COMP 7720 - Online Algorithms

Online Graph Problems

Shahin Kamali

Lecture 22 - Nov. 27th, 2018

University of Manitoba
Review & Plan
Today’s plan

- Online Edge-coloring
- Online bipartite matching (marriage problem)
- Assignment 4 & logistics
Online Edge Coloring in Graphs
Problem Definition

- In edge coloring, the goal is to color edges of a graph with minimum number of colors
  - No two adjacent edges (edges sharing an endpoint) should have the same color
- In the offline setting, the problem is NP-hard!
- For a graph of max-degree $\Delta$, at least $\Delta$ and at most $\Delta + 1$ colors are required (Vizing theorem)
  - This implies that $\text{cost}(\text{OPT}) \approx \Delta$
Problem Definition

- In the online setting, edges arrive one by one, and an algorithm should take an irrevocable decision on coloring the edges without any knowledge about future edges (or how graph looks).
- For example, Greedy family of algorithms maintain a set of colors and use them, if possible, before requesting a new coloring.
- Cost of $\text{OPT}$ is 3
- Cost of Greedy is 4, which is not optimal
Theorem

Greedy has a competitive ratio of at most 2.

For any graph of degree $\Delta$, cost of $\text{OPT}$ is at least $\Delta$.

Cost of greedy is at most $2\Delta - 1$.

- Consider the edge that demands the last color.
  - It is an edge between two vertices, each currently adjacent to at most $\Delta - 1$ edges.
  - The number of colors will be $2(\Delta - 1) + 1 = 2\Delta - 1$
Lower Bound

**Theorem**

*No deterministic online algorithm can have a competitive ratio better than 2.*

- Adversary forms a graph of degree $\Delta$ which can be colored using $\Delta$ colors
  - In doing so, any online algorithm needs to use at least $2\Delta$ colors
Lower Bound

- The input starts by sending star of degree $\Delta - 1$
  - Recall that a star of degree $d$ is a tree formed by a center vertex connected to $d$ leaves.
- There are at most $K = \Delta \binom{2\Delta}{\Delta}$ ways to color a star using $2\Delta$ colors.
  - If we have $K + 1$ stars, at least two of them have the same coloring (pigeonhole principle).
  - If we have $2K + 1$ stars, at least three of them have the same coloring.
  - If we have $(\Delta + 1)K + 1$ stars, at least $\Delta$ of them will have the same coloring.
Lower Bound

- After sending \((\Delta + 1)K + 1\) stars, at least \(\Delta\) stars have the same coloring.

- Adversary reveals edges forming another star, of degree \(\Delta\), connected to centers of these stars.
  - Any of these new edges requires a color other than the \(\Delta - 1\) colors in the old star.
  - In total \((\Delta - 1) + \Delta = 2\Delta - 1\) colors are used by the algorithm.
Lower Bound

- **OPT** colors edges adjacent to the new center (at the bottom) using $\Delta$ colors.
- Other edges form stars connected to only one of the colored edges; each star can be colored using the remaining $\Delta - 1$ colors.
- **OPT** uses $\Delta$ colors.
For any given online algorithm, adversary created a graph $G$ so that:

- $G$ has degree $\Delta$ → $OPT$ uses $\Delta$ colors.
- the online algorithm uses $2\Delta - 1$ colors

Competitive ratio of the algorithm is at least $\frac{2\Delta - 1}{\Delta} \approx 2$
Lower Bound

**Theorem**

No deterministic online algorithm can have a competitive ratio better than 2.

- This implies that greedy algorithms are the best deterministic algorithm
Online Bipartite Matching
(Marriage problem)
Online Bipartite Matching

- Given a bipartite graph, the goal is to create a matching (creating pair of non-adjacent edges) with maximum size.
- In the online setting, vertices in one side of the graph are given, and vertices in the other side arrive one by one.
  - Upon arrival of vertex $x$, all edges connecting $x$ to its neighbors on the left are revealed.
  - An online algorithm should match $x$ with another vertex, if possible, without any information about future vertices.
- **Greedy algorithm:** match with any vertex on the right if possible!
Competitive ratio for max. problems

- In this example, $\text{OPT}$ has a **benefit** of 4 (a perfect matching) while greedy has a benefit of 3!

- Matching is a maximization problem: we would like to maximize the benefit instead of minimizing the cost.

- Competitive ratio is often defined as the maximum value of \[ \frac{\text{Benefit}(\text{OPT})}{\text{benefit}(\text{Alg})} \]

- Greedy algorithm always creates a maximal matching
  - For any ‘mistake’ match, it blocks two possible matches
  - The benefit of greedy is no less than twice that of $\text{OPT}$

**Theorem**

*Greedy has a competitive ratio of at most 2 (details in the next class).*