The Difficulty of Approximating Nash Social Welfare in Online Matching

A Preprint

Szymon Snoeck
Applied Mathematics Department
Columbia University
New York, NY 10027
sgs2179@columbia.edu

February 26, 2025

Abstract

Consider the problem of vertex-weighted online bipartite matching with respect to a given welfare function. It is well known that there exists an online matching algorithm that is an $(1 - \frac{1}{e})$-approximation algorithm when the welfare function is the sum of the weights [Aggarwal et al., 2010]. However, it is show that no constant factor approximation algorithm exists when the welfare function is the Nash Social Welfare.

1 Introduction

Bipartite matching is a classical problem in economics and computer science theory. Recently, its online vertex-weighted counter part has received increased attention due to its prevalence in internet advertising and social media algorithms [Mehta, 2013]. In vertex-weighted online matching, you are given a bipartite graph $G(U, V, E)$ where the vertices $u \in U$ have weights and are known ahead of time. The vertices in $v \in V$ are revealed one-by-one and each must be irrevocably matched to at most one vertex in U upon arrival. The goal is to maximize a given welfare function of the weights of the matched vertices in U.

Karp et al. [1990] was the first to study this problem and gave a beautiful result proving that there exists a $\left(1 - \frac{1}{e}\right)$-approximation algorithm when all weights are equal and the welfare function is the utilitarian social welfare function—the sum of the weights. Moreover, they show that this algorithm is optimal. Aggarwal et al. [2010] adapted Karp et al. [1990]'s algorithm to the case of arbitrary weights, showing that that it retains the $\left(1 - \frac{1}{e}\right)$-approximation factor.

Even more recently, there has been growing interest in vertex-weighted bipartite matching with respect to the Nash Social Welfare (NSW)—the geometric mean of the weights [Jain and Vaish, 2023][Gokhale et al., 2024]. This interest is motivated by the "unreasonably fairness" of the maximum NSW solution in the context of partitioning indivisible goods among a group of players, i.e. many-to-one bipartite matching, [Caragiannis et al., 2019]. The NSW seemingly strikes a nice middle ground between maximizing the sum of the player's utilities and maximizing the minimum player's utility. In particular, Caragiannis et al. [2019] showed that the maximum NSW solution is Pareto optimal and envy-free up to one good, meaning that any envy between two players is resolved by swapping just one good.

In general, it is not possible to have a polynomial time algorithm that approximates the maximum NSW solution of a many-to-one bipartite matching problem with an approximation factor arbitrarily close to 1 [Garg et al., 2017] [Garg et al., 2019]. That said, there still exists several constant factor approximation algorithms under various settings of the problem [Garg et al., 2023] [Barman et al., 2017].

Given that there exists constant factor approximation algorithms for many-to-one bipartite matching with respect to NSW and for online vertex-weighted bipartite matching with respect to the utilitarian social welfare function, one might reasonably hope that there also exists a constant factor approximation algorithm for online vertex-weighted bipartite matching when the welfare function is the NSW. However, this turns out not to be the case:

Theorem 1 (informal)**.** *There does not exist a vertex-weighted online bipartite matching algorithm that is a constant factor approximation algorithm of the maximum Nash Social Welfare solution.*

Thus, attempts to do online matching with respect to the NSW are doomed to fail. This poses a serious roadblock to improving the fairness of online bipartite matching and begs the question: are there fairness guarantees associated with the NSW which are not achievable in an online setting?

2 Formal Statement and Proof of Theorem 1

Let $G(U, V, E, \{w_u\}_{u \in U})$ denote a vertex-weighted bipartite graph where all vertices $u \in U$ have associated weight $w_u \in \mathbb{R}$.

