Finding Teams of Maximum Mutual Respect

Sofia Maria Nikolakaki¹, Evaggelia Pitoura², Panayiotis Tsaparas², Evimaria Terzi¹

¹ Computer Science Department, Boston University, Boston, MA, USA
{smnikol, evimaria}@bu.edu
² Computer Science and Engineering Department, University of Ioannina, Ioannina, Greece
{pitoura, tsap}@cse.uoi.gr

Abstract—Teams that bring together experts with different expertise are important for solving complex problems. However, research shows that teaming up people simply based on their ability is not enough. Team members need to have clear roles, and they should mutually endorse and respect their teammates for the role they assume on the team. In this paper, we define the MAXMUTUALRESPECT problem, a novel team-formation problem that asks for a set of experts, each assigned to a distinct role, such that the total respect that the individuals receive by the rest of the team members for their assigned role is maximized. We show that the problem is NP-hard and we consider approximation and heuristic algorithms. Experiments with real datasets demonstrate that our problem definitions and algorithms work well in practice and yield intuitive results.

Index Terms—mutual respect, team formation, graphs

I. INTRODUCTION

Teams that bring together experts with different expertise for different roles are essential for solving complex problems that are too hard to be tackled by individuals. However, teaming up people simply based on their expertise level is not enough. Articles in research literature [1], [2] and in popular press¹² indicate that dynamics between the members of the team are equally important for the success of the team. In particular, they support that a successful team requires clearly defined roles and responsibilities for each team member, and mutual respect between the team members for their respective roles.

The problem of creating a team of experts while taking into account the relationships between the team members was first formulated in [3]. In that work they assume a set of experts, each associated with a set of skills, organized in a network capturing their ability to collaborate and communicate. The goal is to identify a subset of experts that collectively have the skills for a given task, while they induce a subgraph with low communication cost. There has been considerable follow-up work that considers different variants of this problem [4]–[8].

All prior work assumes that the dynamics in the team are captured by a single undirected graph that represents the overall compatibility between team members. However, an equally important aspect in teams is the level of respect each member enjoys for the specific role assigned to them. Respect between individuals has distinctive characteristics. First, it depends on the role. For instance, in the academic domain, an expert in artificial intelligence may be respected by her peers for her abilities in this field, but she may not be (equally) respected for her abilities in mobile computing, or databases. Second, respect is clearly a directed relationship. For example, it is not reasonable to assume that the degree of respect that a graduate student has for a senior professor is equally reciprocated. Existing work on team formation does not account for such role specialized and asymmetric relationships.

Motivated by these considerations we formulate the novel MAXMUTUALRESPECT problem that asks for a team of experts, each associated with a distinct role such that the total respect that these experts receive by the other team members with respect to their associated role is maximized. In our setting, we have a set of roles that need to be filled, and every role is associated with a distinct directed network over the set of experts that we refer to as the respect graph. An edge $(u, v)$ in the respect graph of role $r$ denotes that $u$ respects and endorses $v$ for the role $r$. Our goal is to create a team of experts that assigns an expert to each role, such that the incoming edges to the designated experts in the corresponding respect graphs, by their teammates is maximized. We study the problem theoretically and experimentally, and we make the following contributions:

- To the best of our knowledge we are the first to formally define and study the novel team-formation problem MAXMUTUALRESPECT which aims to find a team of experts that maximizes the total respect.
- We show that MAXMUTUALRESPECT is NP-complete and design heuristic algorithms for solving it in practice. For the variant of the problem where each respect graph is derived from a ranking of the experts, we design a polynomial algorithm for finding a team with maximum respect, if such a team exists, as well as approximation algorithms that rely on the properties of rankings.
- Our experiments on two real case studies demonstrate that our problem definitions, as well as our algorithms perform well in practice and yield useful and intuitive results.

II. RELATED WORK

Recent studies raise the importance of team formation in different settings [9], [10]. To the best of our knowledge, we are the first to introduce the MAXMUTUALRESPECT problem that takes into account the endorsement that individuals receive with respect to specific skills required for accomplishing a
specific task. Our work is related to a lot of existing work in team formation, rank aggregation and endorsement deduction. Below, we review this work and comment on how it relates to ours.

The importance of trust and respect in teams has been studied in business literature [1], [2]. Their focus is mainly on explaining why these are primary factors in team formation. Our work uses these observations to formally define the problem of creating a team with high respect.

