Finding Teams of Maximum Mutual Respect

Sofia Maria Nikolakaki[1], Evaggelia Pitoura[2], Evimaria Terzi[1], Panayiotis Tsaparas[2]
[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-complete 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 press [1,2] 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 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 and 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

[1]https://www.nytimes.com/guides/business/manage-a-successful-team
[2]https://www.forbes.com/sites/glennllopis/2012/10/01/6-ways-successful-teams-are-built-to-last/\#60ab15bf2b55

team formation, rank aggregation and endorsement deduction. Below, we review this work and discuss 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]–[17], or task-arrival process [18]. There are three key differences between our work and that line of work: 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 [19]–[23]. 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 [24], [25]. 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 $i \in S$. Let $F = f(S)$ denote the selected team of experts. We assume that the function f is injective, that is, each team member can only be used for a single role, and therefore $|F| = |S|$. 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 can prove the following theorem for the complexity of our problem. The proof of the theorem will be provided in an extended version of this manuscript.

Theorem 1: The MaxMutualRespect problem is NP-complete.

We also consider an interesting special case of the MaxMutualRespect problem where each respect graph G_i is derived from a *full ranking* of the experts in V. In this case the input is a set of k rankings $P_1, ..., P_k$, where $k = |S|$ and each ranking corresponds to a 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.

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 algorithm, which assigns a score to every role-expert pair, and at each step it selects the assignment with the highest updated score value. We refer to the algorithm as *Greedy*.

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

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

where $deg^{-}_{G_i}(v)$ and $deg^{+}_{G_i}(v)$ 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:

$$s_F(i,v) = deg^-_{G_i[F\cup\{v\}]}(v) + |\{(v,f(j)) \in E_j : f(j) \neq \emptyset\}| + \frac{1}{k-|F|-1} \sum_{j:f(j)=\emptyset, j\neq i} deg^+_{G_j[V\setminus F]}(v), \quad (2)$$

where $f(j) = \emptyset$ denotes an unassigned role, and $G[F]$ denotes the induced subgraph of the set $F \subseteq V$. 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 ℓ times and we report the assignment with the highest score. The running time of *RandGreedy* is O(ℓkn).

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 1.

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 ℓ, it cancels this assignment, setting $f(\ell) = \emptyset$, and adds v to D thus rendering the expert ineligible for any future assignment. It then continues traversing the ranking P_r of role r; (iii) If $v \in D$ it ignores v and continues traversing the ranking P_r. The algorithm terminates when either of the two following conditions is satisfied: (i) Assignment F contains k experts, one for each role; (ii) All rankings are traversed without finding an assignment F and the algorithm returns the empty set. The running time of *MaxScore* is O(kn).

Algorithm 1 The *MaxScore* algorithm.

Input: Set S of k roles, rankings $\{P_1, ..., P_k\}$ with n experts.
Output: Assignment F.

$F \leftarrow \{\}; D \leftarrow \{\}; C \leftarrow \{1, \ldots, 1\}$
while $|F| < k$ and $\exists i \in S : C[i] \neq n$ **do**
  $r \leftarrow$ pick an unassigned role s.t. $C[r] \neq n$
  for $j \in \{C[r], \ldots, n\}$ **do**
    $C[r] \leftarrow j$
    $v \leftarrow$ the j-th expert in P_r
    if $v \in F : F[\ell] = v$ **then**
      $F[\ell] \leftarrow \emptyset$
      $D \leftarrow D \cup \{v\}$
    else if $v \notin D$ **then**
      $F[r] \leftarrow v$
      break
    end if
  end for
end while

return F

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

Due to space limitations the proof of Lemma 1 will be provided in an extended version of this paper.

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 ℓ times and report the assignment with the highest score. The running time complexity is O(ℓkn).

Lemma 2: Algorithm *TopCandidates* is a $\frac{1}{2}$-approximation algorithm for the MAXRANKINGRESPECT problem.

a) Proof [Sketch]:: When the algorithm considers the i-th role in the worst case it will create $i-1$ respect violations. Therefore, the total number of respect violations of the produced assignment f is at most $k(k-1)/2$, and the respect score $R(f) \geq k(k-1) - k(k-1)/2 = k(k-1)/2$. Since the optimal assignment has score at most $k(k-1)$ the assignment f is a $\frac{1}{2}$-approximation solution. □

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* each time selecting to assign to a role the highest ranked node that has not been assigned. From all the candidate assignments it returns the one assignment with the highest respect score. Since the assignment of *TopCandidates* is one

