Princeton Univ. F'24 COS 597b: Recent Advances in Graph Algorithms

Lecture 16: Dynamic Algorithms for Maximum Matching Size

Lecturer: *Pachara Sawettamalya, Elena Gribelyuk, Akhil Jakatdar* Scribe: *Zhijun Zhang*

1 Approximate Maximum Matching Size

Given a graph $G = (V, E)$ on n vertices, the goal is to design efficient algorithms that can compute good approximations to (the size of) maximum matchings. Relevant works are in two different settings: (1) the sublinear setting where the graph is accessed via queries to its adjacency matrix/list; (2) the dynamic setting where the graph is updated by a sequence of edge insertions and deletions. Previous works and new results are summarized in Table 1.

Table 1: Previous works and new results

Reference	Approximation ratio	(Update) time	Setting	Output
[BGS18]	$1/2$	$O(\text{polylog}(n))$	Dynamic	Edges
[BLM20]	$1/2 + \Omega_\epsilon(1)$	$O(n^\epsilon)$	Dynamic	Edges
[Beh21]	$1/2 - \epsilon$	$\tilde{O}(n/\epsilon^2)$	Sublinear	Size
[BRRS23]	$1/2 + \Omega_\epsilon(1)$	$O(n^{1+\epsilon})$	Sublinear	Size
[Beh23]	$1/2 + \Omega(1)$	$O(\text{polylog}(n))$	Dynamic	Size
[Beh23]	$2/3 + \Omega(1)$	$O(\sqrt{n})$	Dynamic, Bipartite	Size

2 Warmup: Lazy Approach

Throughout, we assume $\mu(G) = \Theta(n)$. This is without loss of generality using known techniques. The basic idea is to transform sublinear algorithms into dynamic algorithms. Suppose there is a sublinear algorithm A for α-approximate maximum matching size in T time. It implies a dynamic algorithm A' for $(\alpha - \epsilon)$-approximate maximum matching size in $O(T/(\epsilon n))$ update time (amortized, but can be de-amortized to be worst-case) as follows.

1. Run A to approximate $\mu(G)$.
2. Do nothing for ϵn steps, and then repeat the first step.

Observe that we do need sublinear algorithms because $T = \Theta(n^2)$ only yields $\Theta(n/\epsilon)$ update time. Using this lazy approach, [Beh21] implies an (almost) 1/2-approximation in $O(\text{polylog}(n))$ update time, and [BRRS23] recovers the result of [BLM20] (but only for the size not the edges). It also means that for our purpose, it is sufficient to design a sublinear algorithm for $(1/2 + \Omega(1))$-approximate maximum matching size in $\tilde{O}(n)$ time.

Lemma 1 (Lemma 4.1 of [Beh23])**.** *If there is a (randomized) semi-dynamic algorithm A with update time $U(n)$ and query time $Q(n, \epsilon)$ that outputs $\tilde{\mu}$ satisfying $\alpha\mu(G) - \epsilon n \leq$*

$\mathbb{E}[\tilde{\mu}] \leq \mu(G)$, then there is a (randomized) fully-dynamic algorithm B with update time $O((U(n) + Q(n, \epsilon^2)/n) \cdot \text{poly}(\log n, 1/\epsilon))$ that w.h.p. outputs $\tilde{\mu}'$ satisfying $(\alpha - \epsilon)\mu(G) \leq \tilde{\mu}' \leq \mu(G)$.

3 Overview of [Beh21]

Given a permutation π over the edge set E of a graph $G = (V, E)$, let $\mathbf{GMM}(G, \pi)$ denote the greedy maximal matching obtained by greedily adding edges of G in the order of E whenever possible. A crucial component used in the proof is a sublinear algorithm of [Beh21] for approximating $\mathbf{GMM}(G, \pi)$ using query access to the adjacency matrix of G.

Proposition 2 (Proposition 4.4 of [Beh23])**.** *There exists a randomized algorithm with query access to adjacency matrix that w.h.p. outputs $\tilde{g}$ satisfying*

$$\mathbb{E}_\pi[|\mathbf{GMM}(G, \pi)|] - \epsilon n \leq \tilde{g} \leq \mathbb{E}_\pi[|\mathbf{GMM}(G, \pi)|]$$

in $\tilde{O}(n/\epsilon^3)$ time.

At a very high level, the algorithm of [Beh21] is roughly as follows. Let M be a matching. Suppose we have oracle access that given $v \in V$, returns whether v is matched by M using Q queries. Then we can sample T vertices and output $\tilde{g} = (\#\text{matched vertices}/T) \cdot (n/2)$. Setting $T = \Theta(1/\epsilon^2)$ approximates the size of M with ϵn additive error. The running time is QT. So the task is basically minimizing Q for a random vertex, which depends on M. It turns out that it is easy to check whether v is matched for a random greedy maximal matching.

4 Bipartite Graphs & Oblivious Adversaries

We first present a semi-dynamic algorithm with update time $U(n) = O(\text{polylog}(n))$, query time $Q(n, \epsilon) = \tilde{O}(n/\epsilon^3)$, and approximation parameter $\alpha = 2 - \sqrt{2}$, in the case of bipartite graphs and oblivious adversaries. Plugging into Lemma 1, we can get a $(2 - \sqrt{2}) \approx 0.585$-approximate dynamic algorithm. The starting point is the following query algorithm inspired by a random-order streaming algorithm of [KMM12].

