Socially Fair Matching: Exact and Approximation Algorithms*

Sayan Bandyapadhyay[1][0000−0001−8875−0102], Fedor Fomin[2][0000−0003−1955−4612], Tanmay Inamdar[2][0000−0002−0184−5932], Fahad Panolan[3][0000−0001−6213−8687], and Kirill Simonov[4][0000−0001−9436−7310]

[1] Portland State University, United States sayanb@pdx.edu
[2] University of Bergen, Norway {Fedor.Fomin,Tanmay.Inamdar}@uib.no
[3] Indian Institute of Technology, Hyderabad, India fahad@cse.iith.ac.in
[4] Hasso Plattner Institute, University of Potsdam, Germany Kirill.Simonov@hpi.de

Abstract. Matching problems are some of the most well-studied problems in graph theory and combinatorial optimization, with a variety of theoretical as well as practical motivations. However, in many applications of optimization problems, a "solution" corresponds to real-life decisions that have major impact on humans belonging to diverse groups defined by attributes such as gender, race, or ethnicity. Due to this motivation, the notion of *algorithmic fairness* has recently emerged to prominence. Depending on specific application, researchers have introduced several notions of fairness.

In this paper, we study a problem called Socially Fair Matching, which combines the traditional Minimum Weight Perfect Matching problem with the notion of *social fairness* that has been studied in clustering literature [Abbasi et al., and Ghadiri et al., FAccT, 2021]. In our problem, the input is an edge-weighted complete bipartite graph, where the bipartition represent two groups of entities. The goal is to find a perfect matching as well as an assignment that assigns the cost of each matched edge to one of its endpoints, such that the maximum of the total cost assigned to either of the two groups is minimized.

Unlike Minimum Weight Perfect Matching, we show that Socially Fair Matching is weakly NP-hard. On the positive side, we design a *deterministic* PTAS for the problem when the edge weights are arbitrary. Furthermore, if the weights are integers and polynomial in the number of vertices, then we give a randomized polynomial-time algorithm that solves the problem exactly. Next, we show that this algorithm can be used to obtain a *randomized* FPTAS when the weights are arbitrary.

Keywords: Fairness · Matching · Approximation Algorithms.

* The research leading to these results has received funding from the Research Council of Norway via the project BWCA (grant no. 314528), and the European Research Council (ERC) via grant LOPPRE, reference 819416.

1 Introduction

Matching is a ubiquitous problem in computer science, since many optimization problems in practice can be interpreted as *assignment* problems, and matchings in (bipartite) graphs are a natural candidate for modeling such problems. The polynomial-time *Blossom* algorithm of Edmonds [9] for computing a maximum matching is one of the cornerstones of algorithmic graph theory and combinatorial optimization. Traditionally, optimization problems have focused on optimizing a single objective function subject to certain constraints based on the problem. In this viewpoint, all the different aspects of a solution are condensed into a single number, called the *cost* of the solution. This model allows for a clean abstraction of the problem, which is useful for studying the problem from theoretical point of view.

In many real-world applications, however, an optimization problem inherently involves different tradeoffs. For example, suppose there are two possible locations for building a new school in a community – one location is cheap, but the location is extremely inconvenient for students of one demographic group in the community over the other; on the other hand the second location is relatively expensive, but is easily accessible to students of all demographic groups. In such a situation, it is vastly preferable to choose the second location for the school. Researchers have considered several approaches to alleviate this issue. One approach is the problem of multi-objective optimization (see [3, 21]), which adds several objective functions to an optimization problem. Another related approach is to model the *fairness* aspect directly into the problem, where the notion of fairness may be specific to the problem at hand. In this paper, we follow the second approach.

We study Socially Fair Matching, which introduces a notion of fairness called *social fairness* in the classical Minimum Weight Perfect Matching problem. Social fairness was introduced very recently in the context of clustering problems [1, 12, 20], which is useful to balance the total clustering cost over all groups. In this work, we introduce the Socially Fair Matching problem. To put it in the context, let us consider the classical edge-weighted bipartite matching problem, where the goal is to find a minimum cost perfect matching in a complete bipartite graph between two groups R and B each containing n vertices. Now, for any matched edge (u, v) with $u \in R$ and $v \in B$, depending on the application, the cost might be paid by either u or v. Thus, the total cost for the two groups R and B might not be well-balanced. To address a similar issue in the context of clustering problems, Abbasi et al. [1], and Ghadiri et al. [12] proposed the notion of *social fairness*. We adopt this notion in the context of matching. Thus, in Socially Fair Matching, the goal is to find a perfect matching in a complete bipartite graph as well as an assignment that assigns the cost of each matched edge to one of its endpoints, such that the maximum of the total cost assigned to either of the two groups is minimized.

Twinning of cities is a legal or social agreement between two geographically and politically distinct cities to promote cultural and commercial ties [23]. This phenomenon goes back centuries, but in modern history, cities in two different countries are twinned as an alternative channel for diplomacy. Consider the situation where $2n$ cities from two different countries desire to be twinned with each other, such that each city of one country is twinned with a city of another country. For each pair of twin cities, the headquarters may be located in one of the two cities, which must bear the administrative cost that depend on the specific parameters for twinning the specific pair of cities. In this application, it might be desirable that we come up with a pairing, such that the total expenses born by cities of each country is minimized. Note that this can be modeled as an instance of Socially Fair Matching, where the weight of each edge represents the administrative cost of twinning two cities.