Consider the following problem: there is a bipartite graph $G(U, V, E, \{w_u\}_{u \in U})$. The vertices $u \in U$ and their weights[1] $w_u \geq 1$ are known a priori, while the vertices $v \in V$ and their edges are revealed in an online fashion. When each vertex $v \in V$ arrives, it must be irrevocably matched to one of its neighbors in U. The goal is to maximize the geometric mean, i.e. Nash Social Welfare (NSW), of the weights of the matched vertices in U. More formally:

Definition 1. *Fix $G(U, V, E, \{w_u\}_{u \in U})$ and some matching $E' \subset E$. Let the matching be identified by $m : U \to V \cup \{0\}$, an injective function such that for all $(u, v) \in E'$, $m(u) = v$ and for all u unmatched, $m(u) = 0$. Then, the* ***Nash Social Welfare*** *of the matching is:*

$$\mathit{NSW}(m) = \left(\prod_{u \in U \text{ s.t. } m(u) \neq 0} w_u \right)^{\frac{1}{|U|}}.$$

Fix some algorithm and a graph $G(U, V, E, \{w_u\}_{u \in U})$, let $m_G^* : U \to V \cup \{0\}$ denote the matching that maximizes the NSW on the graph and let $m_G^{\text{alg}} : U \to V \cup \{0\}$ be the matching produced by the online algorithm. The ability of an algorithm to approximate the maximum NSW across all possible graphs is measured by the competitive ratio:

Definition 2. *Fix $n \in \mathbb{N}$. The* ***competitive ratio*** *of an algorithm is defined as:*

$$\mathit{CR}_n = \inf_{\substack{G(U,V,E,\{w_u\}_{u \in U}) \text{ s.t.} \\ |U| = n \text{ and arrival order of } V}} \frac{\mathit{NSW}(m_G^{alg})}{\mathit{NSW}(m_G^*)}.$$

If the algorithm is random, $\mathit{NSW}(m_G^{alg})$ is replaced by $\mathbb{E}[\mathit{NSW}(m_G^{alg})]$.

Now we can formally state **Theorem 1**:

Theorem 1. *For all even $n \in \mathbb{N}$:*

$$\sup_{\text{algorithm}} \mathit{CR}_n \leq \frac{1}{n^{1/8}} + \frac{\log(n)^2}{32n}.$$

Thus as $n \to \infty$, the $\mathit{CR}_n \to 0$ for all algorithms.

In other words, the above states that no online vertex-weighted bipartite matching algorithm can approximate the optimal solution with in a constant factor.

The key idea behind the proof of **Theorem 1** is to find a family of graphs such that no algorithm does too well on all of them. Let $k \geq 1$ be some constant to be specified later. As a toy example, consider the set of two graphs, $\mathcal{G}_2$, as depicted in fig. 1. Observe that both graphs have opposite optimal matchings despite appearing to be identical after only seeing v_1. Therefore, no algorithm can do better than to randomly match v_1. This is to say that no algorithm will out preform the random algorithm which matches each vertex to a uniformly chosen neighbor on $\mathcal{G}_2$. Let $m_G^{\text{rand}} : U \to V \cup \{0\}$ denote the matching produced by the random algorithm. Now, we can prove this intuition:

Lemma 1. *Fix arbitrary $G' \in \mathcal{G}_2$. For all algorithms, $\inf_{G \in \mathcal{G}_2} \mathbb{E}[\mathit{NSW}(m_G^{alg})] \leq \mathbb{E}[\mathit{NSW}(m_{G'}^{rand})]$.*

Proof. Fix some algorithm, $\mathcal{A}$, and the matchings it generates on $\mathcal{G}_2$: $\{m_G^{\text{alg}}\}_{G \in \mathcal{G}_2}$. Without loss of generality, assume this algorithm $\mathcal{A}$ always matches a vertex if possible. Indeed, any algorithm which willingly leaves a vertex unmatched trivially satisfies the hypothesis since the matching it produces will match at most one vertex and have a NSW of at most $\sqrt{k}$. The random algorithm will always match at least one vertex, so on either graph in $\mathcal{G}_2$, it must always have a NSW exceeding $\sqrt{k}$. Thus the hypothesis is satisfied and for the rest of the proof it can be assumed that $\mathcal{A}$ matches vertices whenever possible.

[1] Note that the weights must be ≥ 1, otherwise matching an edge would reduce the NSW.

