

7

Martingales

7.1 Online Edge Coloring (Online Fair Matchings)

In the online edge coloring problem, we are given a max degree Δ and a sequence of edges $e_1, e_2, \dots, e_m$ on a vertex set V with maximum degree Δ . The edges arrive one by one and we must color each edge as it arrives with a color such that no two adjacent edges share the same color. The goal is to minimize the number of colors used. There has been recent progress on this problem by Blikstad, Svensson, Vintan, and Wajc that settles a conjecture of Bar-Noy, Motwani, and Naor from 1992. They basically show the following:

Theorem 7.1. *There is a randomized online algorithm that uses at most $(1 + o(1))\Delta$ colors in expectation whenever $\Delta = \omega(\sqrt{\log n})$. Moreover, there is a deterministic online algorithm that uses at most $(1 + o(1))\Delta$ colors whenever $\Delta = \omega(\log n)$.*

Both the randomized (and perhaps surprisingly the deterministic) algorithm are based on analyzing Martingales, which lead to rather clean and tight algorithms. We explain this connection in a simplified (but very related) setting of online fair matchings. Specifically, given the max degree Δ , we wish to devise a randomized algorithm that maintains a matching M in an online fashion (whenever an edge is presented we need to irrevocably decide whether to include it in our matching or not) such that the following holds:

$$\Pr[e \in M] \geq \frac{1}{\Delta + q} \quad \forall e \in E,$$

where q is a small "error" term that is $o(\Delta)$. We can notice that if we have an online edge coloring algorithm that uses at most $\Delta + q$ colors, then it is easy to achieve a fair matching algorithm. Simply output one of the color classes uniformly at random. This shows that the online edge coloring problem is at least as hard as the online fair matching problem. Surprisingly, one can also (when randomization

is allowed) go the other direction (up to losing some small factors $o(\log n)$ etc.). In any case, we will focus on the online fair matching problem and illustrate the main ideas there.

7.1.1 The Algorithm

After thinking about this for a while, there is almost only one reasonable algorithm that comes to mind. At each time step t , we will for each pair of vertices (i.e., potential edge e) maintain a *bias* $Q_t(e)$ that we will use to sample the edge e if it arrives at time t . We will initialize $Q_1(e) = 1/(\Delta + q)$ for all edges e . Indeed, why would we ever want to sample an edge with probability larger than $1/(\Delta + q)$? That would only be detrimental, as it would mean we are over-sampling some edges and decreasing the probability that we can take later arriving edges. These biases will change over time, as we see now.

1. When an edge $e_t = (u_t, v_t)$ arrives at time t , we sample it with probability equal to its current bias $Q_t(e_t)$.
2. If we sample e_t , we add it to our matching M and then zero out the bias for all edges e incident to either u_t or v_t , i.e., set $Q_{t+1}(e) = 0$ for all these edges. Indeed, we cannot select any of these neighboring edges in a matching.
3. On the other hand, if we do not sample e_t , we will keep the matching unchanged and update the neighbors' biases as follows:

$$Q_{t+1}(e) := \frac{Q_t(e)}{1 - Q_t(e_t)} \quad \forall e \text{ incident to } u_t \text{ or } v_t.$$

The reason for this update is simple: that we want to “boost” the biases of the edges incident to u_t or v_t since we did not get to sample e_t . Indeed, for an edge e incident to u_t or v_t we have that the probability that we sample that edge if it were to arrive at time $t + 1$ to be

$$\mathbb{E}[Q_{t+1}(e) \mid Q_t(e)] = Q_t(e_t) \cdot 0 + (1 - Q_t(e_t)) \frac{Q_t(e)}{1 - Q_t(e_t)} = Q_t(e),$$

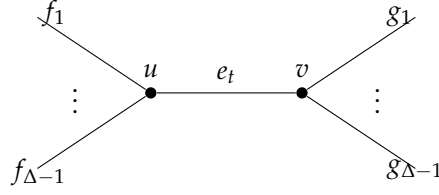
so these biases give us a Martingale!