Our results and contributions. We first observe that Socially Fair Matching is weakly NP-hard, when the edge weights are arbitrary integers. The reduction is via the well-known Partition problem, which asks whether it is possible to partition a given set of integers into two parts with equal weights. In contrast, we show when the edge weights are integers and polynomial in n, the problem can be solved exactly in polynomial time using a randomized algorithm. For this result, we reduce Socially Fair Matching to the problem of polynomial identity testing, which can be solved in randomized polynomial time via an application of the Schwartz-Zippel Lemma. For the case of general weights, we show how to obtain a $(1+\epsilon)$-approximation using two different approaches. First, we show that the case of general weights can be reduced to that with polynomial integer weights at a small loss in the approximation guarantee. Thus, we can obtain a *randomized* FPTAS [5] via the previous result. In a different direction, we show that one can also obtain a *deterministic* PTAS [6] in this case. Despite having a worse running time as compared to the previous FPTAS, we believe that this result is interesting for a few reasons. First, the PTAS is deterministic, unlike the inherently randomized nature of the previous FPTAS due to its reliance on the Schwartz-Zippel lemma. Another reason is that the PTAS is entirely combinatorial – we first guess (i.e., enumerate) a subset of *heavy* edges of an optimal solution, and essentially reduce the problem to classical Minimum Weight Perfect Matching. Thus, another advantage is that it does not rely on any sophisticated algebraic machinery such as polynomial identity testing.

Related work. So-called Partitioned Min-Max Weighted Matching (PMMWM), has been studied in the operations research literature [19]. In this problem, we are given an edge weighted bipartite graph $G = (R \uplus B, E)$, and an integer $m \le |R|$. The goal is to find a partition $R_1, R_2, \ldots, R_m$ of R, and a matching M saturating R, such that the maximum total weight of matched edges (i.e., edges of M) incident to all vertices in R_i is minimized. Furthermore, it is required that for every $1 \le i \le m$, $|R_i| \le u$, where u is an upper bound given in the input. Kress et al. [19] establish hardness results and approximation algorithms

[5] FPTAS stands for *Fully Polynomial-Time Approximation Scheme*, i.e., for any $\epsilon > 0$, an algorithm that returns a $(1+\epsilon)$-approximation in time $(n/\epsilon)^{O(1)}$.

[6] PTAS stands for *Polynomial-Time Approximation Scheme*, i.e., for any $\epsilon > 0$, an algorithm that returns a $(1+\epsilon)$-approximation in time $n^{f(1/\epsilon)}$.

for PMMWM in general case. Even though seemingly unrelated to Socially Fair Matching, note that if for the input to PMMWM, it holds that (i) G is a complete bipartite graph with $|R| = |B| = n$, (ii) m, the number of parts of the partition is equal to 2, and (iii) the upper bound u is equal to $|R|$ (i.e., there is no upper bound on the size of any part), then the resulting problem is equivalent to Socially Fair Matching. To the best of our knowledge, for this special case of PMMWM, no improved approximation results are known. Another problem related to PMMWM is the so-called Min-Max Weighted Matching (MMWM) problem [2, 6], where the only difference from PMMWM is that partition of R into $R_1, R_2, \ldots, R_m$ is given in the input, G is complete, and $n = |R| = |B|$. Duginov [6] establishes several results for this problem. Among these, they observe that MMWM is related to the well-known Exact Perfect Matching problem, which has a randomized polynomial-time algorithm, but obtaining a deterministic one is a long-standing open problem. Although the setting of MMWM appears more similar to Socially Fair Matching as compared to PMMWM, the fact that the partition $R_1, R_2, \ldots, R_m$ is given in the input in MMWM, makes the two problems quite different.

Further related work on fairness and matching. In recent years, researchers have introduced and studied several different notions of fairness, e.g., disparate impact [10], statistical parity [15, 22], individual fairness [7] and group fairness [8]. Kleinberg et al. [18] formulated three notions of fairness and showed that it is theoretically impossible to satisfy them simultaneously. See also [4, 5] for similar exposures.

Several different fair matching problems have been studied in the literature. Huang et al. [13] studied fair b-matching, where matching preferences for each vertex are given as ranks, and the goal is to avoid assigning vertices to high ranked preferences as much as possible. Fair-by-design-matching is studied by Garcia-Soriano and Bonchi [11], where instead of a single matching, a probability distribution over all feasible matchings is computed which guarantees individual fairness. See also [14, 17].

Organization. In Section 2, we formally define our problem. The deterministic PTAS is discussed in Section 3. Section 4 contains the randomized polynomial-time algorithm, the FPTAS, and the hardness result. Finally, in Section 5, we conclude with some interesting open questions.

2 Preliminaries

For an integer $\ell \geq 1$, we use the notation $[\ell] \coloneqq \{1, 2, \ldots, \ell\}$.

Socially Fair Matching. In Socially Fair Matching, the input is a complete bipartite graph $G = (R \uplus B, E)$, where $R \uplus B$ is a bipartiton of $V(G)$, with $|R| = |B| = n$. We will often refer to the vertices of R and B as red and blue respectively. Each edge in $E(G)$ has a non-negative weight. The goal is to compute a perfect matching $M \subseteq E(G)$, and an assignment $f : M \to \{\mathsf{red}, \mathsf{blue}\}$, such that $\max\left\{\sum_{e \in M(\mathsf{red})} w(e), \sum_{e \in M(\mathsf{blue})} w(e)\right\}$ is minimized, where $M(\mathsf{red})$ is the set of edges in M such that $f(e) = \mathsf{red}$, and $M(\mathsf{blue})$ is defined analogously.

