CSCE 689: Special Topics in Modern Algorithms for Data Science Fall 2023

Lecture 20 — October 18, 2023

Prof. Samson Zhou *Scribe: Anmol Anand*

1 Matching in Graphs

A matching in an undirected graph is a set of edges without common vertices between any two pairs of edges in the matching.

1.1 Alternating Path

An alternating path is any path of edges that alternates between edges in and not in the matching.

1.2 Augmenting Path

An augmenting path is any alternating path of edges that does not start and does not end at a vertex in the matching.

1.3 Maximal Matching

A matching is maximal if the addition of any other edge in the graph would no longer induce a matching.

Consider the following greedy algorithm to find a maximal matching:

Algorithm 1: Greedy algorithm to find a maximal matching M' in a graph $G = (V, E)$

$M' \leftarrow \phi$
$\forall (u, v) \in E$: add (u, v) to M' if none of the edges in M' have u or v as endpoints
return M'

1.4 Maximum Matching

A maximum matching is a matching of largest size.

1.5 Properties

(a) Each edge of a maximal matching M' can be incident to at most 2 edges of a maximum matching M^*.

(b) Each edge of a maximum matching M^* must be incident to some edge of any maximal matching M'.

(c) Each maximal matching M' is a 2-approximation to the maximum matching M^*:

$$|M'| \leq |M^*| \leq 2|M'|$$

<u>Charging argument</u>: Can each edge in M^* get a dollar from an incident edge in a given M' where each edge in M' has not more than 2 dollar?

Since each edge in M^* is incident on at least one edge in M' (from (b)) and each edge in M' is incident on not more than 2 edges in M^* (from (a)), therefore, each edge in M^* should be able to get a dollar from an incident edge in M'.

The charging argument is an intuition on why M' is a 2-approximation to M^*:

$$\begin{aligned}
2|M'| &= \sum_{e' \in M'} 2|e'| \\
&\geq \sum_{e' \in M'} [|N_1(e')| + |N_2(e')|] \\
&\geq \sum_{e \in M^*} |e| \\
&= |M^*|
\end{aligned}$$

1.6 Weighted Matching

For a weighted graph G, the weight of a matching M is the sum of the weights of the edges in M.

Algorithm 2: Greedy algorithm to find a maximal weighted matching M' in a weighted graph $G = (V, E)$

Let N be the maximum weight of an edge in G.
For $i = 0, 1, ..., \log_{1+\varepsilon} N$, let S_i be the substream that contains edges with weights between $(1+\varepsilon)^i$ and $(1+\varepsilon)^{i+1}$
Let M_i' be a maximal matching obtained by using greedy algorithm on S_i
Let M' be obtained by greedily adding edges from M_i' for $i = \log_{1+\varepsilon} N, ..., 1, 0$.

<u>Intuition</u>: Each edge e of matching M' can "block" at most two edges of M_i', each of these two edges can "block" at most two edges in the best matching M^*.

Algorithm 2. is a $(4+\varepsilon)$-approximation to the maximum weighted matching in the semi-streaming model [1].

There exists a $(2+\varepsilon)$-approximation to the maximum weighted matching in the semi-streaming model [3].

More generally, there exists a $3+2\sqrt{2} = 5.828$-approximation to the maximum *submodular* matching in the semi-streaming model [2].

1.7 Approximation and Complexity

Greedy algorithm to find maximal matching that uses $O(n)$ space is a 2-approximation to the maximum matching.

It is still an open question if it is possible to achieve a C-approximation to the maximum (cardinality) matching using $n \cdot \text{polylog}(n)$ space for $C < 2$ in the semi-streaming model.

2 Connectivity in Graphs

A graph $G = (V, E)$ is connected if there exists a path between i and j for any pair of vertices $i, j \in V = [n]$.

2.1 Applications

- Transportation networks: Analyzing the connectivity of transportation networks, e.g., roads, railways, flight routes, is critical for optimizing routes, planning public transportation, identifying congested areas, and ensuring efficient travel.

- Electrical power grids: Determining the connectivity of an electric power grid is essential for ensuring a reliable and resilient power supply. Identifying isolated components helps in quickly restoring power after outages.

2.2 Spanning Tree

A subgraph of G that is a tree and contains all the vertices of the graph G.

Observation: A graph G is connected if and only if G has a spanning tree.

2.3 Spanning Forest

A subgraph of G that is a union of trees that contains all the vertices of the graph G.

2.4 Minimum spanning tree algorithms in the offline setting

- Kruskal: Greedily add minimum weight edge to spanning forest
- Prim: Greedily grow minimum spanning tree

2.5 Greedy algorithm for connectivity in an online setting

Algorithm 3: Greedily add edges to minimum spanning forest

```
Initialize F ← ϕ
For each edge e = (u, v):
        If F ∪ (u, v) does not contain a cycle, add (u, v) to F : F ← F ∪ (u, v)
        If |F| = n − 1, return GRAPH IS CONNECTED
Return GRAPH IS NOT CONNECTED
```

Space complexity: Algorithm can keep at most n edges, so the total space usage is $O(n)$ words of space.

References

[1] Michael S. Crouch and Daniel M. Stubbs. Improved streaming algorithms for weighted matching, via unweighted matching. In *Approximation, Randomization, and Combinatorial Optimization. Algorithms and Techniques, APPROX/RANDOM*, pages 96–104, 2014.

[2] Roie Levin and David Wajc. Streaming submodular matching meets the primal-dual method. In *Proceedings of the 2021 ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 1914–1933, 2021.

[3] Ami Paz and Gregory Schwartzman. A $(2 + \varepsilon)$-approximation for maximum weight matching in the semi-streaming model. In Philip N. Klein, editor, *Proceedings of the Twenty-Eighth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA*, pages 2153–2161, 2017.