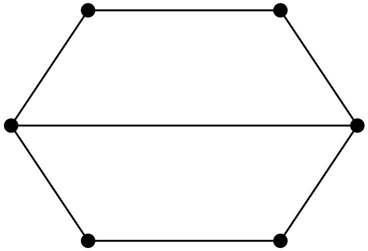
Socially Fair Matching: Exact and Approximation Algorithms

WADS 2023

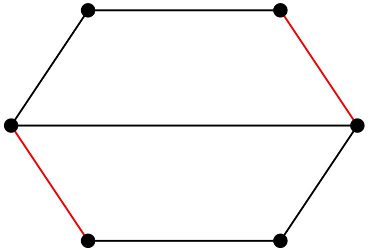
Sayan Bandyapadhyay [1] Fedor V. Fomin [2]
Tanmay Inamdar [2] Fahad Panolan [3] Kirill Simonov [4]

[1]Portland State University, Oregon, USA
[2] University of Bergen, Norway
[3]IIT Hyderabad, India
[4]Hasso Plattner Institute, University of Potsdam, Germany

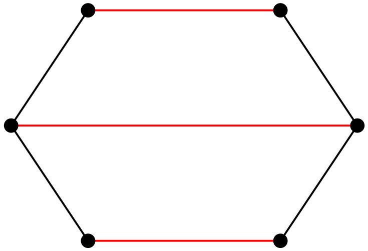
Matching



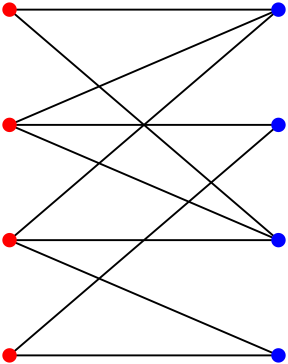
Matching



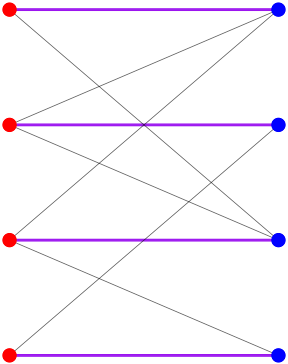
Matching



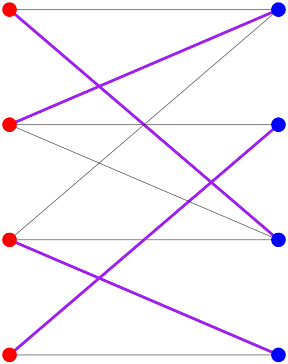
Bipartite Matching



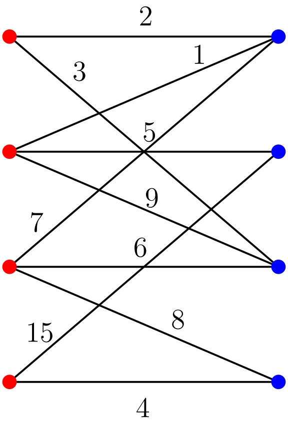
Bipartite Matching



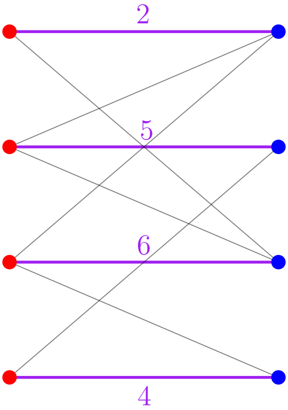
Bipartite Matching



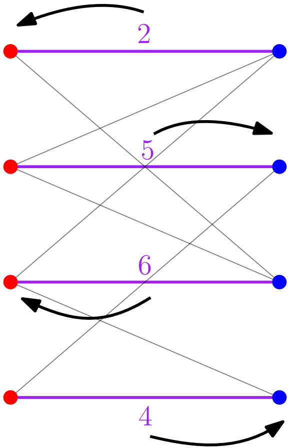
Bipartite Matching



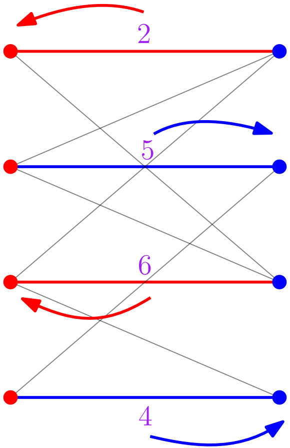
Bipartite Matching



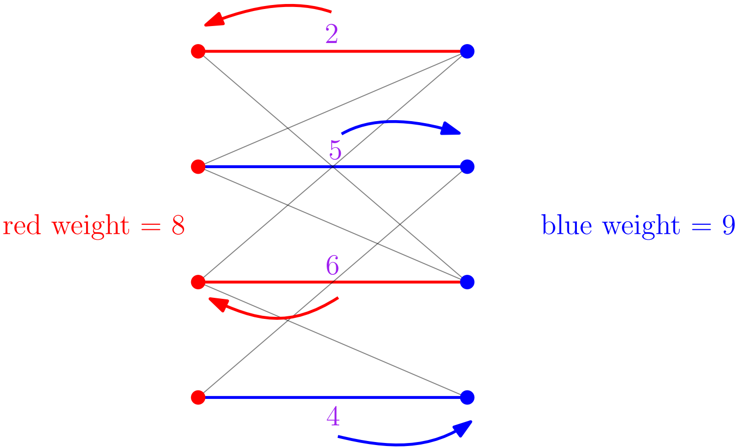
Bipartite Matching



Bipartite Matching



Bipartite Matching



Socially Fair Matching

Socially Fair Matching

Input: Complete bipartite graph $G = (R \uplus B, E)$ and weights $w : E \to \mathbb{R}^+$