Fields, Polynomials, Vectors and Matrices

Here, we review some definitions from linear algebra. We refer to any graduate textbook on algebra for more details. For a finite field $\mathbb{F}$ and a set of variables $X = \{x_1, \ldots, x_n\}$, $\mathbb{F}[X]$ denotes the ring of polynomials in X over $\mathbb{F}$. The *characteristic* of a field is defined as least positive integer m such that $\sum_{i=1}^{m} 1 = 0$.

A vector v over a field $\mathbb{F}$ is an array of values from $\mathbb{F}$. The matrix is said to have dimension $n \times m$ if it has n rows and m columns. For a vector v, we denote its *transpose* by v^T. The the rank of a matrix is the maximum number k such that there is a $k \times k$ submatrix whose determinant is non-zero.

3 A Deterministic PTAS for Arbitrary Weights

Let M be a perfect matching, and $f : M \to \{\mathsf{red}, \mathsf{blue}\}$ be an arbitrary assignment. Then, for a vertex $r \in R$, we define

$$\mu_{M,f}(r) \coloneqq \begin{cases} 0 & \text{if } e = \{r, b\} \in M \text{ with } f(e) = \mathsf{red} \\ w(e) & \text{if } e = \{r, b\} \in M \text{ with } f(e) = \mathsf{blue} \end{cases}$$

and for $b' \in B$, we define

$$\mu_{M,f}(b') \coloneqq \begin{cases} 0 & \text{if } e = \{r', b'\} \in M \text{ with } f(e) = \mathsf{blue} \\ w(e) & \text{if } e = \{r', b'\} \in M \text{ with } f(e) = \mathsf{red} \end{cases}.$$

For a subset $R' \subseteq R$ (resp. $B' \subseteq B$), we define $\mu_{M,f}(R') \coloneqq \sum_{r \in R'} \mu_{M,f}(r)$ (resp. $\sum_{b \in B'} \mu_{M,f}(b)$).

Fix an optimal solution $M^* \subseteq E(G)$, and the corresponding assignment $f^* : M^* \to \{\mathsf{red}, \mathsf{blue}\}$. Define $M^*(\mathsf{red})$ and $M^*(\mathsf{blue})$ as the sets of edges assigned red and blue by f^* respectively. For $v \in V(G)$, we use the shorthand $\mu^*(v) \coloneqq \mu^*_{M^*,f^*}(v)$. Note that $OPT = \max\{\mu^*(R), \mu^*(B)\}$.

Let $t = 1 + \lceil 1/\epsilon \rceil$. Let $R_1 \subseteq R$ be the set of vertices incident to the heaviest $\max\{t, |M^*(\mathsf{red})|\}$ edges in $M^*(\mathsf{red})$. Similarly, let $B_1 \subseteq B$ be the set of vertices incident on the heaviest $\max\{t, |M^*(\mathsf{blue})|\}$ edges in $M^*(\mathsf{blue})$. Here, we assume that the ties are broken arbitrarily in the previous definitions.

Let R_1' denote the matched endpoints of vertices in B_1, i.e., $R_1' \coloneqq \{r \in R : \exists b \in B_1 \text{ such that } \{r, b\} \in M^*\}$. Similarly, define $B_1' \coloneqq \{b \in B : \exists r \in R_1 \text{ such that } \{r, b\} \in M^*\}$. Note that $|R_1| + |R_1'| = |B_1| + |B_1'| \leq 2t$. Now, define $R' = R \setminus (R_1 \cup R_1')$, and $B' = B \setminus (B_1 \cup B_1')$.

The following observation is easy to follow.

Observation 1 *For any $r \in R'$, let $\{r, b\} \in M^*$ be the matched edge incident on r. Then, $b \in B'$. For any $b' \in B'$, let $\{r', b'\} \in M^*$ be the matched edge incident on b'. Then, $r \in R'$.*

We also have the following observation.

Observation 2 *For any $r \in R'$, $\mu^*(r) \leq \epsilon \cdot \mu^*(R) \leq \epsilon \cdot OPT$. For any $b \in B'$, $\mu^*(b) \leq \epsilon \cdot \mu^*(B) \leq \epsilon \cdot OPT$.*

Proof. Suppose for contradiction that there is some $r' \in R' \neq \emptyset$ (wlog) such that $\mu^*(r') > \epsilon \cdot \mu^*(R)$. Then, by the definition of R_1, for all $r \in R_1$, $\mu^*(r) \geq \mu^*(r') > \epsilon \cdot \mu^*(R)$. This implies that $\mu^*(R_1) > \lceil 1/\epsilon \rceil \cdot \epsilon \cdot \mu^*(R) \geq OPT$, which is a contradiction. The proof is exactly the same for any $b' \in B'$.

Let $\tilde{M^*} \subseteq M^*$ be the subset of edges of an optimal solution that are incident on $V' := R_1 \cup R'_1 \cup B_1 \cup B'_1$. The first step is to guess $\tilde{M^*}$ and the partial optimal assignment $f^* : \tilde{M^*} \to \{\mathsf{red}, \mathsf{blue}\}$. Note that since $|\tilde{M^*}| \leq 2t = O(1/\epsilon)$, we can enumerate all $n^{O(t)} = n^{O(1/\epsilon)}$ possible choices. We are left with a smaller instance $R' \cup B'$, such that $|R'| = |B'| = n - 2t$, where $R' = R \setminus (R_1 \cup R'_1)$ and $B' = B \setminus (B_1 \cup B'_1)$. Assuming we are working with the correct guess, we also have an upper bound of $U \leq \epsilon \cdot OPT$, which can be inferred from the smallest distance in the partial solution already guessed.