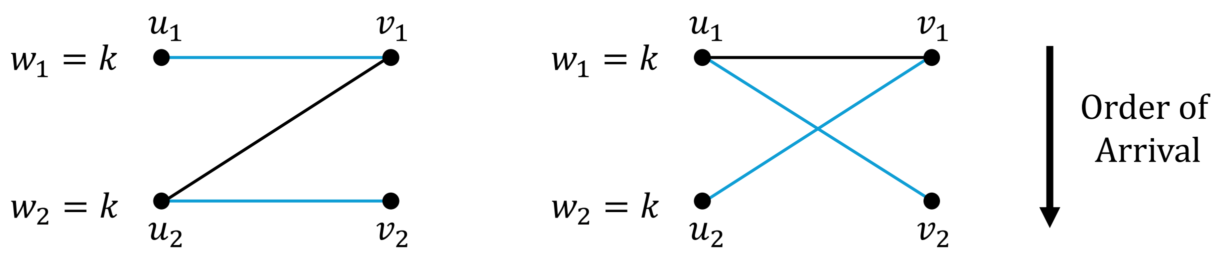

Figure 1: The set of graphs $\mathcal{G}_2$ with each graph's optimal matching highlighted in blue.

Let $m_G^{(1)} : U \to V \cup \{0\}$ be the matching returned on $G \in \mathcal{G}_2$ by the deterministic algorithm that always matches v_1 to u_1 and tries to match v_2 if possible. Let $m_G^{(2)} : U \to V \cup \{0\}$ be the same except the algorithm starts by matching v_1 to u_2. Because our algorithm $\mathcal{A}$ always matches whenever possible, it can be expressed as a distribution over these two deterministic matching strategies. Thus for algorithm $\mathcal{A}$, there exists $p_1, p_2 \in [0,1], p_1 + p_2 = 1$ such that:

$$\inf_{G\in\mathcal{G}_2} \mathbb{E}[\text{NSW}(m_G^{\text{alg}})] = \inf_{G\in\mathcal{G}_2} p_1\text{NSW}(m_G^{(1)}) + p_2\text{NSW}(m_G^{(2)})$$
$$\leq p_1\mathbb{E}_{G\in\mathcal{G}_2}[\text{NSW}(m_G^{(1)})] + p_2\mathbb{E}_{G\in\mathcal{G}_2}[\text{NSW}(m_G^{(2)})] \leq \max_{i\in\{1,2\}} \mathbb{E}_{G\in\mathcal{G}_2}[\text{NSW}(m_G^{(i)})].$$

Fix any $G' \in \mathcal{G}_2$, than for $i \in \{1,2\}$, $\mathbb{E}_{G\in\mathcal{G}_2}[\text{NSW}(m_G^{(i)})] = \mathbb{E}[\text{NSW}(m_{G'}^{\text{rand}})]$ since taking the expectation over the graph is equivalent to taking the expectation over the random algorithm's choice of v_1's match. Thus we get:

$$\inf_{G\in\mathcal{G}_2} \mathbb{E}[\text{NSW}(m_G^{\text{alg}})] \leq \mathbb{E}[\text{NSW}(m_{G'}^{\text{rand}})].$$

□

This lemma can be extended to a bound on CR_2 by computing $\mathbb{E}[\text{NSW}(m_G^{\text{rand}})]$ for $G \in \mathcal{G}_2$. A similar idea is used to prove **Theorem 1**:

Proof of Theorem 1. For n even, define $\mathcal{G}_n$ to be the collection of all graphs $G(U, V, E, \{w_u\}_{u\in U})$ such that G is composed of $\frac{n}{2}$ disconnected graphs from $\mathcal{G}_2$ where the vertices in V of degree 2 arrive before the vertices of degree 1 (see fig. 1), and all weights, w_u, have value $k = \sqrt{n}$.