Task: Find a perfect matching M and a partition of M into (M_R, M_B)

$$\max \{w(M_R), w(M_B)\}$$

Socially Fair Matching

Socially Fair Matching

Input: Complete bipartite graph $G = (R \uplus B, E)$ and weights $w : E \to \mathbb{R}^+$
Task: Find a perfect matching M and a partition of M into (M_R, M_B)

$$\max \{w(M_R), w(M_B)\}$$

A perfect matching M and a corresponding partition of M into (M_R, M_B) with $w(M_R) = r$ and $w(M_B) = b$ will be called **an** (r, b) **perfect matching**

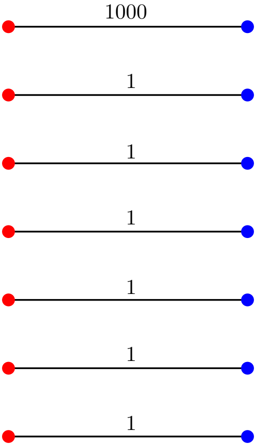
Our Results

- **Deterministic PTAS**: $(1+\epsilon)$-approximation in $n^{O(1/\epsilon)}$ time
- **Randomized** (Monte-Carlo) algorithms:
  - **Polynomial time exact algorithm** when weights are integral and polynomially bounded
  - **Randomized EPTAS** for general weights: $(1+\epsilon)$-approximation in randomized $(n/\epsilon)^{O(1)}$ time

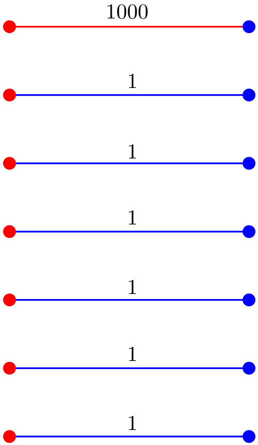
Socially Fair Matching: Starting point

- Goal: find an (r, b) perfect matching with minimum $\max\{r, b\}$
  Minimize maximum of two parts
- So finding minimum-weight perfect matching should be the first step, right?
- Not quite...

