Algorithms for Optimal Diverse Matching

Saba Ahmadi\textsuperscript{1}, Faez Ahmed\textsuperscript{2}, John P. Dickerson\textsuperscript{1}, Mark Fuge\textsuperscript{2}, Samir Khuller\textsuperscript{3}

\textsuperscript{1} Department of Computer Science, University of Maryland
\textsuperscript{2} Department of Mechanical Engineering, University of Maryland
\textsuperscript{3} Department of Computer Science, Northwestern University
saba@cs.umd.edu, faez00@umd.edu, john@cs.umd.edu, fuge@umd.edu, samir.khuller@northwestern.edu

Abstract

Bipartite $b$-matching, where agents on one side of a market are matched to one or more agents or items on the other, is a classical model that is used in myriad application areas such as healthcare, advertising, education, and general resource allocation. Traditionally, the primary goal of such models is to maximize a linear function of the constituent matches (e.g., linear social welfare maximization) subject to some constraints. Recent work has studied a new goal of balancing whole-match diversity and economic efficiency, where the objective is instead a monotone submodular function over the matching. These more general models are largely NP-hard. In this work, we develop a combinatorial algorithm that constructs provably-optimal diverse $b$-matchings in pseudo-polynomial time. Then, we show how to extend our algorithm to solve new variations of the diverse $b$-matching problem. We then compare directly, on real-world datasets, against the state-of-the-art, quadratic-programming-based approach to solving diverse $b$-matching problems and show that our method outperforms it in both speed and (anytime) solution quality.

1 Introduction

The bipartite matching problem occurs in many applications such as healthcare, advertising, and general resource allocation. Weighted bipartite $b$-matching is a generalization of this problem where each node on one side of the market can be matched to many items from the other side, and where edges may also have associated real-valued weights. Examples of weighted bipartite $b$-matching include assigning schools to children (Drummond et al. 2015), reviewers to manuscripts (Charlin and Zemel 2013; Liu et al. 2014), and donor organs to patients (Bertsimas et al. 2017).

Ahmed et al. (2017) introduced the notion of diverse bipartite $b$-matching, where the goal was to simultaneously maximize the “efficiency” of an assignment along with its “diversity.” For example, a firm might want to hire several highly-skilled workers, but if that firm also cares about diversity it may want to ensure that some of those hires occur across marginalized categories of employees. They proposed an objective which combined economic efficiency and diversity demonstrating that, in practice, reducing the efficiency of a matching by small amounts can often lead to significant gains in diversity across a matching. Their formulation relied on solving a general Mixed-Integer Quadratic Program (MIQP), which is flexible but computationally intractable.

In this work, we study the diverse matching problem but develop an algorithm (and extensions) that is guaranteed to find the global optimum of the diverse, weighted, bipartite $b$-matching problem. Our method is also faster than the approach proposed by Ahmed et al. (2017) on the real-world benchmark test cases that they used. We also propose two practical extensions of the problem. First, where each worker has multiple features (country of origin, gender) and our goal is to form diverse teams for all these features. Second, where each task has a requirement of certain skills and workers are endowed with particular skills, which applies to the crowdsourcing setting.

Our contributions. The paper’s main contributions follow:

- We provide the first pseudo-polynomial time algorithm for the diverse bipartite $b$-matching problem with class-specific weights. The key insight lies in detecting negative cycles in the matching graph, which we use to either provide incremental improvements to the incumbent diverse matching or prove that our negative-cycle-detection algorithms have found a globally-optimal matching.

---

\textsuperscript{1}That is, under conditions when the cost of assigning all items from one category to an item on the other side of the graph is the same. This holds when, e.g., one is matching academic papers to reviewers where each reviewer can specify exactly one field of expertise and the cost of assigning a paper to any of the reviewers within the same field is the same but differs across fields.
We then extend the algorithm to the diverse bipartite b-matching problems with general edge weights.

Next, we define a new variation of diverse b-matching problem where instead of considering only one feature for each worker, here country of citizenship, each worker has a set of features, and the diversity of a b-matching is calculated based on the set of all the features. We show how to extend our algorithm to capture this case as well.

In the next step, we extend our algorithm to a more general version of this problem, where each team has a vector of demands corresponding to a specific skill. Each worker has a specific skill. The goal is to find a diverse b-matching satisfying the vector of demands for each team.

Lastly, we demonstrate our algorithm’s applicability to paper-reviewer matching and movie recommendation. Our algorithm takes less time to converge to an optimal solution than the state-of-the-art MIQP approach.

2 Related Work

Matching people to form diverse teams leverages the intersection of two past areas of research: the role of team diversity in collaborative work and how diversity among groups of resources is measured and used to form/match teams.

Compared to related work, this paper provides a practical, high-performing method to perform diverse b-matching that can enable applications like diverse team formation or diverse resource allocation. Below we will use the example of diverse team formation (for example, in project teams within a company) to provide a concrete example to place prior work in context; however, our proposed approach is generally applicable to any diverse matching problem.

In the example of forming teams, the traditional approach is to use weighted bipartite b-matching (WBM) methods (Basu Roy et al. 2015; Liu et al. 2014b). These methods maximize the total weight of the matching while satisfying the constraints. However, there are two major issues with these approaches. First, it assumes that the value provided by a person in a team is always fixed and independent of who else is in the team. This assumption may not hold in many cases. A new team member may provide more added value to the team if she is added to a smaller team compared to the case if she is added to a larger team. This property of diminishing marginal utility can be mathematically captured by a family of functions called submodular functions.

Second, existing approaches do not account for diversity within a team, where teams with workers from different backgrounds may be desirable. For example, researchers have shown that different types of worker diversity have a direct impact on the success rate of tasks (Ross et al. 2010). Likewise, firms with a higher number of employees with higher education and diversity in the types of educations have a higher likelihood of innovating (Ostergaard et al. 2011) and increasing revenue for firms (Hunt et al. 2013). In this paper, we address both these issues.