The team-formation problem defined in [3] is the following: Given a set of experts organized in a network, where each individual is associated with a set of skills, identify a subset of experts that together cover the skills required for completing a task, while at the same time they induce a subgraph with low communication cost. Different variants of the problem consider different notions of communication cost [4]–[8], team-design criteria [11], [12], or task-arrival process [13]. There are three key differences between our work and that of [3]: First, prior work assumes that the expert network is undirected, defined by reciprocal relationships between the experts, while our model assumes directed relationships. Second, we assume a different network for each different role. Finally, in prior work team formation is modeled as a coverage problem, where the goal is to cover the set of skills for the task, while we have an assignment problem where the goal is to assign an expert to each role. These three factors make the problem considered in this work fundamentally different from existing literature.

The variant of the problem where endorsements come in the form of a ranking bears some similarity with the rank aggregation problem [14]–[18]. However, in rank aggregation the goal is to produce a single consensus ranking from the input rankings. In our case, the goal is not to create a ranking of the experts but rather to assign them to specific roles based on the selected team’s consensus.

Finally, there is work on deducing endorsement relations in social networks [19], [20]. Here, we assume that the endorsement graphs are given as inputs. Creating these graphs is out of the scope of this work.

III. Preliminaries

We are given a set of experts $V$, and a set of roles $S$. Every role $i \in S$ is associated with a directed graph $G_i = (V, E_i)$ over the set of experts. A directed edge $(u, v) \in E_i$ denotes that $u$ respects and endorses $v$ for the role $i$. We refer to $G_i$ as the respect graph for role $i$. Our goal is to create a team of experts $F \subseteq V$ such that each role is assigned an expert, and the assigned expert enjoys the respect of as many of the other members in the team as possible for this role.

To formalize this idea, we define a role assignment as a function $f : S \rightarrow V$, where expert $f(i)$ is assigned to role $i \in S$. We assume that the function $f$ is injective, that is, each team member can only be used for a single role. Let $F = f(S)$ denote the selected team of experts. The respect $R_i(f)$ that expert $f(i)$ receives with respect to her role from the selected team is defined as $R_i(f) = \{|(u, f(i)) \in E_i : u \in F, u \neq f(i)\}$, that is, the number of incoming edges in graph $G_i$ from the other team members. The total respect score of the team is defined as the sum of the respect values over all roles: $R(f) = \sum_{i \in S} R_i(f)$.

We can now define the MaxMutualRespect problem.

**Problem 1 (MaxMutualRespect):** Given a set of roles $S$ and the corresponding respect graphs $G_i = (V, E_i), i \in S$, find an assignment $f : S \rightarrow V$, such that $R(f)$ is maximized.

We prove the following theorem for the complexity of our problem.

**Theorem 1:** The MaxMutualRespect problem is NP-complete.

**a) Proof:** We show that the MaxMutualRespect problem is NP-complete even in the special case when the input contains directed acyclic graphs (DAGs). We reduce an instance of the NP-complete $k$-clique [21] problem to the MaxMutualRespect problem. The decision version of the MaxMutualRespect problem asks if there is an assignment $f : S \rightarrow V$ with score $R(f) \geq \theta$, for some value $\theta$. The $k$-clique problem, given an undirected graph $H$ as input, asks if graph $H$ contains a clique of size $k$.

We can reduce an instance of the decision version of the $k$-clique problem to an instance of the decision version of the MaxMutualRespect problem, where the nodes of graph $H$ are the experts. We transform graph $H$ into a DAG $G$ by substituting each of its edges with a bidirectional one. We define a clique in a directed graph $G = (V, E)$ as follows; for every pair of nodes $v_i, v_j$ in the clique, both $(v_i, v_j) \in E$ and $(v_j, v_i) \in E$ hold. We denote as $v_i^j$ that node $i$ was assigned to role $j$. We create a set of $k$ identical instances of $G, G = \{G_1, \ldots, G_k\}$ which is the input to the MaxMutualRespect problem. We set $\theta = k(k-1)$ and see that there exists a $k$-clique in graph $H$, if and only if there is an assignment $f : S \rightarrow V$ with score $R(f) \geq k(k-1)$.