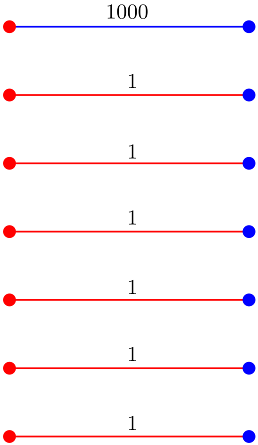
Socially Fair Matching vs Min Weight Perfect Matching



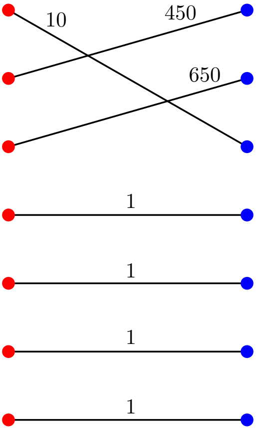
Socially Fair Matching vs Min Weight Perfect Matching



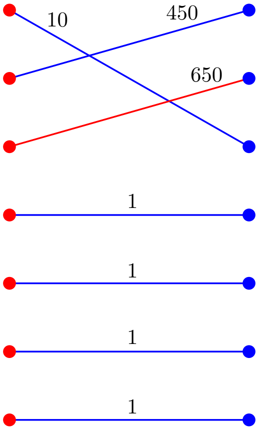
Socially Fair Matching vs Min Weight Perfect Matching



Socially Fair Matching vs Min Weight Perfect Matching



SOCIALLY FAIR MATCHING vs Min Weight Perfect Matching



Socially Fair Matching vs Min Weight Perfect Matching

- Min Weight Perfect Matching is polynomial-time solvable (for arbitrary weights)

Socially Fair Matching vs Min Weight Perfect Matching

- Min Weight Perfect Matching is polynomial-time solvable (for arbitrary weights)
- Previous example immediately suggests reduction from Partition to Socially Fair Matching

Socially Fair Matching vs Min Weight Perfect Matching

- Min Weight Perfect Matching is polynomial-time solvable (for arbitrary weights)
- Previous example immediately suggests reduction from Partition to Socially Fair Matching
  Given integers $\{a_1, a_2, \ldots, a_n\}$ which we want to divide into two subsets of equal sum,
  create a perfect matching with $w(e_i) = a_i$ and set other weights to ∞

Socially Fair Matching vs Min Weight Perfect Matching

- Min Weight Perfect Matching is polynomial-time solvable (for arbitrary weights)
- Previous example immediately suggests reduction from Partition to Socially Fair Matching
  Given integers $\{a_1, a_2, \ldots, a_n\}$ which we want to divide into two subsets of equal sum,
  create a perfect matching with $w(e_i) = a_i$ and set other weights to ∞
- $\implies$ Socially Fair Matching is **weakly NP-hard**
  (i.e., NP-hard if weights are arbitrary)

Socially Fair Matching: Starting point

- Min Weight Perfect Matching may be “bad” if we cannot divide it in a balanced way
- But it already gives a 2-approximation
- If the edge-weights are “small” compared to OPT, then we can balance easily

Socially Fair Matching: Deterministic PTAS

- Heavy edges: weight more than ϵ OPT
  Light edges: weight at most ϵ OPT

Observation

Optimal solution uses no more than $\frac{2}{\epsilon}$ heavy edges

Socially Fair Matching: Deterministic PTAS

- Heavy edges: weight more than ϵ OPT
  Light edges: weight at most ϵ OPT

Observation

Optimal solution uses no more than $\frac{2}{\epsilon}$ heavy edges

Proof.

Suppose not.
Then wlog red side uses more than $1/\epsilon$ heavy edges
Then the total red weight is more than ϵ OPT $\cdot 1/\epsilon =$ OPT, a contradiction. □

Socially Fair Matching: Deterministic PTAS

1. "Guess" at most $\lceil 2/\epsilon \rceil$ heavy edges from the optimal matching: $n^{2/\epsilon}$ guesses

Socially Fair Matching: Deterministic PTAS

1. "Guess" at most $\lceil 2/\epsilon \rceil$ heavy edges from the optimal matching: $n^{2/\epsilon}$ guesses
   Also guess how they are assigned to R or B: $2^{2/\epsilon}$ guesses

Socially Fair Matching: Deterministic PTAS

1. "Guess" at most $\lceil 2/\epsilon \rceil$ heavy edges from the optimal matching: $n^{2/\epsilon}$ guesses
   Also guess how they are assigned to R or B: $2^{2/\epsilon}$ guesses
2. Prune the remaining instance by deleting all matched vertices and heavy edges