Let $OPT' = \max\{\mu^*(R'), \mu^*(B')\}$ denote the optimal assignment cost in the remaining instance. Henceforth, wlog assume that $OPT' = \mu^*(R') = \frac{1}{\delta}\mu^*(B')$ for some $\delta \in [0, 1]$. The other case is symmetric, and we can run the algorithm by exchanging the roles of red and blue points, and select the solution with smaller cost.

Let $G' = (R' \cup B', E')$ be the subgraph of $G[R' \cup B']$, where we delete all edges with weight greater than U. Let M denote a minimum weight perfect matching in G'. We know such a perfect matching exists, because we are working with the correct guess, which implies that the assignment corresponding to OPT' is a perfect matching such that the weight of any edge is at most $U \leq \epsilon \cdot OPT$.

Lemma 1. *$OPT' \leq w(M) \leq \mu^*(R') + \mu^*(B') = (1 + \delta) \cdot OPT'$. Furthermore, there is an assignment $f' : M \to \{\mathsf{red}, \mathsf{blue}\}$ such that $\max\{\mu_{M,f'}(R'), \mu_{M,f'}(B')\} \leq (1 + \delta) \cdot OPT'$.*

Proof. Let M' be a matching corresponding to the optimal solution OPT'. Then,

$$w(M) \leq w(M') = \mu^*(R') + \mu^*(B') = (1 + \delta) \cdot \mu^*(R')$$

Now, we construct an arbitrary assignment $f' : M \to \{\mathsf{red}, \mathsf{blue}\}$, and note that

$$\max\{\mu_{M,f'}(R'), \mu_{M,f'}(B')\} \leq w(M) = (1 + \delta) \cdot \mu^*(R') = (1 + \delta) \cdot OPT'.$$

Finally, since this is a valid solution of cost at most $w(M)$, the cost of optimal solution OPT' must be at most $w(M)$.

As a first step, we try to obtain an assignment $f : M \to \{\mathsf{red}, \mathsf{blue}\}$ that achieves the red to blue cost ratio approximately equal to $1/\delta$. Since we do not know the exact value of δ, we will try the ratios $1/(1 + \epsilon)^s$, for $s = 0, 1, \ldots, q$. In order to upper bound q, let us consider a simpler case when δ is very small.

Case 1: $\delta < 2\epsilon$**.** If $\mu^*(B') < 2\epsilon \cdot OPT$, then for every edge $e \in M$, we define the assignment $f'(e) = B$. We construct the assignment $f : M \cup \tilde{M}^* \to \{\mathsf{red}, \mathsf{blue}\}$ by defining $f(e) = f'(e)$ if $e \in M$; and $f(e) = f^*(e)$ otherwise. Note that,

$$\begin{aligned} \text{Blue cost} &= \sum_{b \in B'} \mu_{M,f'}(b) + \sum_{b \in B_1} \mu^*(b) \\ &= 0 + \mu^*(B_1) \le \mu^*(B) \le OPT, \end{aligned}$$

And,

$$\begin{aligned} \text{Red cost} &= \sum_{r \in R'} \mu_{M,f'}(r) + \sum_{r \in R_1} \mu^*(r) \\ &= w(M) + \mu^*(R_1) \\ &\le \mu^*(R') + \mu^*(B') + \mu^*(R_1) && \text{(From Lemma 1)} \\ &\le \mu^*(R) + 2\epsilon \cdot OPT \\ &\le (1 + 2\epsilon) \cdot OPT. \end{aligned}$$

Case 2: $2\epsilon \le \delta \le 1$**.** In this case, have that $\mu^*(B') = \delta OPT' \ge 2\epsilon OPT \ge 2\epsilon OPT'$. By trying $s = 1, 2, \dots, q$, we want to find a value s such that $(1+\epsilon)^{-(s+1)} \le \frac{\delta}{1+\delta} < (1+\epsilon)^{-s}$. Note that $\frac{\delta}{1+\delta} \ge \frac{2\epsilon}{2} = \epsilon$, since $2\epsilon \le \delta \le 1$. This implies that $q \le -\lceil \log_{1+\epsilon}\left(\frac{\delta}{1+\delta}\right) \rceil \le -\lceil \frac{\log \epsilon}{\log(1+\epsilon)} \rceil = g(\epsilon)$ for some g. Furthermore, this implies that the weight of any non-removed edge is at most $\epsilon \cdot OPT \le (1+\epsilon)^{-s} \cdot OPT$ for any value of s we will consider.

Lemma 2. *When $2\epsilon \le \delta \le 1$, we can obtain an assignment $f : M_c \to \{\mathsf{red}, \mathsf{blue}\}$, where $M_c = M \cup M^*$ such that,*

$$\max \left\{ \sum_{r \in R} \mu_{M_c,f}(r), \quad \sum_{b \in B} \mu_{M_c,f}(b) \right\} \le (1 + 2\epsilon) \cdot OPT.$$

Proof. Order the edges in the matching M in an arbitrary order, and let the weight of edge e_i be w_i. Let j be the index such that $\sum_{i=1}^{j} w_i < (1+\epsilon)^{-s} \cdot w(M)$, but $\sum_{i=1}^{j+1} w_i \ge (1+\epsilon)^{-s} \cdot w(M)$.