Assume that there exists a $k$-clique $C$ in $H$. Then $C$ is also a $k$-clique in each of the graphs of $G$. Without loss of generality let $C = \{v_1, \ldots, v_k\}$. If we assign a different role to each of the nodes in $C$ we have an assignment $f$ where $F = \{v_1^1, \ldots, v_k^1\}$ is a solution to the MaxMutualRespect problem because a different vertex is assigned for each of the roles. Also the input set $G$ contains identical DAGs so for each $H_i \in G$ the set of edges $E_{G_i} = \{v_i^1, \ldots, v_i^k\}$ of the subgraph induced by $F$ is the same. In this induced subgraph each node $v_i^1$ has an incoming edge originating from each of the other $k-1$ nodes in the solution since all nodes belong to the same clique. Thus, the score of this solution is $R(f) \geq k(k-1)$.

Assume now that for the input set $G = \{G_1, \ldots, G_k\}$ there exists an assignment $f$ whose solution is $F = \{v_1^1, \ldots, v_k^1\}$ with score $R(f) \geq k(k-1)$. This requires that for each node $v_i^1 \in F$ there is a directed edge from all other $k-1$ nodes $F \setminus \{v_i^1\}$ to $v_i^1$ in graph $G_i$, and that there is an edge from $v_i^1$ to all of the other $k-1$ nodes $F \setminus \{v_i^1\}$ in graphs $G \setminus \{G_i\}$, respectively. Since all instances in $G$ are identical, this can only happen if the subgraph induced by the $k$ nodes in $F$ forms a clique $C = \{v_1, \ldots, v_k\}$, which corresponds to a $k$-clique in $H$.

We also consider an interesting special case of the Max-MutualRespect problem where each respect graph $G_i$ is
Algorithm 1 The Greedy algorithm.

**Input:** Set of experts $V$, set of $k$ roles $S$, respect graphs \{$G_1, ..., G_k$\}.

**Output:** Assignment $F$.

1: $F \leftarrow \{\}$
2: $(i^*, v^*) = \arg \max_{(i,v) \in S \times V} s(i,v)$
3: $S \leftarrow S \setminus i^*$
4: $V \leftarrow V \setminus v^*$
5: $F[i^*] = v^*$
6: **while** $|F| < k$ **do**
7: $(i^*, v^*) = \arg \max_{(i,v) \in S \times V} s_F(i,v)$
8: $F[i^*] = v^*$
9: $S \leftarrow S \setminus i^*$
10: $V \leftarrow V \setminus v^*$
11: **end while**
12: **return** $F$

The Greedy algorithm is derived from a full ranking of the experts in $V$. In this case the input is a set of $k$ rankings $P_1, ..., P_k$, one for each role, defined as permutations of the nodes in $V$. The value $P_i[v]$ is the position of node $v$ in the ranking of role $i$. Lower value of $P_i[v]$ denotes higher rank. Given a ranking, we assume that an expert respects all experts above her in the ranking, and is respected by all experts below her in the ranking. In the corresponding graph $G_i$ this implies that we place an edge $(u,v)$ for all pairs of nodes such that $P_i[u] > P_i[v]$. We refer to this problem variant as MaxRankingRespect.

The complexity of MaxRankingRespect remains unresolved. In Section IV we show that there is a polynomial algorithm for finding the assignment with maximum possible respect score $R(f) = k(k-1)$, if such an assignment exists. This is the case where for each role, the expert assigned to that role has higher rank than all team members for that role. If such an assignment does not exist, it remains an open problem if there exists a polynomial-time algorithm for finding the optimal assignment.

### IV. ALGORITHMS

In this section, we describe our algorithms for the MaxMutualRespect and the MaxRankingRespect problems.

#### A. Algorithms for MaxMutualRespect

For the MaxMutualRespect problem we consider a greedy heuristic algorithm, which assigns a score to every role-expert pair, and at each step it selects the assignment with the best (updated) score value. We refer to the algorithm as Greedy and present its outline in Algorithm 1.

The algorithm initially computes for each role-expert pair $(i,v)$ the score value:

$$s(i,v) = \deg_{G_i}^-(v) + \frac{1}{k} \sum_{j \in S; j \neq i} \deg_{G_j}^+(v),$$

(1)

where $\deg_{G_i}^-(v)$ and $\deg_{G_j}^+(v)$ denote the in-degree and out-degree of expert $v$ in the respect graph $G_i$, respectively. High in-degree in graph $G_i$ means that node $v$ is highly respected for role $i$, while high average out-degree for the remaining roles means that node $v$ has on average high respect for the other experts in the other roles.