In other words, that each edge e has the same probability of being included in the matching if it were to arrive at time t or $t + 1$, or more generally at any time step. Specifically, each edge e is sampled in the matching with probability exactly $1/(\Delta + q)$.

7.1.2 The Issue

However, an issue with the above “algorithm”: it is only well-defined if we can ensure that $Q_t(e_t) \leq 1$ for all t and e_t .

To get a feeling for this condition let us consider the following example:



Here we think that the edges arrive in the order

$$f_1, f_2, \dots, f_{\Delta-1}, g_1, g_2, \dots, g_{\Delta-1}, e_t.$$

The only chance that we have to pick e_t in the matching is that none of the other edges are selected in the matching. If this is the whole graph, we have that

$$Q_t(e_t) = Q_1(e_t) \cdot \frac{1}{\prod_{i=1}^{\Delta-1} (1 - Q_i(f_i))} \cdot \frac{1}{\prod_{j=1}^{\Delta-1} (1 - Q_{\Delta-1+j}(g_j))},$$

which simplifies to

$$Q_t(e_t) = \frac{1}{\Delta + q} \cdot \frac{1}{S_t(u)} \cdot \frac{1}{S_t(v)},$$

where

$$S_t(u) = 1 - \sum_{i=1}^{\Delta-1} Q_1(f_i) \quad \text{and} \quad S_t(v) = 1 - \sum_{j=1}^{\Delta-1} Q_1(g_j).$$

Calculations aside, this is rather natural as this is the probability that u is free if we match every f_i with probability $Q_1(f_i)$ and similarly for v . Crucially, note that since each $S_t(u)$ is a linear function of the edge biases, it itself is a Martingale.

We thus have that both $S_t(u)$ and $S_t(v)$ are at least $1 - \frac{\Delta-1}{\Delta+q} \geq q/(2\Delta)$, where we used that $q \leq \Delta$. This gives us that

$$Q_t(e_t) = \frac{1}{(\Delta + q)} \frac{1}{S_t(u)} \frac{1}{S_t(v)} \leq 4 \frac{\Delta}{q^2} \leq 1/\sqrt{\Delta}$$

if we set $q = 4\Delta^{3/4}$. So this case is great in that $P(e_t)$ is not only smaller than 1 but actually very small (less than $1/\sqrt{\Delta}$).

However, what if there are edges that arrive before the edges in the figure, incident to the edges f_i and g_i ? This may, if not taken, increase the scaling factors $1/S_t(u)$ and $1/S_t(v)$. Moreover, it is very hard to get a control on these $P(\cdot)$ values as they may be arbitrarily correlated.

Indeed, observe that $Q_2(f_2) = \frac{Q_1(f_2)}{1 - Q_1(f_1)}$ since f_1 was not chosen, and hence $1 - Q_2(f_2) = \frac{1 - Q_1(f_1) - Q_1(f_2)}{1 - Q_1(f_1)}$, etc.

7.1.3 Concentration Save The Day

Martingale concentration allows us to handle this issue and only consider the local 2-hop neighborhood of the edge e_t . Specifically, the scaling factors $S_t(u)$ form a Martingale for each u . So even if $Q_1(f_i)$ was not the original bias of the neighboring edge f_i , but the bias due to all the edges that came before, in expectation it equals $1/(\Delta + q)$.

We can now use the known Martingale concentration inequalities to show that the scaling factors $S_t(u)$ do not deviate too much from its initial value (which is at least $q/(2\Delta)$) over all arriving edges incident to f_i 's. A union bound then shows that $P(e_t) \leq O(\frac{1}{\sqrt{\Delta}})$ with high probability.

In the formal proof, we need to be a bit more careful to bound the step size of the Martingale: this is done by never selecting edges with probability much larger than $1/\sqrt{\Delta}$ (if an edge arrives with probability larger than this, we simply reject it). This then allows us to use Freedman's inequality to conclude that the Martingale does not deviate too much from its initial value.