Note that such an index i always exists, since the weight of any non-removed edge – in particular that of e_1 – is at most $\epsilon \cdot OPT \le (1+\epsilon)^{-q} OPT$, which corresponds to the largest value of s we consider.

Now, let R_C denote the red endpoints of edges $e_1, e_2, \dots, e_{j+1}$, and let B_C denote the blue endpoints of edges $e_{j+2}, e_{j+3}, \dots, e_{k'}$. We construct the assignment $f : M \to \{\mathsf{red}, \mathsf{blue}\}$ as follows. For each edge $e \in \{e_1, \dots, e_{j+1}\}$, we let $f(e) = \mathsf{red}$, and for each edge $e' \in \{e_{j+2}, \dots, ek'\}$, we let $f(e') = \mathsf{blue}$. Finally, we extend this assignment to $\tilde{M}^*$ by assigning for each edge $e \in \tilde{M}^*$ as $f(e) = f^*(e)$. Thus, now f is an assignment from $M \cup \tilde{M}^*$ to $\{\mathsf{red}, \mathsf{blue}\}$. Now we analyze the cost of this assignment.

Consider the following,

$$\begin{aligned}\sum_{i=1}^{j} w_i &\leq (1+\epsilon)^{-s} w(M) \\ &\leq \frac{(\mu^*(R') + \mu^*(B'))}{(1+\epsilon)^s} \\ &= (1+\epsilon)\cdot(1+\epsilon)^{-(s+1)}\cdot\left(\frac{\delta+1}{\delta}\right)\cdot\mu^*(B') \\ &\leq (1+\epsilon)\cdot\mu^*(B') \qquad (1)\end{aligned}$$

$$\begin{aligned}\sum_{i=j+2}^{k'} w_i &= w(M) - \sum_{i=1}^{j+1} w_i \\ &\leq w(M) - \frac{w(M)}{(1+\epsilon)^s} \\ &\leq \frac{\mu^*(R') + \delta\mu^*(R')}{\delta+1} \\ &\leq \mu^*(R'). \qquad (2)\end{aligned}$$

Here, the last inequality in (1) and the second-last inequality in (2) follow from the definition of s, i.e., $(1+\epsilon)^{-(s+1)} \leq \frac{\delta}{1+\delta} < (1+\epsilon)^{-s}$. Therefore,

$$\begin{aligned}\text{Blue cost} &= \sum_{i=1}^{j+1} w_i + \mu^*(B_1) \\ &\leq (1+\epsilon)\cdot\mu^*(B') + \mu(B_1) + w_{j+1} \\ &\leq (1+\epsilon)\cdot\mu^*(B) + \epsilon\cdot OPT\end{aligned}$$

$$\begin{aligned}\text{Red cost} &= \sum_{i=1}^{j+2} w_i + \mu^*(R_1) \\ &\leq \mu^*(R') + \mu^*(R_1) \\ &= \mu^*(R).\end{aligned}$$

Since the cost of the solution returned is the maximum of red cost and the blue cost, it is easy to show that the cost of our solution is upper bounded by $(1+2\epsilon)\cdot OPT$.

Theorem 1. *There exists a deterministic PTAS for* Socially Fair Matching *for arbitrary weights. In other words, for any $\epsilon > 0$, there exists a deterministic algorithm that returns a $(1+\epsilon)$-approximation for* Socially Fair Matching *in time $n^{O(1/\epsilon)}$.*

4 Randomized Polynomial Time algorithm for Polynomial Weights

First, we assume that the weights are all integers in the range $[0, N]$, where N is an integer that is at least n (if not, a simple scaling ensures this property). We will describe an exact randomized algorithm that runs in time polynomial in n and N in this case.

Let $\mathbb{F}$ be a field of characteristic 2 containing at least $4(N+1)^2 \geq n^2$ distinct elements. If $X = \{x_1, x_2, \ldots, x_t\}$ is a set of t variables, then we use $\mathbb{F}[X]$ to denote the ring of polynomials in X.

Let $Z = \{z_{ij} : 1 \leq i, j \leq n\}$ be a set of n^2 variables, and let $X = \{x, y\} \cup Z$, where a variable z_{ij} corresponds to the edge $e_{ij} = \{i, j\} \in E(G)$. We define a matrix $A = (A_{ij})$, where

$$A_{ij} = \begin{cases} 0 & \text{if } \{i,j\} \notin E(G) \\ (x^{w_{ij}} + y^{w_{ij}}) \cdot z_{ij} & \text{if } \{i,j\} \in E(G) \text{ with weight } w_{ij} \end{cases}$$

First, we observe that the permanent of the matrix A computed in $\mathbb{F}[X]$ is equal to the determinant of A, which is a polynomial in X. Let Π be the set of permutations of n. Then, we have the following equality:

$$Q = \det(A) = \sum_{\sigma \in \Pi} \prod_{q=1}^{n} A_{q,\sigma(q)} = \sum_{i=0}^{N} x^i P_i(y, Z) = \sum_{i=0}^{N} x^i \sum_{j=0}^{N} y^j \cdot P_{i,j}(Z)$$

where $Q = Q(x, y, Z)$ is a polynomial in variables x, y and Z, each $P_i(y, Z)$ is a polynomial in y and Z, and $P_{i,j}(Z)$ is a polynomial in variables Z. Note that the degree of each polynomial $P_{i,j}(Z)$, which is equal to the maximum degree of any of its monomials, is at most n.