	Team 1	Team 2	Team 3	Team 4	Team 5	Team 6
# Experts	68.9k	31.7k	95.9k	37.5k	56.7k	90.8k
# Roles	4	4	8	4	4	8
Avg. End./Role	728.7k	44.2k	659.5k	373.7k	586.4k	527.4k
Avg. End./Expert	42.25	55.85	55	39.82	41.33	46.46
Max End./Expert	2.6k	775	2.6k	920	1.88k	1.9k
# Overlap. Experts	13k	15.1k	37.3k	10.2k	4.6k	11.6k

TABLE I: A summary of the *Citations* dataset statistics; # Experts: Number of experts; # Roles: Number of roles to be fulfilled; Avg. End./Role: Average in-degree of the respect graphs; Avg. End./Expert: Average in-degree of all experts; Max End./Expert: Maximum in-degree of all experts; # Overlap. Experts: Number of experts encountered in all the respect graphs.

of the assignments considered by *AllCandidates*, it follows that *AllCandidates* is also a $\frac{1}{2}$-approximation algorithm for the problem. The running time complexity is O($\ell k^2 n^2$).

V. Experiments

This section explores the practicality of our algorithms. Specifically, (i) we evaluate the performance of our algorithms on real-world datasets, (ii) we provide a runtime analysis.

For all randomized algorithms and experiments we set the parameter ℓ (the maximum number of iterations) to 50. 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 [3].

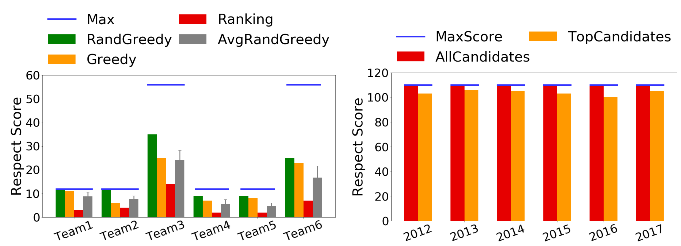
A. *Results for* MaxMutualRespect

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 *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 ℓ different runs (different order in role selection), and report the solution with the maximum score. The running time complexity of *Ranking* is O($kn \log n + \ell kn$). Finally, in our experiments *Max* denotes the maximum possible respect score that can be achieved even though a solution with such score might not exists.

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

[3]https://github.com/smnikolakaki/teammutualrespect



(a) *Citations* dataset (b) *NBA* dataset

Fig. 1: Respect Score (R) of the algorithms for the MaxMutualRespect problem (left) and the MaxRankingRespect problem (right).

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 with scientists from AI, NN, NLP, and Robotics; (2) Team 2 is a Data & Algorithms team with scientists from DM, Algorithms, DB, and Theory; (3) Team 3 has scientists from the fields of Teams 1 and 2; (4) Team 4 is a Systems team with scientists from SE, HPC, DPC, and OS; (5) Team 5 is a Networks team with scientists from SP, CN, IR, and Wireless; (6) Team 6 has scientists from the fields of Teams 5 and 6. Table I exhibits some statistics on the datasets.

Figure 1a shows the performance of the *Greedy*, *RandGreedy*, *Ranking* and *AvgRandGreedy* algorithms for the different teams. The results show that *RandGreedy* outperforms *Greedy*, while *Ranking* performs poorly on all tasks. In the cases of Team 1 and Team 2, *RandGreedy* is able to find teams of maximum score. Interestingly, the performance of *AvgRandGreedy* is closer to the one of *Greedy*, and for Team 2 it performs slightly better. This indicates that, in this dataset, *Greedy* is more easily trapped in local optima. Furthermore, we see that as the number of candidate experts grows (Teams 3 and 6) the performance differences between *Greedy* and *RandGreedy* are more pronounced. This is because *Greedy*

[4]https://dl.acm.org/ccs/ccs_flat.cfm

[5]https://aminer.org/citation

	AI	**NN**	**NLP**	**Robotics**	**DM**	**Algorithms**	**DB**	**Theory**
Top	J.Lafferty	G.Hinton	E.Hovy	V.Kumar	C.Aggarwal	A.Goldberg	R.Agrawal	M.Szegedy
Team1	L.Zettlemoyer	D.Koller	C.Manning	A.Ng	-	-	-	-
Team2	-	-	-	-	A.Tomkins	D.Sivakumar	R.Kumar	S. Muthukrishnan
Team3	R.Mooney	M.Jordan	A.McCallum	D.Fox	I.Dhillon	W.Wang	Q.Yang	Y. Freund
	SE	**HPC**	**DPC**	**OS**	**SP**	**CN**	**IR**	**Wireless**
Top	G.Rothermel	I.Foster	L.Ni	M.Kaashoek	G.Giannakis	D.Towsley	C.Buckley	J.Polastre
Team4	R.Gupta	P.Balaji	A.Vishnu	D.Panda	-	-	-	-
Team5	-	-	-	-	Z.Han	S.Zhong	B.Li	W.Trappe
Team6	M.Li	J.Wu	X.Li	B.Li	Z.Yang	Y.Liu	Z.Li	Y.Wang

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. A dash in a column denotes that this role was not requested for the team.

gets trapped in local optima, while *RandGreedy* is able to avoid them through random selections.