First, Greedy selects the role-expert assignment pair with the highest score. It then proceeds iteratively, where, given the partial assignment $F$ the algorithm computes a new value for each unassigned role-expert pair $(i,v)$ as follows:

$$s_F(i,v) = \deg_{G_i[F \cup \{v\}]}^-(v) + \frac{1}{k - |F| + 1} \sum_{j : f(j) = \emptyset} \deg_{G_j[V \setminus F]}^+(v),$$

(2)

where $f(j) = \emptyset$ denotes an unassigned role, and $G[F]$ denotes the induced subgraph of the set $F \subset V$. Intuitively, a pair $(i,v)$ receives high score if node $v$ has a lot of incoming edges (respect) from the assigned nodes in $F$ for role $i$, it has a lot of outgoing edges (respect) to the nodes of the assigned roles, and has high average respect for the unassigned nodes in the unassigned roles. The terms in the above values are normalized to be in the same scale, and we use dictionaries to efficiently update them in each iteration. This iterative selection step continues until all roles have been assigned an expert. The running time of Greedy is $O(k^2n)$. Note that Greedy makes local decisions by considering exhaustively all the available valid assignments and selecting the locally optimal one. However, as we see in Section V this may lead the algorithm to get stuck in local optima. To overcome this limitation we propose a randomized variant of Greedy that we denote as RandGreedy.

RandGreedy follows the same score computations as Greedy, but instead of selecting the $(i,v)$ pair that maximizes the score, it first selects a role $i \in S : f(i) = \emptyset$ uniformly at random, and then selects the assignment pair $(i,v)$ that maximizes the score. We repeat the algorithm $f$ times and we report the assignment with the highest score. The running time of RandGreedy is $O(\ell kn)$.

#### B. Algorithms for MaxRankingRespect

For the MaxRankingRespect problem, we first present the MaxScore algorithm that finds an assignment with maximum possible respect score $R(f) = k(k-1)$, if such a solution exists. The outline of the algorithm is shown in Algorithm 2.

The algorithm maintains a dictionary $F$ that stores which experts have been assigned to which roles, and a set $D$ that maintains experts that are ineligible for assignment. MaxScore proceeds iteratively and repeats the following steps in each iteration. It picks uniformly at random an unassigned role $r \in S : f(r) = \emptyset$, and traverses the full ranking of $r$ in a top-down order. For each encountered expert $v$ we have the following cases: (i) If $v$ has never been encountered before it assigns it to role $r$, $f(r) = v$ and continues with another unassigned role; (ii) If $v \in F$ and it is assigned to some other role $\ell$, it cancels this assignment, setting $f(\ell) = \emptyset$, and
Algorithm 2: The MaxScore algorithm.

**Input:** A set of $n$ experts $V$, a set of $k$ roles $S$, rankings $\{P_1, \ldots, P_k\}$.

**Output:** Assignment $F$.

1. $F \leftarrow \{\}$
2. $D \leftarrow \{\}$
3. $C \leftarrow \{1, \ldots, 1\}$
4. **while** $|F| < k$ and $\exists i \in S : C[i] \neq n$ **do**
5. $r \leftarrow$ pick an unassigned role s.t. $C[r] \neq n$
6. **for** $j \in \{C[r], \ldots, n\}$ **do**
7. $C[r] \leftarrow j$
8. $v \leftarrow$ the $j$-th expert in $P_r$
9. **if** $v \in F : F[\ell] = v$ **then**
10. $F[\ell] \leftarrow 0$
11. $D \leftarrow D \cup \{v\}$
12. **else if** $v \notin D$ **then**
13. $F[r] \leftarrow v$
14. **break**
15. **end if**
16. **end for**
17. **end while**
18. **return** $F$

From the algorithm it is clear that all nodes traversed before an assignment are either already in $D$, or added in $D$ during the traversal. In this case the assignment $f(r) = v$ is not possible so $P_i[v] < P_r[u]$ cannot hold either.