Observation 3 *There exists a perfect matching M and an assignment $f : M \to \{\mathsf{red}, \mathsf{blue}\}$ such that $\mu_{M,f}(R) = w_r$ and $\mu_{M,f}(B) = w_b$ iff the polynomial $P_{w_r,w_b}(z)$ is not identically equal to zero.*

Next, using the definition of polynomial Q, the following equalities are easy to see:

$$\left(1\ x\ \ldots\ x^N\right) \cdot \left(P_0(y, Z)\ P_1(y, Z) \ldots P_N(y, Z)\right)^\top = Q(x, y, Z) \tag{3}$$

and for each $0 \leq i \leq N$, we have that

$$\left(1\ y\ \ldots\ y^N\right) \cdot \left(P_{i,0}(Z)\ P_{i,1}(y, Z) \ldots P_{i,N}(Z)\right)^\top = P_i(y, Z) \tag{4}$$

Computing Polynomials at Specified Values. For a set $P = \{p_1, p_2, \ldots, p_{k+1}\} \subseteq \mathbb{F}$ of size $k+1$, let $V(P) \in \mathbb{F}^{(k+1)\times(k+1)}$ be the *Vandermonde matrix*, whose entries are given by $V(P)_{ij} = (p_i)^{j-1}$. That is, $V(P)$ looks as follows:

$$V(P) = \begin{pmatrix} p_1^0 & p_1^1 & p_1^2 & \cdots & p_1^k \\ p_2^0 & p_2^1 & p_2^2 & \cdots & p_2^k \\ \vdots & & \ddots & & \vdots \\ p_{k+1}^0 & p_{k+1}^1 & p_{k+1}^2 & \cdots & p_{k+1}^k \end{pmatrix}$$

Note that if P consists of $k+1$ distinct non-zero elements of $\mathbb{F}$, then $V(P)$ is invertible over $\mathbb{F}$. In this case, we let $W(P) = V^{-1}(P)$ be its inverse.

Next, we observe the following:

Let $T = \{y_1, y_2, \ldots, y_{N+1}\}$ be a set of distinct non-zero values of $\mathbb{F}$, then (4) implies that for any $0 \le i \le N$, the following holds:

$$V(T) \cdot \begin{pmatrix} P_{i,0}(Z) \\ P_{i,1}(Z) \\ \vdots \\ P_{i,N}(y, Z) \end{pmatrix} = \begin{pmatrix} P_i(y_1, Z) \\ P_i(y_2, Z) \\ \vdots \\ P_i(y_N, Z) \end{pmatrix} \tag{5}$$

which implies that

$$\begin{pmatrix} P_{i,0}(Z) \\ P_{i,1}(Z) \\ \vdots \\ P_{i,N}(y, Z) \end{pmatrix} = W(T) \begin{pmatrix} P_i(y_1, Z) \\ P_i(y_2, Z) \\ \vdots \\ P_i(y_N, Z) \end{pmatrix} \tag{6}$$

In particular, the polynomials $P_{i,j}(Z)$ at the given values $Z \leftarrow Z'$ can be evaluated in time polynomial in N using the computation above, assuming we can evaluate the polynomial $P_i(y, Z)$ at values $y \leftarrow y'$, and $Z \leftarrow Z'$. Next, we show how to do this computation.

From (3), we get that if $S = \{x_1, x_2, \ldots, x_{N+1}\}$ are distinct non-zero values of $\mathbb{F}$, then:

$$V(S) \cdot \begin{pmatrix} P_0(y, Z) \\ P_1(y, Z) \\ \vdots \\ P_N(y, Z) \end{pmatrix} = \begin{pmatrix} Q(x_1, y, Z) \\ Q(x_2, y, Z) \\ \vdots \\ Q(x_N, y, Z) \end{pmatrix} \implies \begin{pmatrix} P_0(y, Z) \\ P_1(y, Z) \\ \vdots \\ P_N(y, Z) \end{pmatrix} = W(S) \cdot \begin{pmatrix} Q(x_1, y, Z) \\ Q(x_2, y, Z) \\ \vdots \\ Q(x_N, y, Z) \end{pmatrix}$$

In particular, given the values $y \leftarrow y'$, and $z_{ij} \leftarrow z'_{ij}$, where $y', z'_{ij} \in \mathbb{F}$, the polynomials $P_i(y', Z')$ can be evaluated in time polynomial in N, assuming the polynomial $Q(x, y, Z)$ can be evaluated at the specified values $x \leftarrow x'$, $y \leftarrow y'$ and $Z \leftarrow Z'$. However, note that the polynomial Q is equal to the determinant of the matrix A. Thus, this can be implemented in polynomial time.

Recall that we want to determine whether the polynomial $P_{w_r, w_b}(Z)$ is identically equal to zero (cf. Proposition 3). To this end, we sample the values $Z' = \{z'_{ij}\}$ from $\mathbb{F}$ – note that $\mathbb{F}$ contains at least $4(N+1)^2 \ge n^2$ distinct elements, and the degree of the polynomial P_{w_r, w_b} is at most n. Therefore, by Schwartz-Zippel lemma, the probability that the polynomial is non-zero, when evaluated at Z' is equal to zero is at most $n/(N+1) \le 1/n$. Thus, we obtain the following theorem.