Fix some algorithm $\mathcal{A}$. Since each of the $\frac{n}{2}$ components of the graphs in $\mathcal{G}_n$ are disconnected and independently picked from $\mathcal{G}_2$, we can assume the algorithm never willingly leaves a vertex unmatched via the same argument used in **Lemma 1**'s proof. Let $\mathcal{M}$ be the set of all deterministic matching strategies on $\mathcal{G}_n$, i.e. all sequences of applying either $m_G^{(1)}$ or $m_G^{(2)}$, from the proof of **Lemma 1**, to each of the $\frac{n}{2}$ disconnected components from $\mathcal{G}_2$. For example, one strategy $a \in \mathcal{M}$ would be to alternate applying $m_G^{(1)}$ and $m_G^{(2)}$ to the $\frac{n}{2}$ components as they arrive. The matching found on $G \in \mathcal{G}_n$ by some $a \in \mathcal{M}$ is written m_G^a. By assumption $\mathcal{A}$ always matches when possible, so, it can be represented as a distribution, $\mathcal{D}$, over the set of all deterministic strategies, $\mathcal{M}$. Thus:

$$\text{CR}_n \leq \inf_{G\in\mathcal{G}_n} \frac{\mathbb{E}[\text{NSW}(m_G^{\text{alg}})]}{\text{NSW}(m_G^*)}$$
$$= \inf_{G\in\mathcal{G}_n} \frac{\mathbb{E}_{a\sim_{\mathcal{D}}\mathcal{M}}[\text{NSW}(m_G^a)]}{\sqrt{n}} \leq \frac{1}{\sqrt{n}} \mathop{\mathbb{E}}_{a\sim_{\mathcal{D}}\mathcal{M}} \left[\mathop{\mathbb{E}}_{G\in\mathcal{G}_n} [\text{NSW}(m_G^a)] \right]$$
$$\leq \max_{a\in\mathcal{M}} \frac{1}{\sqrt{n}} \mathop{\mathbb{E}}_{G\in\mathcal{G}_n} [\text{NSW}(m_G^a)] = \frac{1}{\sqrt{n}} \mathbb{E}[\text{NSW}(m_{G'}^{\text{rand}})],$$

where $G' \in \mathcal{G}_n$ is some arbitrarily chosen graph, and the last inequality holds since expectation over $G \in \mathcal{G}_n$ is equivalent to expectation over the internal choices of the random algorithm. Now all that remains is to bound $\mathbb{E}[\text{NSW}(m_{G'}^{\text{rand}})]$.

Observe the following technical fact: for some random variable x supported on $[0,1]$ and $k \geq 1$ it holds that:

$$\mathbb{E}[k^x] - k^{\mathbb{E}[x]} \leq \frac{1}{2} k \log(k)^2 \text{Var}(x).$$

Indeed $\frac{d^2}{d^2x} k^x = k^x \log(k)^2 \leq k \log(k)^2$ for $x \in [0,1]$. Thus, $k^x - \frac{1}{2} k \log(k)^2 x^2$ is a concave function. By Jensen's inequality:

$$\begin{aligned} \mathbb{E}\left[k^x - \frac{1}{2} k \log(k)^2 x^2\right] &\leq k^{\mathbb{E}[x]} - \frac{1}{2} k \log(k)^2 \mathbb{E}[x]^2 \\ \iff \mathbb{E}[k^x] - k^{\mathbb{E}[x]} &\leq \frac{1}{2} k \log(k)^2 \left(\mathbb{E}[x^2] - \mathbb{E}[x]^2\right) \\ &= \frac{1}{2} k \log(k)^2 \text{Var}(x). \end{aligned}$$

Define the random variable n_G^{rand} as the number of vertices matched by the random algorithm on a bipartite graph G with a fixed order of arrival. Recall all edges in $G \in \mathcal{G}_n$ have weight $\sqrt{n}$. Thus:

$$\begin{aligned} \mathbb{E}[\text{NSW}(m_{G'}^{\text{rand}})] &= \mathbb{E}[n^{\frac{n_G^{\text{rand}}}{2n}}] \\ &= n^{\mathbb{E}[\frac{n_G^{\text{rand}}}{2n}]} + (\mathbb{E}[n^{\frac{n_G^{\text{rand}}}{2n}}] - n^{\mathbb{E}[\frac{n_G^{\text{rand}}}{2n}]}) \\ &\leq n^{\mathbb{E}[\frac{n_G^{\text{rand}}}{2n}]} + \frac{1}{8}\sqrt{n} \log(n)^2 \text{Var}(\frac{n_G^{\text{rand}}}{n}). \end{aligned}$$