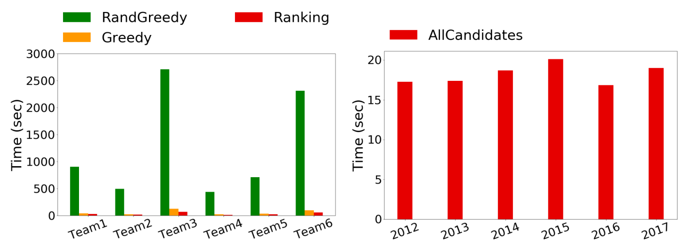
We demonstrate the quality of our results in Table II where we present the experts selected by *RandGreedy* for the different teams. For calibration, we also present the scientists with the highest number of citations in each field (Rows 2 and 7 denoted as *Top*). We observe that in all experiments the 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

In this section, we 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. 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. We refer the reader to [7] for the description of these attributes. 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; 2010: 278 players; 2011: 289 players; 2012: 286 players; 2013: 291 players; 2014: 294 players; 2015: 319 players; 2016: 299 players; 2017: 310 players. We create the ranking for each performance metric 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* which always finding a maximum respect score solution, performs slightly better than *TopCandidates*.



(a) *Citations* dataset (b) *NBA* dataset

Fig. 2: Runtime performance of the algorithms for the MaxMutualRespect problem (left) and the MaxRankingRespect problem (right).

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. Westbook, 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 do not have sufficient respect for the player that finally occupy 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 datasets *Citations* and *NBA* 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*. 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 $O(kn)$ and its running time becomes comparable to that of *Greedy*. Note, however, that even though for smaller values of ℓ *RandGreedy* is faster, its performance is also more likely to drop.

[6]https://www.kaggle.com/drgilermo/nba-players-stats
[7]https://www.basketball-reference.com/about/glossary.html

	Season 2010			Season 2016		
	MaxScore	***AllCandidates***	***TopCandidates***	***MaxScore***	***AllCandidates***	***TopCandidates***
STL	R.Rondo	R.Rondo	R.Rondo	R.Rubio	R.Rubio	R.Westbrook
AST	S.Nash	S.Nash	S.Nash	R.Rondo	R.Rondo	R.Rondo
FT	C.Anthony	C.Anthony	D.Nowitzki	D.DeRozan	D.Gallinari	J.Harden
BLK	A.Bogut	S.Battier	A.Bogut	H.Whiteside	R.Gobert	H.Whiteside
FG	K.Bryant	K.Bryant	K.Durant	C.J.McCollum	E.Fournier	K.Durant
TRB	Z.Randolph	Z.Randolph	D.Howard	A.Drummond	K.Love	A.Drummond
2P	A.Stoudemire	D.Lee	A.Stoudemire	K.A. Towns	B.Griffin	L.James
3P	A.Brooks	A.Brooks	A.Brooks	D.Lillard	J.R.Smith	K.Thompson
DBPM	M.Camby	T.Ratliff	M.Camby	A.Bogut	A.Bogut	A.Bogut
OBPM	M.Ginobili	M.Ginobili	D.Wade	C.Paul	M.Conley	S.Curry
VORP	J.Smith	J.Kidd	L.James	K.Lowry	N.Jokic	K.Lowry

TABLE III: Teams of basketball players for the seasons 2010 (left) and 2016 (right). Column 1 represents the team roles. Each of the columns 2-7 represent a different team found by the corresponding algorithm.

The results for MAXRANKINGRESPECT using the *NBA* dataset are shown in Figure 2b. In Figure 2b we compare the performances of *MaxScore*, *AllCandidates* and *TopCandidates*. 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.

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 datasets demonstrate the utility of our algorithms in practice. For future work, we are interested in studying the weighted version of our problem.

VII. ACKNOWLEDGMENTS

This work was supported and by ERDF and Greece RCI project T1EDK-00108 and by NSF grants: IIS 1908510 and IIS 1813406.