Past researchers have generally measured diversity by defining some notion of coverage—that is, a diverse set is one that covers the space of available variation. Mathematically, researchers have done so via the use of submodular functions, which encode the notion of diminishing returns (Lin and Bilmes 2011; Lin and Bilmes 2012)—that is, as one adds items to a set that are similar to previous items, one gains less utility if the existing items in the set already “cover” the characteristics added by that new item. For example, many previous diversity metrics used in the information retrieval or search communities—including Maximum Marginal Relevance (MRR) (Carbonell and Goldstein 1998), absorbing random walks (Zhu et al. 2007), subtopic retrieval (Zhai et al. 2003) and Determinantal Point Processes (Kulesza et al. 2012)—are instances of submodular functions. These functions can model notions of coverage, representation, and diversity (Ahmed and Fuge 2018) and they have been shown to achieve top results on common automatic document summarization benchmarks—e.g., at the Document Understanding Conference (Lin and Bilmes 2011; Lin and Bilmes 2012). These functions are widely used in extractive document summarization (Lin and Bilmes 2011) to get a diverse high-quality summary of documents.

Within matching, our work is closest to Ahmed et al. (2017), which used a supermodular function to propose a diverse matching optimization method. Other researchers have also approached similar problems, with diversity either as an objective or as a constraint. For instance, Götz and Procaccia (2018) match migrants to localities in a way that maximizes the expected number of migrants who find employment. The authors solve a maximization problem of an approximately submodular function subject to matroid constraints. Benabbou et al. (2018) studies the trade-off between diversity and social welfare for Singapore housing allocation. They model the problem as an extension of the classic assignment problem, with additional diversity constraints. Lian et al. (2018) solve the assignment problem when preferences from one side over the other side are given and both sides have capacity constraints. They use order weighted averages to propose a polynomial-time algorithm which leads to high quality and more fair assignments. Agrawal et al. (2018) show that a simple iterative proportional allocation algorithm can be tuned to produce maximum matching with high entropy. Finally, Kobren et al. (2019) proposed two fairness-promoting algorithms for papers reviewer matching problem. They demonstrate that their algorithm achieves higher objective score compared to state of the art matching algorithms that optimize for fairness only. In contrast, our goal is to maximize diversity as an objective along with having constraints on reviewer load.

In contrast to the work above, our approach defines a utility function that can be tuned to balance the diversity and total weight of matching. The diversity function is inspired by the Herfindahl index, which is a statistical measure of concentration and commonly used in economics. We provide a new algorithm that models the problem using an auxiliary graph and uses a heuristic improvement of the negative cycle detection of Bellman-Ford by Goldberg and Radzik (1993).
to find negative cycles and cancel them on a new graph to obtain an optimal solution for the original problem.

### 3 Preliminaries

In this section, we first define the preliminaries for a diverse matching problem, where workers are to be matched to teams and each team wants workers belonging to a diverse set of countries. In our problem, we are given a set of countries $\mathcal{C} = \{C_1, \ldots, C_m\}$ and each country $C_j$ has $|C_j|$ workers inside it. The set of workers is denoted by $X = \{x_1, \ldots, x_n\}$. We wish to form a set of teams $\{T_1, \ldots, T_t\}$ of the workers where each team $T_i$ has a demand of $d_i$, specifying the number of workers that needs to be assigned to it. Each worker can be assigned to exactly one team. The cost of assigning each worker from country $C_j$ to team $T_i$ is denoted by $u_{i,j}$. We assume all costs are integers. The number of workers assigned to team $T_i$ from country $C_j$ is $c_{i,j}$. The total cost of an assignment is $TU = \sum_{i=1}^t \sum_{j=1}^m u_{i,j} \cdot c_{i,j}$. The diversity of an assignment is denoted by $D$ and is equal to $\sum_{i=1}^t \sum_{j=1}^m c_{i,j}^2$. The goal is to minimize the objective function which is equal to $\lambda \cdot D + TU$, where $\lambda > 0$ is a constant. We assume $\lambda$ is a rational number and $\lambda = \frac{\lambda_1}{\lambda_2}$ where $\lambda_1, \lambda_2 \in \mathbb{Z}^+$. We also assume $\lambda_1, \lambda_2$ are constants. Minimizing $\lambda \cdot D + TU$ is equivalent to minimizing $f = \lambda_1 \cdot D + \lambda_2 \cdot TU$ and we use this new objective function when explaining the algorithm.

In this representation, each column of a matrix $F$ is $\text{country, gender}$, for example $F = \{\text{country}, \text{gender}\}$. The goal is to minimize $\sum_{i=1}^t \lambda_2 D_k$, where $D_k$ shows the diversity w.r.t. the feature $f_k$.

In this representation, each column of a matrix $F$ is $\text{country, gender}$, for example $F = \{\text{country}, \text{gender}\}$. The goal is to minimize $\sum_{i=1}^t \lambda_2 D_k$, where $D_k$ shows the diversity w.r.t. the feature $f_k$.

In this representation, each column of a matrix $F$ is $\text{country, gender}$, for example $F = \{\text{country}, \text{gender}\}$. The goal is to minimize $\sum_{i=1}^t \lambda_2 D_k$, where $D_k$ shows the diversity w.r.t. the feature $f_k$.

In this representation, each column of a matrix $F$ is $\text{country, gender}$, for example $F = \{\text{country}, \text{gender}\}$. The goal is to minimize $\sum_{i=1}^t \lambda_2 D_k$, where $D_k$ shows the diversity w.r.t. the feature $f_k$.

### 3.1 Matrix Representation

In this representation, each column corresponds to a country, and each row corresponds to a team. Entry $c_{i,j}$ shows the number of workers from the country $C_j$ assigned to the team $T_i$. We introduce team $T_0$ as a dummy team, and $c_{0,i}$ shows the number of workers from country $C_j$ which are not assigned to any team. An example of matrix representation with three teams and two countries is shown in Fig. 2.

### 3.2 Matching Representation

Here, a bipartite graph $G = (\mathcal{X} \cup \mathcal{T}, E)$ is given. The nodes in $\mathcal{X}$ are corresponding to the workers, and they are partitioned into $m$ subsets $C_1, \ldots, C_m$, which are corresponding to the countries. The nodes in $\mathcal{T}$ are corresponding to the teams. The set of workers from the $i$th team is $T_i$. Entry $F_{i,j}$ shows the number of workers from country $C_j$ assigned to the team $T_i$. The total cost of an assignment is $TU = \sum_{i=1}^t \sum_{j=1}^m u_{i,j} \cdot F_{i,j}$. The diversity of an assignment is $D = \sum_{i=1}^t \sum_{j=1}^m F_{i,j}^2$. The goal is to minimize $\sum_{i=1}^t \lambda_2 D_k$, where $D_k$ shows the diversity w.r.t. the feature $f_k$.