Theorem 2. *There exists a randomized algorithm that, given an* Socially Fair Matching *instance on n vertices, and where all edge weights are integers in range $[0, N]$, with $N \ge n$, runs in time $(n+N)^{O(1)}$, and finds an optimal solution with probability at least $1 - 1/n$.*

4.1 FPTAS For General Weights via Reduction to Polynomial Integer Weights.

Let $0 < \epsilon \leq 1$ be a fixed constant. By appropriately scaling, we assume that the smallest positive weight is at least $3/\epsilon$. Then, we round all weights of all the edges up to the nearest integer. Note that the weight of any edge is increased by strictly smaller than 1, which is at most $\epsilon/3$ factor of its original weight. Thus, assume that all weights are non-negative integers. Say, this is preprocessing step A.

By iterating over all edges, we "guess" the largest weight of an edge (after rounding up) that is part of an optimal solution. Let L denote a guess for the largest weight, and note that L is an integer. Then, we delete all the edges with weight larger than L. Suppose $L \leq 2n/\epsilon$. Then, we skip the following preprocessing step B, and directly use Theorem 2 as described subsequently.

Now, suppose that $L > 2n/\epsilon$. Then, for each edge with weight (after preprocessing step A) w, we define its weight to be $\lceil \frac{w}{L/(2n/\epsilon)} \rceil \cdot \frac{L}{2n/\epsilon}$. We say that this is preprocessing step B.

Claim. Suppose we guess the maximum weight L of an edge in an optimal solution correctly. Then, after preprocessing step A, and step B in the iteration corresponding to L, the optimal solution w.r.t. new weights is at most $1+\epsilon$ times the original optimal weight.

Proof. As argued previously, step A incurs at most an $(1+\epsilon/3)$ factor increase in the cost of any solution. Consider the iteration corresponding to L, the maximum weight of an edge in some optimal solution $F \subseteq E$. By removing edges with weight larger than L, we do not delete any edge of an optimal solution. Note that the total increase in the weight of any edge due to step B is at most $\frac{L}{2n/\epsilon}$. Thus, for any set of edges of size at most n, the total increase in the weight is at most $\frac{\epsilon L}{2n} \cdot n \leq \frac{\epsilon L}{2} \leq \frac{\epsilon \cdot OPT}{2}$. Thus, the total increase in the weight due to preprocessing steps A and B can be upper bounded by $(1+\epsilon/3) \cdot (1+\epsilon/2) OPT \leq (1+\epsilon) \cdot OPT$.

After preprocessing step B, the weights are of the form $t \cdot \frac{L}{2n/\epsilon}$, where t is an integer in the range $[0, \lceil n/\epsilon \rceil]$. By dividing each weight by a factor of $L/(2n/\epsilon)$, we obtain an instance where all the weights are integers in the range $[0, \lceil n/\epsilon \rceil]$, i.e., $N = \lceil n/\epsilon \rceil \geq n$. Then, the algorithm from Theorem 2 can be used to find an optimal solution in time $(n/\epsilon)^{O(1)}$, with probability at least $1 - 1/n$. Therefore, we obtain the following theorem.

Theorem 3. *There exists a randomized FPTAS for SOCIALLY FAIR MATCHING for arbitrary weights. In other words, for any $\epsilon > 0$, there exists a randomized algorithm that returns a $(1+\epsilon)$-approximation for SOCIALLY FAIR MATCHING in time $(n/\epsilon)^{O(1)}$, with probability at least $1 - 1/n$.*

4.2 NP-hardness of SOCIALLY FAIR MATCHING

The reduction is from a variant of PARTITION. The input is a set $A = \{a_1, a_2, \ldots, a_n\}$ of n positive integers, and an integer k. The problem asks whether it is possible to partition A into two sets A_1 and A_2 such that the sum of the integers in A_1 and A_2 are equal. It is known that PARTITION is weakly NP-hard, i.e., if the integers in A are given in binary [16].

We reduce this to SOCIALLY FAIR MATCHING as follows. First, let $R = \{r_1, r_2, \ldots, r_n\}$, and $B = \{b_1, b_2, \ldots, b_n\}$ be two disjoint sets of $2n$ vertices. Let $G = (R \cup B)$ be a *complete* bipartite graph, i.e., there is an edge between every r_i and b_j, $1 \le i, j \le n$. Now we define the weights on the edges. For $1 \le i \le n$, set $w(r_i, b_i) = a_i$. For $1 \le i \ne j \le n$, let $w(r_i, b_j) = n \cdot L$, where $L = \sum_{i=1}^{n} a_i$.

Note that any solution of cost at most L must output the matching $M = \{\{r_i, b_i\} : 1 \le i \le n\}$. Restricting our attention to such a solution, now the task is to find an assignment $f : M \to \{\mathsf{red}, \mathsf{blue}\}$. It is easy to see that there is a bijection between an assignment f, and a partition $\{A_1, A_2\}$ of the integers A in the given PARTITION instance. In particular, deciding whether there exists an assignment $f : M \to \{\mathsf{red}, \mathsf{blue}\}$, such that $\mu_{M,f}(R) = \mu_{M,f}(B) = \frac{L}{2}$ is equivalent to determining that the input A of PARTITION can be partitioned into two sets with equal sum. Therefore, finding an optimal solution to SOCIALLY FAIR MATCHING is weakly NP-hard.

5 Conclusions

In this work, we introduce a well-motivated matching problem, namely SOCIALLY FAIR MATCHING, and systemically study the complexity of the problem in terms of exact and approximate computation. Our results draw a nearly complete picture of the computational complexity of the problem. On the one hand, we show that the problem is weakly NP-hard when the edge weights are arbitrary integers. On the other hand, we obtain a randomized polynomial-time algorithm when the weights are polynomially bounded. The latter result leads to a randomized FPTAS for the general problem. We also obtain a deterministic PTAS in the general case, which is a simple, combinatorial algorithm.