For the backward direction, suppose there exists an assignment $f^*$ with score $R(f^*) = k(k - 1)$, Now assume for the sake of contradiction that the algorithm returns an empty solution. Since the algorithm goes through all the nodes in all rankings this means that the experts in $f^*$ were considered and rejected by the algorithm. Without loss of generality assume that the first expert rejected by the algorithm is $v = f^*(i)$ for the role $i$. Let $f$ be the (partial) assignment at that time. Expert $v$ is rejected when $v$ has already been added in set $D$. This can happen in two ways:

- The algorithm is going down the ranking of role $i$, and when we encounter $f^*(i)$ is already assigned to role $j$, $j \neq i$. We thus have that $f(j) = v$. Recall that $i$ is the first role that has its expert rejected. Since as we have argued all nodes before $f(j)$ in $P_j$ are in $D$, it must be that $P_i[v] < P_j[f^*(j)]$. Therefore, $P_i[f^*(i)] < P_j[f^*(j)]$, which is a respect violation. Therefore, $f^*$ cannot have score $k(k - 1)$, leading to a contradiction.

- We have that $f(i) = f^*(i) = v$. The algorithm is going down the ranking of role $j$ and it encounters node $v$. Since role $i$ is the first to have its expert rejected, and role $j$ is not assigned an expert it must be that $P_i[v] < P_j[f^*(j)]$. Therefore, $P_i[f^*(i)] < P_j[f^*(j)]$, which is a respect violation, leading to a contradiction.

Therefore, if there exists a max-score assignment $f^*$ the algorithm will return a non-empty solution.

The MaxScore algorithm will return the assignment with maximum score if such exists, but returns no solution otherwise. It remains an open question if there exists a polynomial-time algorithm that can find the optimal assignment. We consider an approximation algorithm for this case.

Furthermore, we propose the TopCandidates algorithm which works as follows. The algorithm considers the roles in a random order. For each role $r$ it assigns the expert highest in the ranking $P_r$ that has not already been assigned. We repeat the algorithm $\ell$ times and report the assignment with the highest score. The running time complexity of TopCandidates is $O(\ell nk)$.

**Lemma 1:** Algorithm MaxScore returns a non-empty assignment $f$ if and only if there exists an assignment with maximum score $k(k - 1)$.

**Proof:** To prove the forward direction we need to define the notion of a respect violation. Given an assignment (or partial assignment) $f$, there is a respect violation if there are roles $i, j$ such that $P_i[f(j)] < P_i[f(i)]$. In this case $f(i)$ does not get respect from $f(j)$ for her role. A complete assignment $f^*$, $|f^*| = k$ with no respect violations has maximum score $R(f^*) = k(k - 1)$.

We make the following claim: when the algorithm assigns a node $v$ to role $r$ in line 13, it creates no respect violations. A respect violation may be created if there exists role $\ell$ such that $f(\ell) = u$ and either $P_r[u] < P_r[v]$, or $P_r[v] < P_r[u]$. Recall that in order to assign an expert to a role, we first need to traverse the ranking of the role until we reach the node that corresponds to the expert. Therefore if $P_r[u] < P_r[v]$, we would encounter node $u$ while traversing the ranking $P_r$ before reaching node $v$. Since $u \in F$ for a different role than $r$ the respect violation $P_i[u] < P_r[w]$ cannot hold. Similarly, in order to assign $f(\ell) = u$, we must have first traversed the nodes preceding $u$ in ranking $P_i$ before performing the assignment.

We also propose the AllCandidates algorithm, an extension of TopCandidates that exhaustively makes each possible role-expert pair $(i, v) \in S \times V$ as a first assignment. After the first assignment, it proceeds in the same manner as TopCandidates.
TABLE I: A summary of the Citations dataset statistics; # Experts is the total number of experts, # Roles is the number of roles required to be fulfilled, Avg. End./Role denotes the average in-degree of the respect graphs, Avg. End./Expert denotes the average in-degree of all experts, Max End./Expert denotes the maximum in-degree of all experts, # Overlap. Experts is the number of experts encountered in all the respect graphs, k is an abbreviation for thousands.

<table>
<thead>
<tr>
<th>Team 1</th>
<th>Team 2</th>
<th>Team 3</th>
<th>Team 4</th>
<th>Team 5</th>
<th>Team 6</th>
</tr>
</thead>
<tbody>
<tr>
<td># Experts</td>
<td>68.9k</td>
<td>31.7k</td>
<td>95.9k</td>
<td>37.5k</td>
<td>56.7k</td>
</tr>
<tr>
<td># Roles</td>
<td>4</td>
<td>4</td>
<td>8</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td>Avg. End./Role</td>
<td>728.7k</td>
<td>428.7k</td>
<td>689.6k</td>
<td>373.7k</td>
<td>586.4k</td>
</tr>
<tr>
<td>Avg. End./Expert</td>
<td>42.25</td>
<td>80.8</td>
<td>90.8k</td>
<td>39.82</td>
<td>41.33</td>
</tr>
<tr>
<td>Max End./Expert</td>
<td>13k</td>
<td>775</td>
<td>2.6k</td>
<td>37.3k</td>
<td>4.6k</td>
</tr>
</tbody>
</table>