In this representation, each column of a matrix $F$ is $\text{country, gender}$, for example $F = \{\text{country}, \text{gender}\}$. The goal is to minimize $\sum_{i=1}^t \lambda_2 D_k$, where $D_k$ shows the diversity w.r.t. the feature $f_k$.

In this representation, each column of a matrix $F$ is $\text{country, gender}$, for example $F = \{\text{country}, \text{gender}\}$. The goal is to minimize $\sum_{i=1}^t \lambda_2 D_k$, where $D_k$ shows the diversity w.r.t. the feature $f_k$.

In this representation, each column of a matrix $F$ is $\text{country, gender}$, for example $F = \{\text{country}, \text{gender}\}$. The goal is to minimize $\sum_{i=1}^t \lambda_2 D_k$, where $D_k$ shows the diversity w.r.t. the feature $f_k$.

In this representation, each column of a matrix $F$ is $\text{country, gender}$, for example $F = \{\text{country}, \text{gender}\}$. The goal is to minimize $\sum_{i=1}^t \lambda_2 D_k$, where $D_k$ shows the diversity w.r.t. the feature $f_k$.

In this representation, each column of a matrix $F$ is $\text{country, gender}$, for example $F = \{\text{country}, \text{gender}\}$. The goal is to minimize $\sum_{i=1}^t \lambda_2 D_k$, where $D_k$ shows the diversity w.r.t. the feature $f_k$.

In this representation, each column of a matrix $F$ is $\text{country, gender}$, for example $F = \{\text{country}, \text{gender}\}$. The goal is to minimize $\sum_{i=1}^t \lambda_2 D_k$, where $D_k$ shows the diversity w.r.t. the feature $f_k$.

### 3.3 Local Exchange

**Local Exchange**: A local exchange happens when a group of teams decides to transfer one or more workers between each other while maintaining the total number of workers in each of them. The exchange is done in a way that the initial demands of all the teams are fulfilled. Arrows in Fig. 2 show a local exchange using a matrix representation. In this exchange, one worker from $C_2$ is moved from $T_3$ to $T_1$. Two workers from $C_1$ are moved. One is moved from $T_1$ to $T_2$, and the other one is moved from $T_2$ to $T_3$. The set of edges of local exchange in a matrix representation is called a cycle. The source-transitions of a cycle are the cells without any input edges, and the sink-transitions are the cells without any output edges. In Fig. 2, the nodes corresponding to $c_{3,2}$ and $c_{1,1}$ are source-transitions nodes, and the nodes corresponding to $c_{1,2}$ and $c_{3,1}$ are sink-transition nodes.

**Gain of a local exchange**: Our goal is to minimize the objective function $f$, by doing a local exchange. To find out, we first calculate the marginal gain from a given exchange operation which is the difference between the objective values before and after a local exchange. For example, gain of the local exchange in Figure 2 is $\lambda_1 \cdot (c_{3,2} - c_{3,2}^2 - c_{1,2}^2 + (c_1,1 - 1)^2 - c_{1,1}^2 + (c_{3,1} + 1)^2 - c_{3,1}^2) + \lambda_2 \cdot (u_{3,2} + u_{1,2} - u_{1,1} + u_{3,1})$. It can be seen the contribution of the nodes which are not source-transition or sink-transition to the gain of a local exchange is zero. If the net gain is negative, then the local exchange can be considered successful and we can transfer the workers.

### Figure 2: Matrix representation of three teams and workers from two countries. Dummy team $T_0$ accommodates unassigned workers. Arrows represent a local exchange.

### Figure 3: Local exchange operation (in matching representation).
4 Negative-Cycle-Detection-based Algorithms

In this section, we explain our algorithm for finding the optimum assignment. First, we build an auxiliary graph \( G' \). For each team \( T_i \), there is a switch in \( G' \) with \( m \) input ports, and \( m \) output ports, where \( m \) is the number of countries. Each port is a node in \( G' \), and each switch is a directed bipartite graph, with edges going from its input ports (nodes) to its output ports. In Figure 2, each box is a switch. A dummy team \( T_0 \) is introduced to accommodate all unassigned workers in the matching. For each pair of input output ports \((I^i_j, O^i_j)\) in switch \( T_i \), there is a directed edge \( e^i_{j1,j2} \) from \( I^i_j \) to \( O^i_j \), its weight is defined in the following way:

\[
w(e^i_{j1,j2}) = \begin{cases} -2\lambda_1 & \text{if } j_1 = j_2, i \neq 0 \\ 0 & \text{o.w.} \end{cases}
\]

The reason behind putting the weight of edges in the first case equal to \(-2\lambda_1\) is to force the nodes which are not a source-transition or a sink-transition and do not belong to \( T_0 \), to have zero contribution to gain of a cycle.

For each pair of teams \( T_{i1} \) and \( T_{i2} \), where \( i_1 \neq i_2 \), and for each country \( c_j \), there is a directed edge from output port \( O^j_{1} \) of switch \( T_{i1} \) to the input port \( I^j_{i2} \) of switch \( T_{i2} \), and weight of this edge is defined as:

\[
\begin{align*}
\lambda_1 ((c_{i2,j} + 1)^2 - c_{i2,j}^2 + (c_{i,j} - 1)^2 - c_{i1,j}^2) \\
+ \lambda_2 (u_{i2,j} - u_{i1,j}) & \quad \text{if } i_1, i_2 \neq 0 \\
\lambda_1 ((c_{i2,j} + 1)^2 - c_{i2,j}^2) + \lambda_2 (u_{i2,j}) & \quad i_1 = 0 \\
\lambda_1 ((c_{i,j} - 1)^2 - c_{i1,j}^2) + \lambda_2 (-u_{i1,j}) & \quad i_2 = 0
\end{align*}
\]

There are separate cases for \( i_1 \) or \( i_2 = 0 \) since \( T_0 \) is a node corresponding to a dummy team, which is allocated all the unassigned workers from each country. The contribution of this team to the objective function must be zero.