Socially Fair Matching: Deterministic PTAS

1. "Guess" at most $\lceil 2/\epsilon \rceil$ heavy edges from the optimal matching: $n^{2/\epsilon}$ guesses
   Also guess how they are assigned to R or B: $2^{2/\epsilon}$ guesses
2. Prune the remaining instance by deleting all matched vertices and heavy edges
3. Remaining bipartite graph only contains light edges
4. Find a minimum-weight perfect matching for remaining vertices containing only light edges

Socially Fair Matching: Deterministic PTAS

1. "Guess" at most $\lceil 2/\epsilon \rceil$ heavy edges from the optimal matching: $n^{2/\epsilon}$ guesses
   Also guess how they are assigned to R or B: $2^{2/\epsilon}$ guesses
2. Prune the remaining instance by deleting all matched vertices and heavy edges
3. Remaining bipartite graph only contains light edges
4. Find a minimum-weight perfect matching for remaining vertices containing only light edges
   which can be approximately balanced to any ratio
5. Results in $(1 + \epsilon)$-approximation

Randomized Exact Algorithm

- Edge-weights are integers from $[0, N]$ for some integer N
- Algebraic tools: DeMillo–Lipton–Schwartz–Zippel Lemma and Polynomial Identity Testing (PIT)
- Reduce the problem to checking whether certain polynomials are not identically zero
- Polynomials are implicitly defined: no explicit representation, but can be efficiently evaluated at given points
- Polynomial $P_{r,b}(x, y, Z)$ is not identically zero iff there exists a perfect matching M that can be partitioned into (M_R, M_B) where $w(M_R) = r$ and $w(M_B) = b$.

Randomized Exact Algorithm

- Let $\mathbb{F}$ be a field of characteristic 2 and let $\mathbb{F}[U]$ be the ring of polynomials where $U = \{x, y, z_{1,1}, z_{1,2}, \ldots, z_{n,n}\}$ is a set of variables
- Define matrix $A = (a_{ij})$ where $a_{ij} = (x^{w_{ij}} + y^{w_{ij}}) \cdot z_{ij}$.
- $\det(A) = Q(x, y, Z)$ is a polynomial in $\mathbb{F}[U]$, where $Z = \{z_{1,1}, \ldots z_{n,n}\}$

$$Q(x, y, Z) = \sum_{r=0}^{N} \sum_{b=0}^{N} x^r y^b \cdot P_{r,b}(Z)$$

- Degree of $Q(x, y, Z)$ is at most $n \cdot N$

Observation

$P_{r,b}(Z)$ is not identically 0 iff there exists a (r, b) perfect matching.

Polynomial Identity Testing

- Goal is to find $0 \leq r, b \leq N$ with minimum $\{r, b\}$ such that, there exists a (r, b) perfect matching
- Equivalently, iterate over all $0 \leq r, b \leq N$ and check whether $P_{r,b}(Z)$ is identically zero
- DLSZ Lemma implies
  If we evaluate $P_{r,b}(Z)$ at randomly sampled $k \gg n^2$ values, with high probability,
  all evaluations are zero iff $P_{r,b}(Z)$ is identically zero
- Each evaluation takes time polynomial in n and N
- Can solve Socially Fair Matching exactly in time $(n+N)^{O(1)}$ with high probability

Randomized EPTAS

- We assumed that weights are integral and from range $[0, N]$
- Reduce general case to this case (heavily inspired from knapsack-type EPTAS)
- Weights can be made integers in the range $[0, n/\epsilon]$, at the expense of $(1+\epsilon)$-loss
- Then solve exactly (via PIT)
- Randomized $(n/\epsilon)^{O(1)}$ time algorithm that computes a $(1+\epsilon)$-approximation with high probability

Summary and Open Problems

- Deterministic PTAS for general weights
  Randomized EPTAS (exact if weights are integer and polynomially bounded)
- NP-hard for general weights (reduction from Partition)
- Can we get deterministic EPTAS?
- Problem naturally extends to 3 colors and more, but algorithms unclear
  3-Dimensional Matching is NP-hard
  Approximations? Beyond polynomial-time?