V. EXPERIMENTS

This section explores the practicality of our algorithms using real-world datasets. Specifically, (i) we use real-world datasets to analyze the performance of our algorithms for the maximutualrespect problem; (ii) we use real-world datasets to analyze the performance of our algorithms for the MaxRankingRespect problem; (iii) we provide a runtime analysis of our algorithms for all datasets.

For all randomized algorithms and experiments we set the parameter $\ell$ (the maximum number of iterations) to 50 because we saw that in real applications it generally leads to reasonable solutions and runtime performances. For all our experiments we use a single process implementation of the algorithms on a 64-bit MacBook Pro with an Intel Core i7 CPU at 2.6GHz and 16GB RAM. We make the code, the datasets and the chosen parameters available online.

A. Results for MaxMutualRespect

For the MaxMutualRespect problem we will experiment with the algorithms Greedy and RandGreedy presented in Section IV. For the latter, we also report the average and standard deviation score it achieves (denoted as $\text{AvgRandGreedy}$ in the plots). We also compare against a baseline Ranking that sorts the experts in each role according to their score $s(v, i)$, and then runs TopCandidates, selecting the top candidates in each position. This corresponds to a greedy algorithm that computes the scores once, and then assigns the candidates with the highest score. We perform $\ell$ different runs (different order in role selection), and report the solution with the maximum score. The running time complexity of Ranking is $O(\ell k n \log n + \ell k n)$. Finally, in our experiments Max denotes the maximum possible respect score that can be achieved even though a solution with such score might not exist.

1) Citation networks: We study the MaxMutualRespect problem on real data generated from academic citation networks. In this setting, the experts are scientists, and the roles correspond to scientific fields. The respect graph is formed by citations: author $v$ respects author $u$ in scientific field $i$, if author $u$ has a paper in field $i$, and author $v$ has a publication that cites that paper.

More precisely, we consider the following scientific fields in Computer Science: Artificial Intelligence (AI), Neural Networks (NN), Natural Language Processing (NLP), Robotics, Data Mining (DM), Algorithms, Data Bases (DB), Theory, Signal Processing (SP), Computer Networking (CN), Information Retrieval (IR), Wireless Networks and Mobile Computing (Wireless), Software Engineering (SE), High-Performance Computing (HPC), Distributed and Parallel Computing (DPC), Operating Systems (OS). Using a publicly available resource\(^4\) we find the top-tier conferences for each field. We then use the DBLP dataset\(^5\) to extract the set of publications and authors that belong to these top-tier conferences, and create the citation networks for the different fields. To reduce noise we removed all self-loops from the graphs, and iteratively pruned authors with less than 5 incoming and outgoing citations.

We consider six possible teams: (1) Team 1 is an AI & Applications team that requires scientists for the fields AI, NN, NLP, and Robotics; (2) Team 2 is a Data & Algorithms team that requires scientists for the fields DM, Algorithms, DB, and Theory; (3) Team 3 requires scientists for all fields of Teams 1 and 2; (4) Team 4 is a Systems team that requires scientists for the fields SE, HPC, DPC, and OS; (5) Team 5 is a Networks team that requires scientists for the fields SP, CN, IR, and Wireless; (6) Team 6 requires scientists for all fields of Teams 5 and 6. Table I exhibits some statistics on the size and the properties of the input dataset for each team.

An interesting observation is that on average and for all teams an expert receives at least 39 citations from her peers (row 4). Furthermore, we see that Team 1 (AI & Applications) receives on average more endorsements per field, while Team 2 (Data & Algorithms) receives on average less endorsements per field (row 4). Finally, note that we consider teams from fields that are related to each other and therefore include over-