Each cycle in this graph is corresponding to a cycle in a matrix representation and local exchanges along them have the same gain. Figure 4 shows a cycle which is corresponding to the cycles in Figures 2 and 3.

After constructing the auxiliary graph, we run Algorithm 1. Algorithm 1 moves workers from one team to another if it detects a negative cycle. The movement of a worker is always from an output port of a team to an input port of another team.

Algorithm 1 takes as input an initial feasible solution \( M \) as input. To find \( M \), we first find a feasible solution, which satisfies all the demand constraints. In order to find an initial feasible solution, in each iteration, consider the first country \( C_j \) with at least one un-assigned worker, and the first team \( (T_i) \) such that the number of workers assigned to it is less than its demand. In the first iteration, we start with \( C_1, T_1 \), and all the workers are un-assigned). Assign un-assigned workers from \( C_j \) to \( T_i \), until either demand of \( T_i \) is fully satisfied, in this case, move to the next team \((i = i + 1) \), or all the workers from \( C_j \) are assigned, then let \( j = j + 1 \). Repeat this procedure until all the demand constraints are satisfied. Time complexity of this procedure is \( O(m + t) \).

In Algorithm 1, any negative cycle detection algorithm can be used to detect negative cycles in \( G' \). We use a heuristic improvement of Bellman-Ford proposed by Goldberg and Radzik (Goldberg and Radzik 1993) in our experiments.

5 Proof of Optimality

In this section, we prove that Algorithm 1 gives the optimum solution for diverse bipartite \( b \)-matching problem.

Assume after the algorithm ends, the final assignment is a local optimum \( M \), and the optimum solution is \( M^* \). Consider the matching representations of \( M \) and \( M^* \). The symmetric difference of \( M \) and \( M^* \) \((M \pm M^*) \) can be decomposed into a set of alternating cycles and paths of even length. The reason that the length of alternating paths is even is that size of both of the matchings is equal: \(|M| = |M^*| = \sum_{i=1}^t d_i \).

Each local exchange along an alternating cycle is corresponding to a cycle in a matrix representation. A local exchange along an alternating path is corresponding to a cycle in a matrix representation which includes vertices from the row \( T_0 \). For example, Figure 5 shows a local exchange along

Algorithm 1: Find optimal diverse \( b \)-matching

Input: Directed weighted graph \( G' \), initial feasible \( b \)-matching \( M \) which satisfies team demands.

Output: Optimal diverse \( b \)-matching

while \( \exists \) a negative cycle \( C \in G' \) do

// Perform a local exchange operation along \( C \);
for \( e \in C \) do

// Assume edge \( e \) is from output port \( O^j_{1} \) of team \( T_{i1} \) to input port \( I^j_{i2} \) of another team \( T_{i2} \);
// Move one worker of country \( c_j \) from team \( T_{i1} \) to team \( T_{i2} \);
\( c_{i1,j} - 1 = 1; \)
\( c_{i2,j} + 1 = 1; \)
update weight of edges of \( G' \) w.r.t. to the new values of \( c_{i1,j} \) and \( c_{i2,j} \).

Figure 4: Local Exchange in Graph Representation. The edge weights are \( w(e_1) = w(e_5) = 0, w(e_2) = \lambda_1 ((c_{i2,j} + 1)^2 - c_{i2,j}^2 + (c_{i1,j} - 1)^2 - c_{i1,j}^2) + \lambda_2 (u_{i2,j} - u_{i1,j}) \), \( w(e_3) = -2\lambda_1, w(e_4) = \lambda_1 ((c_{i1,j} + 1)^2 - c_{i1,j}^2 + (c_{i,j} - 1)^2 - c_{i,j}^2) + \lambda_2 (u_{i1,j} - u_{i2,j}) \).

Figure 5: Optimal diverse \( b \)-matching.
an alternating path in \( M \oplus M^* \) where red edges belong to \( M^* \), and black edges belong to \( M \). Matrix representation corresponding to this local exchange is shown in Figure 6.

Before proving Thm. 1, we need the following definitions:

**Maximal Cycle:** A cycle \( c \) in the matrix representation is maximal if its source-transitions (nodes with zero incoming edges) and sink-transitions (nodes with zero outgoing edges) are source-transition and sink-transition w.r.t all the edges in \( M \oplus M^* \) as well. For example, consider the cycles in Figure 7. Let’s call the green cycle \( c_g \), the red cycle \( c_r \), and the blue cycle \( c_b \). The \( c_b \) has two source-transitions \( c_{0,2}, c_{0,1} \), and it has two sink-transitions \( c_{0,1}, c_{1,2} \). Since there are no edges going out of \( c_{1,2}, c_{0,1} \), and no edges going into \( c_{0,2}, c_{1,1}, c_g \) is a maximal cycle. Cycle \( c_r \) is not a maximal cycle since \( c_{2,2} \) is a source-transition w.r.t the red edges, but it is not a source-transition w.r.t all edges and a blue edge is going into it. Also \( c_{2,1} \) is a sink-transition w.r.t the red edges, but there is a blue edge going out of it. Cycle \( c_b \) is not a maximal cycle as well.

**Lemma 1.** The set of all the edges of \( M \oplus M^* \) can be decomposed into a set of maximal cycles.

**Proof.** Consider an arbitrary decomposition of the edges of \( M \oplus M^* \) in the matrix representation into a set of cycles \( \{c_1, \ldots, c_r\} \). If there exists a cycle in \( M \oplus M^* \) without any source-transitions and sink-transitions, it means the gain of this cycle is zero and it could be discarded. In that case, there exists any cycle \( c_p \) which is not maximal, then there exists another cycle \( c_q \) which makes \( c_p \) not to be maximal. For example in Figure 7 \( c_r \) is not maximal because of \( c_b \). In this case, union \( c_p \) and \( c_q \), and make \( c_p \cup c_q \) a single cycle in the decomposition. At the end, all the edges in \( M \oplus M^* \) will be decomposed into a set of maximal cycles. Let’s call the set of maximal cycles \( \{c_1', \ldots, c_r'\} \).