Our work leads to several interesting open questions. An obvious question is to obtain a deterministic FPTAS for the problem. Also, it would be interesting to see for which subclasses of graphs our problem admits polynomial-time algorithms. Finally, one might be interested in suitably extending our model to multiple groups.

Bibliography

[1] Abbasi, M., Bhaskara, A., Venkatasubramanian, S.: Fair clustering via equitable group representations. In: Elish, M.C., Isaac, W., Zemel, R.S. (eds.) FAccT '21: 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event / Toronto, Canada, March 3-10, 2021. pp. 504–514. ACM (2021)

[2] Barketau, M., Pesch, E., Shafransky, Y.: Minimizing maximum weight of subsets of a maximum matching in a bipartite graph. Discrete Applied Mathematics **196**, 4–19 (2015)

[3] Berger, A., Bonifaci, V., Grandoni, F., Schäfer, G.: Budgeted matching and budgeted matroid intersection via the gasoline puzzle. Math. Program. **128**(1-2), 355–372 (2011). https://doi.org/10.1007/s10107-009-0307-4, https://doi.org/10.1007/s10107-009-0307-4

[4] Chouldechova, A.: Fair prediction with disparate impact: A study of bias in recidivism prediction instruments. Big data **5**(2), 153–163 (2017)

[5] Corbett-Davies, S., Pierson, E., Feller, A., Goel, S., Huq, A.: Algorithmic decision making and the cost of fairness. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017. pp. 797–806. ACM (2017)

[6] Duginov, O.: Weighted perfect matching with constraints on the total weight of its parts. Journal of Applied and Industrial Mathematics **15**(3), 393–412 (2021)

[7] Dwork, C., Hardt, M., Pitassi, T., Reingold, O., Zemel, R.: Fairness through awareness. In: Proceedings of the 3rd innovations in theoretical computer science conference. pp. 214–226 (2012)

[8] Dwork, C., Ilvento, C.: Group fairness under composition. In: Proceedings of the 2018 Conference on Fairness, Accountability, and Transparency (FAT* 2018) (2018)

[9] Edmonds, J.: Paths, trees, and flowers. Canadian Journal of mathematics **17**, 449–467 (1965)

[10] Feldman, M., Friedler, S.A., Moeller, J., Scheidegger, C., Venkatasubramanian, S.: Certifying and removing disparate impact. In: proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining. pp. 259–268 (2015)

[11] García-Soriano, D., Bonchi, F.: Fair-by-design matching. Data Min. Knowl. Discov. **34**(5), 1291–1335 (2020)

[12] Ghadiri, M., Samadi, S., Vempala, S.S.: Socially fair k-means clustering. In: Elish, M.C., Isaac, W., Zemel, R.S. (eds.) FAccT '21: 2021 ACM Conference on Fairness, Accountability, and Transparency, Virtual Event / Toronto, Canada, March 3-10, 2021. pp. 438–448. ACM (2021)

[13] Huang, C., Kavitha, T., Mehlhorn, K., Michail, D.: Fair matchings and related problems. Algorithmica **74**(3), 1184–1203 (2016)

[14] Kamada, Y., Kojima, F.: Fair matching under constraints: Theory and applications (2020)

[15] Kamishima, T., Akaho, S., Sakuma, J.: Fairness-aware learning through regularization approach. In: Spiliopoulou, M., Wang, H., Cook, D.J., Pei, J., Wang, W., Zaïane, O.R., Wu, X. (eds.) Data Mining Workshops (ICDMW), 2011 IEEE 11th International Conference on, Vancouver, BC, Canada, December 11, 2011. pp. 643–650. IEEE Computer Society (2011)

[16] Karp, R.M.: Reducibility among combinatorial problems. In: Complexity of computer computations, pp. 85–103. Springer (1972)

[17] Klaus, B., Klijn, F.: Procedurally fair and stable matching. Economic Theory **27**(2), 431–447 (2006)

[18] Kleinberg, J., Mullainathan, S., Raghavan, M.: Inherent trade-offs in the fair determination of risk scores. In: 8th Innovations in Theoretical Computer Science Conference (ITCS 2017). Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik (2017)

[19] Kress, D., Meiswinkel, S., Pesch, E.: The partitioning min–max weighted matching problem. European Journal of Operational Research **247**(3), 745–754 (2015)

[20] Makarychev, Y., Vakilian, A.: Approximation algorithms for socially fair clustering. CoRR **abs/2103.02512** (2021)

[21] Papadimitriou, C.H., Yannakakis, M.: On the approximability of trade-offs and optimal access of web sources. In: 41st Annual Symposium on Foundations of Computer Science, FOCS 2000, 12-14 November 2000, Redondo Beach, California, USA. pp. 86–92. IEEE Computer Society (2000). https://doi.org/10.1109/SFCS.2000.892068, https://doi.org/10.1109/SFCS.2000.892068

[22] Thanh, B.L., Ruggieri, S., Turini, F.: k-nn as an implementation of situation testing for discrimination discovery and prevention. In: Apté, C., Ghosh, J., Smyth, P. (eds.) Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Diego, CA, USA, August 21-24, 2011. pp. 502–510. ACM (2011)

[23] Wikipedia contributors: Sister city — Wikipedia, the free encyclopedia (2022), https://en.wikipedia.org/w/index.php?title=Sister_city&oldid=1107517947, [Online; accessed 25-September-2022]