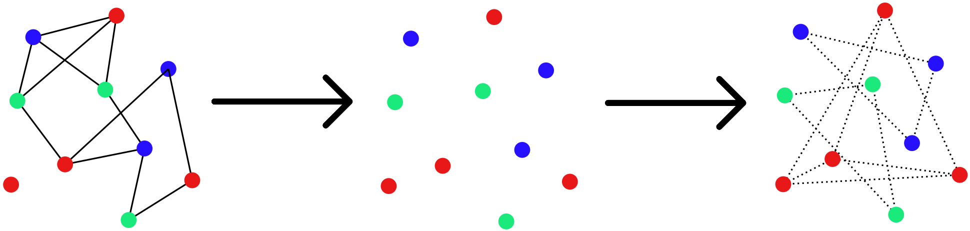
Alice has $U \subseteq [t]$ of size a, sends message to Bob

Bob wants to learn any $V \subseteq [t]$ of size b disjoint from U

Lemma: $R^{\rightarrow}_{1/3}\left(AVOID\left(t, a, b\right)\right) = \Omega\left(ab/t\right) - O\left(1\right)$

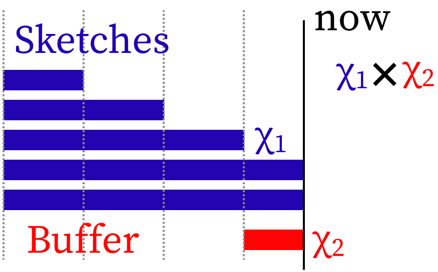
Lower bound: Reducing graph coloring to AVOID

- Given an adversarially robust sketch, can find many non-edges



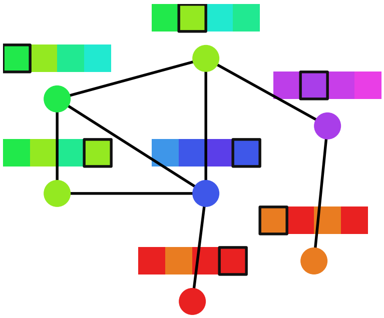
- Because sketch works even with adaptive adversary, can add some revealed edges back in, and find even more non-edges
- Can send the sketch as part of a protocol for multiple copies of AVOID
- Lower bound on AVOID message size implies the desired lower bound on sketch size

Upper bound: Robust $O(\Delta^2)$-coloring with $\tilde{O}\left(n\sqrt{\Delta}\right)$ space



- Algorithm maintains $\sqrt{\Delta}$ copies of [ACK19] sketch (which is *not* adversarially robust), and a buffer of incoming edges that is emptied every $n\sqrt{\Delta}$ edges
- Every time the buffer is emptied, freeze another sketch
- The product $\chi_1 \times \chi_2$ of the colorings χ_1 from the last frozen sketch and χ_2 from the buffer is a valid coloring for the entire stream up to this point

Upper bound: Robust $O(\Delta^3)$-coloring with $\tilde{O}(n)$ space and $\tilde{O}(n\Delta)$ random bits



- Every vertex gets a random palette of $O(\log n)$ colors *for each possible value of its degree*
- When you add an edge, recolor vertices using palette
- Store all edges which *could* produce a conflict in the future; palettes are small, so there aren't many
- Adversarially robust, because current colors reveal *nothing* about how future colors are decided, making adversary oblivious

Conclusions

- $\Omega\left(n\Delta^2/K\right)$ lower bound for robust algorithms to K-color an edge insertion stream of max degree Δ
- $O\left(\Delta^3\right)$ robust coloring algorithm using $\tilde{O}(n)$ space, $\tilde{O}(n\Delta)$ random bits
- $O\left(\Delta^2\right)$ robust coloring algorithm using $\tilde{O}\left(n\sqrt{\Delta}\right)$ space, and variations for ins/del streams and color-space tradeoffs (see full paper)

Open problems

- Is the lower bound tight? With $\tilde{O}(n)$-space, it predicts $\Omega\left(\Delta^2\right)$ colors needed; best we did is $O\left(\Delta^3\right)$ colors
- Efficient robust coloring algorithms for long ins/del streams
- Find oblivious vs adversarial separation for natural *estimation* problem, like approximating F_2 over turnstile streams

Conclusions

- $\Omega\left(n\Delta^2/K\right)$ lower bound for robust algorithms to K-color an edge insertion stream of max degree Δ
- $O\left(\Delta^3\right)$ robust coloring algorithm using $\tilde{O}\left(n\right)$ space, $\tilde{O}\left(n\Delta\right)$ random bits
- $O\left(\Delta^2\right)$ robust coloring algorithm using $\tilde{O}\left(n\sqrt{\Delta}\right)$ space, and variations for ins/del streams and color-space tradeoffs (see full paper)

Open problems

- Is the lower bound tight? With $\tilde{O}\left(n\right)$-space, it predicts $\Omega\left(\Delta^2\right)$ colors needed; best we did is $O\left(\Delta^3\right)$ colors
- Efficient robust coloring algorithms for long ins/del streams
- Find oblivious vs adversarial separation for natural *estimation* problem, like approximating F_2 over turnstile streams