For example, in Figure 7 decomposition \( \{c_g, c_r \cup c_b\} \) is a maximal cycle decomposition. Now we are ready to prove the following theorem:

**Theorem 1.** Algorithm 1 finds the global optimum for the diverse b-matching problem.

**Proof.** Let \( f(M) \) show the value of the objective function for the assignment \( M \). \( f(M^*) - f(M) < 0 \) therefore:

\[
    f(M^*) - f(M) = \text{gain}(e_{g,1}) + \text{gain}(e_{g,2}) + \cdots + \text{gain}(e_{g,m}) < 0
\]

Where \( e_k^g (1 \leq k \leq l) \) is the \( k^{th} \) cycle in the maximal cycle decomposition, and \( c_{k,g} \) is applying the local exchange of the cycle \( c_g^* \) at step \( k \). The initial step is the assignment \( M \). Since \( f(M^*) - f(M) < 0 \), there must be a maximal cycle \( c_g^* \) such that \( \text{gain}(e_{g,m}) < 0 \). We wish to show \( \text{gain}(e_{g,1}) < 0 \), which implies starting from the initial assignment \( M \), a local exchange can be done with a negative gain, and \( M \) is not a local optimum which is a contradiction.

Consider \( e_{g,m} \) in a matrix representation. There are four types of vertices in \( e_{g,m}^g \):

- Vertices in the form of \( e_{0,j} \) where \( 1 \leq j \leq m \). These vertices have contribution zero to both \( \text{gain}(e_{g,m}) \) and \( \text{gain}(e_{g,1}) \).
- Vertices that are not sink-transition or source-transition, like \( e_{2,1} \) in Figure 2. It could be seen that contribution of these nodes to both \( \text{gain}(e_{g,m}) \) and \( \text{gain}(e_{g,1}) \) is zero.
- Sink-transitions: Consider an arbitrary sink-transition \( v \) in \( e_{g,m}^g \). Assume the value of this node at the beginning of step \( m \) is \( v_g \). The contribution of \( v \) to \( \text{gain}(e_{g,m}) \) is positive and is equal to \( \lambda_1((v_g - 1)^2 - v^2_g) + \lambda_2 u \) for some cost value \( u \). Since \( v \) is a sink-transition and there are no edges out of \( v \), \( v^2_g \geq v_1 \). Therefore, \( \lambda_1((v_g - 1)^2 - v^2_g) + \lambda_2 u \geq \lambda_1(v_1 - 1)^2 - v^2_1 + \lambda_2 u \). As a result, the contribution of \( v \) to \( \text{gain}(e_{g,1}) \) is upper bounded by its contribution to \( \text{gain}(e_{g,m}) \).

- Source-transitions: Consider an arbitrary source-transition \( v \) in \( e_{g,m}^g \). The contribution of \( v \) to \( \text{gain}(e_{g,m}) \) is \( \lambda_1((v_g - 1)^2 - v^2_g) - \lambda_2 u \). \( v \) is a source-transition and therefore \( v_1 \geq v_g \). As a result, \( \lambda_1((v_g - 1)^2 - v^2_g) - \lambda_2 u \geq \lambda_1((v_1 - 1)^2 - v^2_1) - \lambda_2 u \). At the end, contribution of all the vertices to \( \text{gain}(e_{g,1}) \) is upper bounded by their contribution to \( \text{gain}(e_{g,m}) \). Therefore if \( \text{gain}(e_{g,m}) < 0 \), then \( \text{gain}(e_{g,1}) < 0 \).

**Theorem 2.** The running time of the algorithm is \( O((\lambda_1 \cdot n^2 + \lambda_2 U) \cdot m \cdot t^2(m + t)) \), where \( U \) is the maximum cost of an initial feasible b-matching.

In order to prove this theorem, first we show the following lemmas hold.

**Lemma 2.** The number of iterations of our algorithm is at most \( (\lambda_1 \cdot n^2 + \lambda_2 U) \).

**Proof.** The initial state of the algorithm is a feasible b-matching with cost at most \( U \). Diversity of any matching is at most \( n^2 \). Therefore, the maximum value of the objective function is at most \( \lambda_1 \cdot n^2 + \lambda_2 U \). At each iteration, we find a negative weight cycle and since all the weights are integers,
its weight can be at most \(-1\). Therefore, the objective function decreases by at least 1 at each step, and since the value of the objective function is always positive, the number of iterations is at most \(\lambda_1 \cdot n^2 + \lambda_2 U\).

**Lemma 3.** The complexity of each iteration of the algorithm is \(O(m^2 \cdot t^2(\lambda_1 \cdot n^2 + \lambda_2 U))\).

**Proof.** At each iteration, we use a negative cycle detection algorithm with running time \(O(\left| V \right| \cdot |E|)\) (where \(|V|\) is the number of nodes in the auxiliary graph and \(|E|\) is the number of edges). The number of nodes in the graph is \(2m \cdot (t + 1)\), since there are \(t + 1\) switches in the graph and each switch has exactly \(2m\) ports and each node is a port in the graph. The number of edges incident on each port is \(O(t \cdot (m + t))\). Therefore, the total number of edges is \(O(m \cdot t(m + t))\). Hence, the complexity of each iteration is \(O(m^2 \cdot t^2(\lambda_1 \cdot n^2 + \lambda_2 U))\).

Combining Lemma 2 with Lemma 3, and considering \(O(m + t)\) time complexity for finding an initial feasible solution, yields Theorem 2.

6  Diverse Weighted Bipartite b-Matching

In this section, we extend our algorithm to solve the case where cost of assigning workers from the same country to a team can be different. First, in each switch we put input and output ports for each worker. Inside each switch, there is a complete bipartite graph from input ports to the output ports. If there is an edge between an input port to an output port where ports are belonging to the same country, its weight is \(-2\lambda_1\). Otherwise, its weight is 0.

Consider an edge from output port \(x^i_k\) of switch \(T_{i_1}\) to the input port \(x^j_{i_2}\) of switch \(T_{i_2}\), where \(x_k \in C_i\). The weight of this edge is equal to the change in the objective function by moving one worker from \(C_j\) out of \(T_{i_1}\), and adding that worker to \(T_{i_2}\). Proof of the following theorem is similar to Theorem 2.

**Theorem 3.** The running time of the algorithm for general weights is \(O((\lambda_1 \cdot n^2 + \lambda_2 U) \cdot n^2 \cdot t^2(n + t))\), where \(U\) is the maximum cost of any feasible b-matching.

