# A Greedy-Based Approximation Algorithm for Maximum Matching

Special Mathematics Lecture (Graph Theory)
Nagoya University, Spring 2024

YANG Ming-Tian
The University of Hong Kong

20th July 2024

## 1 Introduction

A matching and a maximum matching of a graph can be defined as follows [3]:

**Definition 1** (Matching)**.** A matching in a graph $G = (V, E)$ is a set $F \subseteq E$ such that no two edges in $F$ have a common end point.

**Definition 2** (Maximum matching)**.** A maximum matching of a graph $G = (V, E)$ is a matching which contains the greatest possible number of edges.

In class, we solved the maximum bipartite matching problem by transforming it into a maximal flow problem [3]. In fact, for non-bipartite graphs, there are also many polynomial time algorithms.

However, this report will not discuss them. Instead, we will introduce a greedy algorithm, whose solution may not be the optimum, but through some algorithm analysis techniques, we can prove that it can approximate the optimal solution by a certain ratio. Moreover, it has a much better time complexity and thus is somehow more pragmatic in real-world applications.

This algorithm is based on my solution to Exercise 35-4 in [1].

## 2 Algorithm Design & Analysis

Initially, let the set $F$ be empty. We enumerate each edge $e \in E$ in an arbitrary order: if $e$ has no common end points with any edge in $F$, then we add $e$ to $F$; otherwise, we discard it. The set $F$ we finally get is our approximate solution.

## 2.1 Correctness

The set $F$ meets the definition of a matching: it is a subset of $E$, and no two edges in $F$ have a common end point, since we have discarded the edges that may introduce duplicated end points.

## 2.2 Time Complexity

Since we only need to enumerate the edges, the algorithm has a linear time complexity.

## 2.3 Approximation Ratio

$F$ is a maximal matching, which is defined as follows [1]:

**Definition 3** (Maximal matching)**.** A maximal matching of a graph $G = (V, E)$ is a matching $F$ such that for every edge $e \in E \setminus F$, the edge set $F \cup \{e\}$ fails to be a matching.

A maximum matching is always maximal, but the reverse does not always hold. However, we can show that a maximal matching $F$ can approximate the maximum matching $F^*$ well. To show this, we first need to define our measure of approximation, i.e., approximation ratio [1]:

**Definition 4** (Approximation ratio)**.** An algorithm for a problem has an approximation ratio of $\rho(n)$ if for any input of size $n$, the cost $C$ of the solution produced by the algorithm is within a factor of $\rho(n)$ of the cost $C^*$ of an optimal solution:

$$\max\left\{\frac{C}{C^*}, \frac{C^*}{C}\right\} \leq \rho(n).$$

If an algorithm achieves an approximation ratio of $\rho(n)$, we call it a $\rho(n)$-approximation algorithm.

The above definition of approximation ratio apply to both minimisation problems, where $C \geq C^*$, and maximisation problems, where $C \leq C^*$. The maximum matching problem is a maximisation problem, where $C := |F|$ and $C^* := |F^*|$. To show our greedy algorithm is 2-approximation, we only need to show that $|F^*| \leq 2|F|$ holds for any maximal matching $F$.

*Proof.* For any edge $e \in F^* \setminus F$, there must be an edge $e' \in F \setminus F^*$ such that $e$ and $e'$ share a common end point. Each edge $F \setminus F^*$ is responsible for at most two edges in $F^* \setminus F$. Thus,

$$\begin{aligned} |F^*| &= |F^* \setminus F| + |F^* \cap F| \\ &\leq 2|F \setminus F^*| + |F^* \cap F| \\ &\leq 2|F| \end{aligned}$$

□

Therefore, our greedy algorithm is 2-approximation.

## 3 Maximum Weight Matching

We can further expand the algorithm's scope of application and apply it to the maximum weight matching problem, which is defined as follows [4]:

**Definition 5** (Maximum weight matching)**.** A maximum weight matching of a weighted graph $G = (V, E, \omega)$ is a matching in which the sum of weights is maximised.

This problem has a classical algorithm which runs in $O(|E||V|^2)$ [4]. However, we can still use our greedy algorithm to get an approximate solution. The only modification we need to make is to first sort all the edges in descending order of weight. The time complexity is thus $O(|E| \log |E|)$.

To show our greedy algorithm is 2-approximation, the old analysis for the unweighted case can be made work here. The difference is that instead of focusing on the number of edges, we should reason about the sum of the edge weights. Furthermore, the ordering of edges plays an important role here, so $F$ is no longer an arbitrary maximal matching. Nevertheless, we can still prove that $\sum_{e \in F^*} \omega_E(e) \leq 2 \sum_{e \in F} \omega_E(e)$.

## 4 Summary

Using greedy algorithms, we can quickly get approximate solutions to some graph theory problems. They are especially useful when we focus on time efficiency rather than over-pursuing the optimal solution.

There have been many discussions on approximate algorithms for the maximum matching problem, and there are many algorithms with better approximation ratio and time complexity. A summary can be found in [2]. However, the elegant algorithms presented in this report demonstrate a fundamental pattern for analysing approximate algorithms, which can be transferred to other similar greedy-based approximation algorithm analysis.

## References


[1] Cormen, T. H., Leiserson, C. E., Rivest, R. L., & Stein, C. (2022). *Introduction to algorithms* (4th ed.). MIT Press.

[2] Duan, R., & Pettie, S. (2014). Linear-time approximation for maximum weight matching. *Journal of the ACM*, *61*(1), 1–23. https://doi.org/10.1145/2529989

[3] Richard, S. (2024). Graph theory. Retrieved 2024-07-20, from <http://www.math.nagoya-u.ac.jp/~richard/teaching/s2024/Graph.pdf>.

[4] Wikipedia. (2024). Maximum weight matching - Wikipedia. Retrieved 2024-07-20, from <https://en.wikipedia.org/wiki/Maximum_weight_matching>.