1. Let M be a maximal matching that we maintain.
2. Let $M' \subseteq M$ include each matching edge independently with probability p.
3. Let $V' = V(M')$ and $U = V \setminus V(M)$.
4. Let $H = G[V', U]$.
5. Let $g = \mathbb{E}_\pi[|\mathbf{GMM}(H, \pi)|]$.
6. Return $\tilde{\mu} = |M| + \max(0, g - |M'|)$.

The update time is $O(\text{polylog}(n))$ for maintaining M using [BGS18]. To see the query time, the computation of M', V', U can all be done in $O(n)$ time given M. Although explicitly maintaining H is too time-consuming, g can be approximated up to ϵn additive error in $\tilde{O}(n/\epsilon^3)$ time by Proposition 2, which is tolerable by Lemma 1. So it remains to show $\tilde{\mu}$ is a $(2-\sqrt{2})$-approximation of $\mu(G)$ in expectation. To this end, we need the following result from [KMM12].

Proposition 3. *Let $0 < p \leq 1$. Let $G = (A \sqcup B, E)$ be a bipartite graph. Let $A' \subseteq A$ include each vertex independently with probability p. Let H be the induced bipartite subgraph on A', B. Then, for any permutation π over E, it holds that*

$$\mathbb{E}_{A'}[|\mathbf{GMM}(H,\pi)|] \geq p/(1+p) \cdot \mu(G).$$

Proof of approximation ratio. We first show the lower bound. Define $F_L = G[V(M) \cap L, U \cap R]$, $F_R = G[V(M) \cap R, U \cap L]$, $H_L = G[V(M') \cap L, U \cap R]$, and $H_R = G[V(M') \cap R, U \cap L]$. Fix a maximum matching M^* of G. Consider the symmetric difference $M^* \oplus M$. There are exactly $|M^*| - |M| = \mu(G) - |M|$ disjoint augmenting paths with respect to M by the optimality of M^*. Furthermore, every augmenting path has length at least 3 since M is a maximal matching (length-1 augmenting path is simply an isolated edge). Any such augmenting path must starts with an edge in $F_L \cap M^*$ and ends with an edge in $F_R \cap M^*$ (or conversely). As all edges of M^* are disjoint, we can get $\mu(F_L) \geq \mu(G) - |M|$ and $\mu(F_R) \geq \mu(G) - |M|$. Applying Proposition 3, we have that for any π,

$$\mathbb{E}_{M'}[|\mathbf{GMM}(H_L,\pi)|] \geq p/(1+p) \cdot \mu(F_L) \geq p/(1+p) \cdot (\mu(G) - |M|),$$

and

$$\mathbb{E}_{M'}[|\mathbf{GMM}(H_R,\pi)|] \geq p/(1+p) \cdot \mu(F_R) \geq p/(1+p) \cdot (\mu(G) - |M|).$$

Since H_L and H_R are disjoint, for $H = H_L \sqcup H_R$, we can get

$$\mathbb{E}_{M'}[|\mathbf{GMM}(H,\pi)|] \geq 2p/(1+p) \cdot (\mu(G) - |M|).$$

Finally, taking expectation over $\tilde{\mu}$, we have

$$\begin{aligned}
\mathbb{E}_{M'}[\tilde{\mu}] &= \mathbb{E}_{M'}[|M| + \max(0, g - |M'|)] \\
&\geq |M| + \max(0, \mathbb{E}_{M'}[g - |M'|]) \\
&\geq |M| + \max(0, 2p/(1+p) \cdot (\mu(G) - |M|) - p|M|) \\
&\geq (1 - 2p/(1+p) - p)|M| + 2p/(1+p) \cdot \mu(G).
\end{aligned}$$

The lower bound follows by setting $p = \sqrt{2} - 1$.

Upper bound: to be continued. □

References

[Beh21] Soheil Behnezhad. Time-optimal sublinear algorithms for matching and vertex cover. In *62nd IEEE Annual Symposium on Foundations of Computer Science, FOCS 2021, Denver, CO, USA, February 7-10, 2022*, pages 873–884, 2021.

[Beh23] Soheil Behnezhad. Dynamic algorithms for maximum matching size. In *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 129–162, 2023.

[BGS18] Surender Baswana, Manoj Gupta, and Sandeep Sen. Fully dynamic maximal matching in o(log n) update time. *SIAM J. Comput.*, 47(3):617–650, 2018.

[BLM20] Soheil Behnezhad, Jakub Lacki, and Vahab S. Mirrokni. Fully dynamic matching: Beating 2-approximation in δ^{ϵ} update time. In *Proceedings of the 2020 ACM-SIAM Symposium on Discrete Algorithms, SODA 2020, Salt Lake City, UT, USA, January 5-8, 2020*, pages 2492–2508, 2020.

[BRRS23] Soheil Behnezhad, Mohammad Roghani, Aviad Rubinstein, and Amin Saberi. Beating greedy matching in sublinear time. In *Proceedings of the 2023 ACM-SIAM Symposium on Discrete Algorithms, SODA 2023, Florence, Italy, January 22-25, 2023*, pages 3900–3945, 2023.

[KMM12] Christian Konrad, Frédéric Magniez, and Claire Mathieu. Maximum matching in semi-streaming with few passes. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques - 15th International Workshop, APPROX 2012, and 16th International Workshop, RANDOM 2012, Cambridge, MA, USA, August 15-17, 2012. Proceedings*, volume 7408 of *Lecture Notes in Computer Science*, pages 231–242, 2012.