7  Workers with Multiple Features

In this section, we consider the diverse matching problem, however, instead of having only one feature for each worker (country of origin), there is a vector of features. An example of a feature set could be country of citizenship and gender. Let \(F = \{f_1, \ldots, f_F\}\) denote the feature set for the workers. In this case, the goal is to minimize \(\lambda_{f_k} = \sum_{k=1}^{|F|} \lambda_k D_k\), where \(D_k\) shows the diversity w.r.t. the feature \(f_k\).

In the following, we show how to extend our framework to solve this problem. Assume the workers have two features \(X, Y\), where \(X\) could get any of the values \(X_1, X_2, \ldots, \) and \(Y\) could get any of the values \(Y_1, Y_2, \ldots\). In the graph representation, there is a switch per each team, and each switch has an input port and an output port per each possible combination of features (in this example the possible combinations of features are \(\{(X_1, Y_1), (X_2, Y_1), (X_1, Y_2), (X_2, Y_2)\}\)). Between two switches \(T_{i_1}, T_{i_2}\), there is a directed edge from an output port of \(T_{i_1}\), to an input port of \(T_{i_2}\), if and only if the ports are corresponding to the same combination of features. The weight of this edge denotes the change in the objective function corresponding to one exchange.

Inside a switch \(T_i\), there is a directed edge from each input port to each output port. If the directed edge is connecting two ports such that their corresponding combinations of features have no value in common, the weight of this edge is equal to 0. Otherwise, per each feature \(f_k\) that has the same value, \(-2\lambda_1\) is added to this edge. For example, if there is an edge in \(T_i\) from the input port corresponding to \((X_1, Y_1)\) to the output port \((X_1, Y_2)\), the weight of this edge is equal to \(-2\lambda_1\). By applying a similar negative cycle detection algorithm, the optimum solution could be found. In the supplemental material, a sketch of the proof of optimality of the algorithm is given.

8  Skilled Workers

In this section, we consider a variant of the diverse matching problem with additional assumptions that each worker has a specific skill. Each team has a demand vector, denoting how many workers are needed for each skill. The set of skills is shown with \(S = \{s_1, s_2, \ldots, s_S\}\). \(d_{i,k}\) denotes demand of team \(T_i\) for skill \(s_k\). The overall demand of \(T_i\) is \(\sum_{k=1}^{|S|} d_{i,k}\). The goal is to minimize the same objective function \(\lambda_1 \cdot U + \lambda_2 T U\) while satisfying all the demands.

To extend our algorithm to capture this problem, we put a switch per team, and there are I/O ports per each pair of country and skill. There exists an edge between two ports if and only if the ports are corresponding to the same skill. If two ports in a switch corresponding to the same skill also correspond to the same country, the edge between them has weight \(-2\lambda_1\), otherwise, it has weight 0.

Between two switches \(T_{i_1}\) and \(T_{i_2}\), there exists an edge between an output port of \(T_{i_1}\) to an input port of \(T_{i_2}\), if and only if the ports are corresponding to the same team and skill. Consider an edge from an output port \((C_i, S_k)\) in switch \(T_{i_1}\) to an input port \((C_j, S_k)\) in switch \(T_{i_2}\). Weight of this edge is the change in the objective function by moving one worker from country \(C_j\) in team \(T_{i_1}\) to team \(T_{i_2}\).

9  Experimental Validation & Discussion

To demonstrate the efficacy of the proposed method, we apply it to two domains: matching movies to users and matching reviewers to papers. First, we show the trade-off front between diversity and the total weight of matching for the reviewer assignment problem. Next, we compare our algorithm with existing methods in the literature and show that it outperforms them in terms of time taken to converge.

**Application to Reviewer Assignment**

We now present an application of Algorithm 1 for diverse matching problem to automatically determine the most appropriate reviewers for a manuscript by also ensuring that reviewers are different from each other.
We use the multi-aspect review assignment evaluation dataset (Karimzadehgan and Zhai 2009), a benchmark dataset from UIUC. It contains 73 papers accepted by SIGIR 2007, and 189 prospective reviewers who had published in the main information retrieval conferences. The dataset provides 25 major topics and for each paper in the set, an expert provided 25-dimensional label on that paper based on a set of defined topics. Similarly for the 189 reviewers, a 25-dimensional expertise representation is provided.

To set up the graph, we first cluster the reviewers into 5 clusters based on their topic vectors using spectral clustering. To calculate the relevance of each cluster for any paper, we take the average cosine similarity of label vectors of reviewers in that cluster and the paper. We set the constraints such that each paper matches with exactly 4 reviewers, and no reviewer is allocated more than 1 paper. We need to double the number of reviewers in each cluster to make sure the total demand of the papers could be satisfied.

We first find a minimum cost (maximum weighted) matching for this problem using the Gurobi Integer Linear Program solver (ILP for finding a maximum weighted bipartite matching is explained in the supplementary material). The resultant matching is found to be non-diverse. All 73 papers are allocated three reviewers who are all from the same cluster. This gives the resultant matching zero average diversity, as measured by Shannon entropy. Using the graph algorithm, we show that by varying $\lambda$, we can obtain the trade-off between cost and diversity. Figure 8 shows the trade-off front between average Shannon entropy and the total weight of the matching for different values of $\lambda$. For this problem, once $\lambda$ is greater than 0.26, all matchings are maximum diversity matching and they result in the same matching allocation.

The trade-off front allows us to understand the impact of diversity on the total cost of the matching for any given domain. For instance, in Fig. 8 by marginally increasing $\lambda$ above 0, we see a large gain in entropy with little loss in the total cost of the matching. This shows that we can greatly increase diversity at little cost to matching efficiency. In the subsequent sections, we set $\lambda = 1$ fixed.

To demonstrate the efficacy of our algorithm for the case when reviewers have multiple features (like cluster assignment and gender), we ran additional experiments with a modified version of this dataset (by randomly adding genders to each reviewer). We showed that our method discussed in Section 7 finds the optimum solution. We further validated it against MIQP formulation of the same problem, which is discussed in the supplementary material.