\(^3\)https://www.dropbox.com/sh/47kakiyhf58jahb/AAJDjIiLcJqNM-GvSf-d0Gxa?dl=0

\(^4\)https://dl.acm.org/ccs/ccs_flat.cfm

\(^5\)https://aminer.org/citation
We demonstrate the quality of our results in Table II where we present the experts selected by RandGreedy for the different teams. For calibration, we also present the scientists with the highest number of citations in each field (Rows 2 and 7 denoted as Top). We observe that in all experiments the produced teams contain acclaimed researchers who cite and acknowledge the contributions of their peers in different fields. However, none of the teams contains the most cited author in any of the fields. Also, the assigned scientists for Team 3 differ from those assigned in Teams 1 & 2 even though the set of roles required by Team 3 is a superset of those in Teams 1 & 2. Furthermore, in Team 2, the algorithm selects A. Tomkins for DM, D. Sivakumar for Algorithms, R. Kumar for DB, and S. Muthukrishnan for Theory. The first three authors have worked a lot in these fields and they have heavily cited each other, while S. Muthukrishnan is a well-known theorist who has also publications in DB and DM venues.

B. Results for MAXRANKINGRESPECT

We now evaluate the algorithms for the MAXRANKINGRESPECT problem.

1) NBA Statistics: We evaluate the algorithms MAXRANKINGRESPECT using the NBA dataset\(^6\). The dataset contains individual basketball player statistics for different NBA seasons, for a range of basic statistics, such as points, assists, rebounds etc., to more advanced performance metrics such as value over replacement. We use data for the seasons 2010 - 2017, and the following subset of 11 performance metrics that we consider important in assembling a basketball team: STL, AST, FT, BLK, FG, TRB, 2P, 3P, DBPM, OBPM, VORP\(^7\). These performance metrics correspond to roles in our setting. We prune the set of players so as to keep the ones that have played in at least one third of the games of the season, and have played at least 15 minutes per game. In the resulting data we have the following number of players in each year; (i) year 2010: 278 players, (ii) year 2011: 289 players, (iii) year 2012: 286 players, (iv) year 2013: 291 players, (v) year 2014: 294 players, (v) year 2015: 319 players, (v) year 2016: 299 players, (v) year 2017: 310 players. We create the ranking for each season by sorting the players in decreasing order of the metric value.

Figure 1b shows the performance of AllCandidates and TopCandidates. Note that for all seasons a solution with maximum respect score exists and was found by MaxScore. We observe that AllCandidates that always finding a maximum respect score solution, performs slightly better than TopCandidates.

Table III shows indicatively the results of the three algorithms for the seasons 2010 and 2016. Interestingly, the TopCandidates algorithm, which does not achieve the maximum score, selects many players such as Lebron James, R. Westbrook, or Stephen Curry, that are at the top, or close to the top of their corresponding ranking. These players are also at the top of other role rankings as well, and thus they do not

\(^6\)https://www.kaggle.com/drgilermo/nba-players-stats

\(^7\)We refer the reader to https://www.basketball-reference.com/about/glossary.html for the description of these attributes
### TABLE II: Teams produced by RandGreedy on different subsets of scientific fields. Top denotes the scientists with the highest number of citations in the corresponding field. Rows 1 and 6 represent the team roles. Each of the rows 3-5 and 8-10 represent a different team found by the proposed algorithm RandGreedy.