Where in the last step, the technical fact from earlier is used. Note that on each disconnected component the random algorithm will match both vertices correctly with 50% probability or connect only one with 50% probability. Hence n^{rand} is equivalent in distribution to $X + \frac{n}{2}$ where X is binomial random variable: $X \sim B(\frac{n}{2}, 0.5)$. Therefore the above becomes:

$$= n^{\mathbb{E}[\frac{X+\frac{n}{2}}{2n}]} + \frac{1}{8} \frac{\sqrt{n} \log(n)^2}{n^2} \text{Var}(X + \frac{n}{2}) = n^{\frac{3}{8}} + \frac{\log(n)^2}{32\sqrt{n}}.$$

Thus:

$$\sup_{\text{algorithm}} \text{CR}_n \leq \frac{1}{\sqrt{n}} \mathbb{E}[\text{NSW}(m_G^{\text{rand}})] \leq \frac{1}{\sqrt{n}} \left(n^{\frac{3}{8}} + \frac{\log(n)^2}{32\sqrt{n}}\right) = \frac{1}{n^{1/8}} + \frac{\log(n)^2}{32n}.$$

□

Acknowledgments. I thank Professor Yuri Faenza and Christopher En for their expert guidance and advice while doing my research.

References

G. Aggarwal, G. Goel, C. Karande, and A. Mehta. Online vertex-weighted bipartite matching and single-bid budgeted allocations. *CoRR*, abs/1007.1271, 2010. URL `http://arxiv.org/abs/1007.1271`.

S. Barman, S. K. K. Murthy, and R. Vaish. Finding fair and efficient allocations. *CoRR*, abs/1707.04731, 2017. URL `http://arxiv.org/abs/1707.04731`.

I. Caragiannis, D. Kurokawa, H. Moulin, A. D. Procaccia, N. Shah, and J. Wang. The unreasonable fairness of maximum nash welfare. *ACM Trans. Econ. Comput.*, 7(3), 9 2019. ISSN 2167-8375. doi: 10.1145/3355902. URL `https://doi.org/10.1145/3355902`.

J. Garg, M. Hoefer, and K. Mehlhorn. Approximating the nash social welfare with budget-additive valuations. *CoRR*, abs/1707.04428, 2017. URL `http://arxiv.org/abs/1707.04428`.

J. Garg, P. Kulkarni, and R. Kulkarni. Approximating nash social welfare under submodular valuations through (un)matchings. *CoRR*, abs/1912.12541, 2019. URL `http://arxiv.org/abs/1912.12541`.

J. Garg, E. Husić, W. Li, L. A. Végh, and J. Vondrák. Approximating nash social welfare by matching and local search. In *Proceedings of the 55th Annual ACM Symposium on Theory of Computing*, STOC 2023, page 1298–1310, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450399135. doi: 10.1145/3564246.3585255. URL `https://doi.org/10.1145/3564246.3585255`.

S. Gokhale, H. Sagar, R. Vaish, and J. Yadav. Approximating one-sided and two-sided nash social welfare with capacities, 2024. URL `https://arxiv.org/abs/2411.14007`.

P. Jain and R. Vaish. Maximizing nash social welfare under two-sided preferences, 2023. URL `https://arxiv.org/abs/2312.09167`.

R. M. Karp, U. V. Vazirani, and V. V. Vazirani. An optimal algorithm for on-line bipartite matching. In *Proceedings of the Twenty-Second Annual ACM Symposium on Theory of Computing*, STOC '90, page 352–358, New York, NY, USA, 1990. Association for Computing Machinery. ISBN 0897913612. doi: 10.1145/100216.100262. URL `https://doi.org/10.1145/100216.100262`.

A. Mehta. Online matching and ad allocation. *Found. Trends Theor. Comput. Sci.*, 8(4):265–368, Oct. 2013. ISSN 1551-305X. doi: 10.1561/0400000057. URL `https://doi.org/10.1561/0400000057`.