REFERENCES

[1] V. Govindarajan and A. K. Gupta, "Building an effective global business team," *MIT Sloan Management Review*, vol. 42, no. 4, pp. 63–63, 2001.
[2] L. Mealiea and R. Baltazar, "A strategic guide for building effective teams," *Public Personnel Management*, vol. 34, no. 2, pp. 141–160, 2005.
[3] T. Lappas, K. Liu, and E. Terzi, "Finding a team of experts in social networks," in *KDD*, 2009.
[4] M. Kargar, A. An, and M. Zihayat, "Efficient bi-objective team formation in social networks," in *ECML PKDD*, 2012.
[5] A. Majumder, S. Datta, and K. Naidu, "Capacitated team formation problem on social networks," in *KDD*, 2012.
[6] C.-T. Li, M.-K. Shan, and S.-D. Lin, "On team formation with expertise query in collaborative social networks," *KAIS*, 2015.
[7] S. S. Rangapuram, T. Bühler, and M. Hein, "Towards realistic team formation in social networks based on densest subgraphs," in *WWW*, 2013.
[8] X. Yin, C. Qu, Q. Wang, F. Wu, B. Liu, F. Chen, X. Chen, and D. Fang, "Social connection aware team formation for participatory tasks," *IEEE Access*, 2018.
[9] L. Li and H. Tong, "Network science of teams: Characterization, prediction, and optimization," in *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, 2018, pp. 783–784.
[10] X. Wang, Z. Zhao, and W. Ng, "A comparative study of team formation in social networks," in *International conference on database systems for advanced applications*. Springer, 2015, pp. 389–404.
[11] A. Bhowmik, V. Borkar, D. Garg, and M. Pallan, "Submodularity in team formation problem," in *SDM*, 2014.
[12] Z. Cheng, Y. Yang, C. Tan, D. Cheng, A. Cheng, and Y. Zhuang, "What makes a good team? a large-scale study on the effect of team composition in honor of kings," in *The World Wide Web Conference*, 2019, pp. 2666–2672.
[13] A. Ene, S. M. Nikolakaki, and E. Terzi, "Team formation: Striking a balance between coverage and cost." *arXiv: Data Structures and Algorithms*, 2020.
[14] M. Kargar, M. Zihayat, and A. An, "Finding affordable and collaborative teams from a network of experts," in *SDM*, 2013.
[15] I. Kouvatis, K. Semertzidis, M. Zerva, E. Pitoura, and P. Tsaparas, "Forming compatible teams in signed networks," *arXiv preprint arXiv:2001.03128*, 2020.
[16] C.-T. Li, M.-Y. Huang, and R. Yan, "Team formation with influence maximization for influential event organization on social networks," *World Wide Web*, vol. 21, no. 4, pp. 939–959, 2018.
[17] S. M. Nikolakaki, O. Dibie, A. Beirami, N. Peterson, N. Aghdaie, and K. Zaman, "Competitive balance in team sports games," *arXiv preprint arXiv:2006.13763*, 2020.
[18] A. Anagnostopoulos, L. Becchetti, C. Castillo, A. Gionis, and S. Leonardi, "Online team formation in social networks," in *WWW*, 2012.
[19] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar, "Rank aggregation methods for the web," in *Proceedings of the 10th international conference on World Wide Web*, 2001, pp. 613–622.
[20] R. Fagin, R. Kumar, K. S. McCurley, J. Novak, D. Sivakumar, J. A. Tomlin, and D. P. Williamson, "Searching the workplace web," in *Proceedings of the 12th international conference on World Wide Web*. ACM, 2003, pp. 366–375.
[21] D. Sculley, "Rank aggregation for similar items," in *Proceedings of the 2007 SIAM international conference on data mining*. SIAM, 2007, pp. 587–592.
[22] S. S. Sohail, J. Siddiqui, and R. Ali, "User feedback based evaluation of a product recommendation system using rank aggregation method," in *Advances in Intelligent Informatics*. Springer, 2015, pp. 349–358.
[23] N. Ailon, M. Charikar, and A. Newman, "Aggregating inconsistent information: ranking and clustering," *Journal of the ACM (JACM)*, vol. 55, no. 5, p. 23, 2008.
[24] H. Pérez-Rosés, F. Sebé, and J. M. Ribó, "Endorsement deduction and ranking in social networks," *Computer Communications*, vol. 73, pp. 200–210, 2016.
[25] Y. Wu, N. Dhakal, D. Xu, and J.-H. Cho, "Analysis and prediction of endorsement-based skill assessment in linkedin," in *2018 IEEE 42nd Annual Computer Software and Applications Conference (COMPSAC)*, vol. 1. IEEE, 2018, pp. 461–470.