<table>
<thead>
<tr>
<th>Season 2010</th>
<th>Season 2016</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>MaxScore</strong></td>
<td><strong>MaxScore</strong></td>
</tr>
<tr>
<td>STL</td>
<td>R.Rondo</td>
</tr>
<tr>
<td>AST</td>
<td>S.Nash</td>
</tr>
<tr>
<td>FT</td>
<td>C.Anthony</td>
</tr>
<tr>
<td>BLK</td>
<td>A.Bogut</td>
</tr>
<tr>
<td>FG</td>
<td>K.Bryant</td>
</tr>
<tr>
<td>TRB</td>
<td>Z.Randolph</td>
</tr>
<tr>
<td>2P</td>
<td>A.Stoudemire</td>
</tr>
<tr>
<td>3P</td>
<td>A.Brooks</td>
</tr>
<tr>
<td>DBPM</td>
<td>M.Camby</td>
</tr>
<tr>
<td>OBPM</td>
<td>M.Ginobili</td>
</tr>
<tr>
<td>VORP</td>
<td>J.Smith</td>
</tr>
<tr>
<td><strong>AllCandidates</strong></td>
<td><strong>AllCandidates</strong></td>
</tr>
<tr>
<td>STL</td>
<td>R.Rondo</td>
</tr>
<tr>
<td>AST</td>
<td>S.Nash</td>
</tr>
<tr>
<td>FT</td>
<td>C.Anthony</td>
</tr>
<tr>
<td>BLK</td>
<td>A.Bogut</td>
</tr>
<tr>
<td>FG</td>
<td>K.Bryant</td>
</tr>
<tr>
<td>TRB</td>
<td>Z.Randolph</td>
</tr>
<tr>
<td>2P</td>
<td>A.Stoudemire</td>
</tr>
<tr>
<td>3P</td>
<td>A.Brooks</td>
</tr>
<tr>
<td>DBPM</td>
<td>M.Camby</td>
</tr>
<tr>
<td>OBPM</td>
<td>M.Ginobili</td>
</tr>
<tr>
<td>VORP</td>
<td>J.Smith</td>
</tr>
<tr>
<td><strong>TopCandidates</strong></td>
<td><strong>TopCandidates</strong></td>
</tr>
<tr>
<td>STL</td>
<td>R.Rondo</td>
</tr>
<tr>
<td>AST</td>
<td>S.Nash</td>
</tr>
<tr>
<td>FT</td>
<td>C.Anthony</td>
</tr>
<tr>
<td>BLK</td>
<td>A.Bogut</td>
</tr>
<tr>
<td>FG</td>
<td>K.Bryant</td>
</tr>
<tr>
<td>TRB</td>
<td>Z.Randolph</td>
</tr>
<tr>
<td>2P</td>
<td>A.Stoudemire</td>
</tr>
<tr>
<td>3P</td>
<td>A.Brooks</td>
</tr>
<tr>
<td>DBPM</td>
<td>M.Camby</td>
</tr>
<tr>
<td>OBPM</td>
<td>M.Ginobili</td>
</tr>
<tr>
<td>VORP</td>
<td>J.Smith</td>
</tr>
</tbody>
</table>

### VI. Conclusion

We introduced the novel problem of creating teams of experts associated with distinct roles such that the total respect that these experts receive by the other team members with respect to their associated role is maximized. We showed that the problem is NP-hard to solve and designed heuristic algorithms for solving it in practice. For the variant of the problem where respect graphs are derived from rankings, we design a polynomial algorithm for finding a team with maximum respect, if such a team exists, as well as approximation algorithms that rely on the properties of the rankings. Our experiments with real-world and synthetic datasets demonstrate the utility of our algorithms in practice. For future work, we are interested in studying the weighted version of our problem.

have sufficient respect for the player that finally assumes this role (a common phenomenon with star players in team sports).

### C. Runtime analysis

We now investigate the runtime efficiency of all our algorithms. We report the running times of the algorithms on the real datasets Citations and NBA and the synthetic datasets for the MAXMutualRespect and the MAXRankingRespect problems, respectively. All times are averaged over 5 runs and are reported in seconds.

The results for MAXMutualRespect using the Citations dataset are shown in Figure 2a. We compare the runtime performances of Greedy, RandGreedy and Ranking with asymptotic running time complexities $O(k^2 n)$, $O(\ell k n)$ and $O(\ell k n \log n + \ell k n)$, respectively. We observe that the algorithms Greedy and Ranking are very efficient. In fact, their execution times are less than a minute which renders them very scalable. RandGreedy appears to be slower than the other two algorithms. We noticed that for one iteration of RandGreedy ($\ell = 1$) its corresponding asymptotic runtime complexity becomes $O(\ell k n)$ and its running time becomes comparable to that of Greedy. Note, however, that even though for smaller values of $\ell$ RandGreedy is faster, its performance is also more likely to drop.

The results for MAXRankingRespect using the NBA dataset are shown in Figure 2b. We compare the performance of MaxScore, AllCandidates and TopCandidates with asymptotic running time complexities $O(\ell k n)$, $O(\ell k^2 n^2)$ and $O(\ell k n)$, respectively. The results are shown in Figure 2b. Note that we only report the running time of AllCandidates which does not exceed 20 seconds, but we omit the running times of MaxScore and TopCandidates because these are less than a millisecond. Here, we see that the asymptotic running time complexities agree with the algorithms' performances; MaxScore and TopCandidates are highly efficient while AllCandidates is the slowest of the three algorithms.
Fig. 2: Time (sec) for the MAXMUTUALRESPECT problem on the real-world dataset Citations (top) and the MAXRANKINGRESPECT problem on the real-world dataset NBA (bottom).

REFERENCES