Application to MovieLens Data

In this section, we compare our negative cycle detection algorithm for the first variation of diverse matching that we discussed, with the MIQP approach in (Ahmed et al. 2017) with increasing size of the graph. This example considers matching movies to users while ensuring that the movies contain diverse genres. We use a subset of the MovieLens 1M dataset (Harper and Konstan 2016), which includes one million ratings by 6,040 users for 3,900 items. This dataset contains both users’ movie ratings between 1 and 5 and genre categories for each movie (e.g., comedy, romance, and action). We first train a standard collaborative recommender system (Bradley 2016) to obtain ratings for all movies by every user. We cluster the movies into 5 clusters using their vector of 18 genres using spectral clustering so that each movie gets a unique cluster label.

We solve the diverse matching problem using both negative cycle detection algorithm, and MIQP approaches for 1500 movies and match each user with three movies. The number of users is increased in steps of 50 to compare the timing performance of our approach to MIQP. For MIQP, we set a maximum run time of two hours (7200 seconds), at which we report the current best MIQP solution.

Table [1:2:1] shows that for all cases with the number of movies greater than 75, MIQP does not converge within two hours, while our method finds the optimum solution in lesser time. Interestingly, MIQP current solutions are found to be the same as the optimum solution found by our method. This shows that for this application, MIQP was able to search the solution but it was not able to prove that the solution is optimum. In contrast, our method finds the solution faster as well as guarantees that it is optimum.

<table>
<thead>
<tr>
<th>Number of movies</th>
<th>MIQP Time (s)</th>
<th>Our Method Time (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>50</td>
<td>133</td>
<td>33</td>
</tr>
<tr>
<td>75</td>
<td>7200</td>
<td>98</td>
</tr>
<tr>
<td>100</td>
<td>7200</td>
<td>247</td>
</tr>
<tr>
<td>125</td>
<td>7200</td>
<td>663</td>
</tr>
<tr>
<td>150</td>
<td>7200</td>
<td>1069</td>
</tr>
<tr>
<td>175</td>
<td>7200</td>
<td>2581</td>
</tr>
<tr>
<td>200</td>
<td>7200</td>
<td>2316</td>
</tr>
<tr>
<td>250</td>
<td>7200</td>
<td>4609</td>
</tr>
</tbody>
</table>

Table 1: Comparison of MIQP and our method for MovieLens.

10 Conclusion & Future Research

In this paper, we proposed the first pseudo-polynomial time algorithms for diverse weighted bipartite $b$-matching. We showed that our algorithms not only guarantee optimal solutions but also converges faster than the existing state-of-the-art approach using a black-box industrial MIQP solver. We demonstrated our results on two datasets for paper reviewer matching and movie recommendation. Future work could explore the extension of this
method to online diverse matching (Dickerson et al. 2019), where vertices arrive sequentially and must match immediately; this has direct application in advertising, where one could balance notions of reach, frequency, and immediate monetary return. Exploring connections to fairness in machine learning (Grgić-Hlača et al. 2018) and hiring (Schumann et al. 2019) by way of diversity are also of immediate interest.

References


[Ross et al. 2010] Joel Ross, Lilly Irani, M Silberman, Andrew Zaldivar, and Bill Tomlinson. Who are the crowdworkers?: shifting


Proof Sketch. In the matrix representation for this model, there is column per each possible set of features. In the example we mentioned in Fig. 2 we put a column per each element in the set \( \{(X_1, Y_1), (X_1, Y_2), (X_2, Y_2), (X_2, Y_2)\} \). First, it can be shown that a maximal cycle decomposition w.r.t all the features could be found in the matrix representation. The procedure is as following, while there exists a cycle \( c_p \), which is not maximal w.r.t at least one feature, assuming \( c_q \) is preventing \( c_p \) from being maximal, union \( c_p \) and \( c_q \) and report them as one cycle in the cycle decomposition. Same as what we did in the proof of Theorem 1 it can be shown there exists a maximal cycle \( c_g \) such that if a local exchange is done along it in step \( g \), \( \text{gain}(c_g) < 0 \).

Given a cycle \( c \), let \( D(\text{gain}(c)) \) denote the diversity part of \( \text{gain}(c) \), and \( U(\text{gain}(c)) \) denote its cost part, which means \( \text{gain}(c) = D(\text{gain}(c)) + U(\text{gain}(c)) \).

Then \( D(\text{gain}(c_g)) = \sum_{k=1}^{|F|} D(\text{gain}_k(c_g)) \). Where \( \text{gain}_k(c_g) \) is the gain of cycle \( c_g \) in step \( g \) only w.r.t the feature \( f_k \). Additionally, since \( c_g \) is maximal w.r.t to all the features, \( D(\text{gain}_k(c_g)) \leq D(\text{gain}_k(c_g)) \). Therefore:

\[
D(\text{gain}(c_g)) = \sum_{k=1}^{|F|} D(\text{gain}_k(c_g)) \geq \sum_{k=1}^{|F|} D(\text{gain}_k(c_g))
\]

Since \( U(\text{gain}(c_g)) = U(\text{gain}(c_g)) \), therefore \( \text{gain}(c_g) \geq \text{gain}(c_g) \). Following proof of Theorem 1, it can be concluded that the algorithm finds the global optimum.

### B Weighted Bipartite Matching (WBM) Model

Weighted bipartite \( b \)-matching is a combinatorial optimization problem in which the goal is to maximize the total weight of the matching. Its optimal solution will maximize the weight of a matching, emphasizing only on efficiency and neglecting diversity. We formulate the problem as follows. Given a weighted bipartite graph \( G = (U, V, E) \) with weights \( W : E \rightarrow R^+ \), where \( U, V \) and \( E \) represent left vertices, right vertices and edges between them, the weighted bipartite \( b \)-matching problem is to find a sub-graph \( T \subset G \) such that each vertex \( i \) in \( T \) has at most \( b \) edges (i.e., a degree constraint). One can formulate the WBM to maximize or minimize the objective function depending on the application.

For the problem discussed in Section 3 we use a similar notation to [Ahmed et al. 2017] for defining a weighted bipartite \( b \)-matching minimization problem with node-specific upper cardinality constraints. Our goal is to match teams with workers, where each worker belongs to a country and workers within a country are exchangeable (country is used as a representative term to explain the algorithm and the methods generally apply to any form of group allocation). There are \( t \) teams on the right side of the bipartite graph and \( m \) countries on the left side of the graph. The weighted bipartite \( b \)-matching (WBM) problem can be expressed as follows.

\[
\begin{align*}
\max_y & \quad \sum_{i=1}^{t} \sum_{j=1}^{m} w_{i,j}y_{i,j} \\
\text{s.t.} & \quad \sum_{j=1}^{m} y_{i,j} = d_i \quad \forall i \in \{1, \ldots, t\} \\
& \quad \sum_{i=1}^{t} y_{i,j} \leq cd_j \quad \forall j \in \{1, \ldots, m\}
\end{align*}
\]

There are \( m \) countries on the left side of the bipartite graph and \( t \) teams on the right side. \( y_{i,j} \) is a positive integer value, denoting how many workers are allocated in the solution from country \( C_j \) to team \( T_i \). \( w_{i,j} \) is the value provided by a worker from country \( C_j \) to team \( T_i \) (weight of the edges in the bipartite graph). \( cd_j \) is the maximum number of workers belonging to a country \( C_j \). The second constraint ensures that the total allocation from each country should be below this limit. \( d_i \) is the number of workers needed in team \( T_i \). The first constraint ensures that the demand of each team should be exactly met. The above formulation maximizes the total weight of the matching without taking into account the diversity.

### C Diverse Weighted Bipartite Matching Model

In this section, we define the Diverse Weighted Bipartite Matching (D-WBM) objective function for a node on the right side of a bipartite graph. We assume that nodes on left side of a graph belong to \( m \) countries and our goal is to obtain even coverage over all countries (maximize diversity) as well as match a task to workers which provide high weight (low cost). We define an objective function to measure the weight and diversity for a team \( T_i \) on the right side of the graph as:

\[
f(T_i) = \sum_{j=1}^{m} u_{i,j}|S_{T_i,C_j}| + \lambda_c \sum_{j=1}^{m} |S_{T_i,C_j}|^2
\]

\( S_{T_i,C_j} \) denotes the set of workers from country \( C_j \) assigned to the team \( T_i \), \( \lambda_c \) is the diversity knob, which controls how much diversity is needed in the matching. \( u_{i,j} \) is the cost of assigning a worker from \( C_j \) to \( T_i \) (which we calculate as the negative of weight provided by the edge, i.e. \( u_{i,j} = -w_{i,j} \)). The function gives lower cost to solutions with even coverage over all countries. By minimizing this function, we can maximize the weight and diversity of the matching for team \( T_i \). If \( \lambda_c \) approaches zero, this objective function reduces to the WBM problem.

We define the optimization problem for Diverse WBM as:

\[
\begin{align*}
\min_y & \quad \sum_{i=1}^{t} \sum_{j=1}^{m} -w_{i,j}y_{i,j} \\
& \quad + \lambda_c \sum_{i=1}^{t} \sum_{j=1}^{m} y_{i,j}^2 \\
\text{s.t.} & \quad \sum_{j=1}^{m} y_{i,j} = d_i \quad \forall i \in \{1, \ldots, t\} \\
& \quad \sum_{i=1}^{t} y_{i,j} \leq cd_j \quad \forall j \in \{1, \ldots, m\}
\end{align*}
\]

Similar to the WBM problem, there are knapsack constraints on each left side node of the bipartite graph, which means each country has a maximum number of workers and
the allocation cannot exceed that limit. Each team also has a
demand of $d_i$ workers. The first part of the objective function
measures the cost of matching and the second part measures
the homogeneity. Hence, minimizing Eq. [3] leads to a diverse
and high quality solution. Unlike the formulation proposed
in (Ahmed et al. 2017), our formulation provides the flexi-
bility to use $\lambda$ as a diversity knob, to increase or decrease
the amount of diversity required for each application.

D Experimental Validation for Workers with
Multiple Features

In this section, we discuss the importance of workers with
multiple features. In Section 9, we demonstrated how di-
verse matching methods can help obtain solutions for the pa-
paper reviewer assignment problem. Each reviewer belonged
to one of five clusters and the goal was to match a paper
with reviewers from different clusters. By varying a diver-
sity parameter, we obtain the entire trade-off front between
diversity and total cost of the matching.

In this section, we solve a different variant of the prob-
lem. We assume that each reviewer has more than one fea-
ture (for example, country and gender). We want to match
each paper with reviewers who are not only from different
countries (clusters defined originally), but also belong to dif-
ferent genders.

Mixed Integer Quadratic Formulation

$$f(T_i) = \sum_{j=1}^{m} u_{i,j} |S_{T_i, c_j}| + \lambda_c \sum_{j=1}^{m} |S_{T_i, c_j}|^2 + \lambda_g \sum_{g=1}^{G} \sum_{j=1}^{m} |S_{T_i, c_j, g}|^2$$

(4)

$\lambda_g$ controls how much weight we assign to gender di-
versity and $\lambda_c$ is the weight assigned to country diversity.
$|S_{T_i, c_j, g}|$ is the number of workers from country $j$ and
gender $g$ assigned to team $T_i$. The objective function is equal
to $\sum_{i=1}^{n} f(T_i)$.

To demonstrate our algorithm, we solve this problem us-
b ing both MIQP and our auxiliary graph method. We demon-
strate our method on the modified UIUC dataset used in Sec-
section 9. We randomly allocate genders to each reviewer, such
that 49.2% workers are male and the rest are female. We use
$\lambda_c = \lambda_g = 1$ for our experiments.

We run the negative cycle detection algorithm, and the
MIQP solver using Gurobi (to minimize the objective func-
tion discussed in Eq. [4], and both find the optimum solu-
tion. In the optimum solution, all 73 papers receive two male
reviewers and two female reviewers, which shows that the
method was capable of balancing gender diversity. If we
only optimize for country diversity, it is possible that the
gender ratio for individual paper gets skewed. When we run
the same model with $\lambda_g = 0$ (no weight to gender diversity),
we find that out of 73 papers, 12 papers receive all four re-
viewers of the same gender and 41 papers receive three re-
viewers of the same gender. Hence, only 27.3% teams of re-
viewers are gender balanced. However, one should note that
when we do not keep gender as an objective, the resultant al-
location is random and different skewness can be observed
in different runs based on the initial solution.