Multi-Objective Fairness in Team Assembly

Rodrigo Borges, Otto Sahlgrens, Sami Koivunen, Kostas Stefanidis(✉), Thomas Olsson, and Arto Laitinen

Tampere University, Tampere, Finland
{rodrigo.borges,otto.sahlgren,sami.koivunen,konstantinos.stefanidis, thomas.olsson,arto.laitinen}@tuni.fi

Abstract. Team assembly is a problem that demands trade-offs between multiple fairness criteria and computational optimization. We focus on four criteria: (i) fair distribution of workloads within the team, (ii) fair distribution of skills and expertise regarding project requirements, (iii) fair distribution of protected classes in the team, and (iv) fair distribution of the team cost among protected classes. For this problem, we propose a two-stage algorithmic solution. First, a multi-objective optimization procedure is executed and the Pareto candidates that satisfy the project requirements are selected. Second, N random groups are formed containing combinations of these candidates, and a second round of multi-objective optimization is executed, but this time for selecting the groups that optimize the team-assembly criteria. We also discuss the conflicts between those objectives when trying to understand the impact of fairness constraints in the utility associated with the formed team.

1 Introduction

Given a set of optimization criteria and constraints, team assembly targets at selecting, from a pool of candidates who each have a set of skills, a set of individuals that jointly fulfils the requirements of a predefined project. Decision-makers have to establish a clear understanding of project requirements and teams' envisioned tasks so that they can be translated into computationally tractable formal requirements, respectively, as well as choose between various ways of assigning candidates into teams [9]. Moreover, team assembly is often a socially, ethically and legally sensitive activity, especially when conducted in high-stakes domains, such as formal education or professional work contexts. A particularly salient set of concerns relates to unfair bias which can disadvantage members of protected groups (such as gender or ethnic groups) and marginalized communities. Whether technical or social in terms of its origin [7], the bias introduced in (or reproduced by) team assembly algorithms can result in unfair treatment of candidates in the team assembly process, even unlawful discrimination.

Existing work has developed methods for improving team-assembly algorithms in different respects, such as reducing the cost of team assembly [2], distributing the workload more equitably among candidates [1] and improving the representation of different demographic groups in the resulting teams [2].

https://doi.org/10.1007/978-3-031-42941-5_10

Whereas a large body of work is devoted to developing methods for identifying and mitigating wrongful bias and unfairness in algorithms and software [13], research that addresses these issues in the context of computational team assembly remains scarce. Most existing approaches are designed for incremental solutions where teams are formed by selecting one candidate in sequence after the other, and optimize only a single distributive desideratum. Our work is motivated by the observation of two problems with this approach. On the one hand, decision-makers often have multiple objectives that need to be balanced or prioritized [11], and it is unlikely that a single fairness-objective can capture a holistic set of contextual values relevant to a given team-assembly process. On the other hand, an incremental approach to team-assembly can be undesirable in certain team-assembly contexts, such as when choosing one candidate at time t_1 closes off the possibility to choose another more suitable candidate later at time t_2.

To address these issues, we formulate team-assembly as a multi-objective optimization procedure motivated by the assumption that fairness-aware team-assembly should achieve several objectives constitutive a more holistic notion of fairness in team-assembly. We describe our framework and illustrate its benefits by employing four criteria for fairness-aware team-assembly. Ideally, a team assembly algorithm would compare every possible team-composition in light of these criteria and choose the one that minimizes a target objective. However, this approach can be expensive especially when the candidate pool is large. To address this issue, we propose a two-step team-assembly procedure: First, a multi-objective optimization procedure is executed and the Pareto candidates that satisfy the project requirements are selected. Second, N random groups are formed containing combinations of these candidates, and a second round of multi-objective optimization is executed, but this time for selecting the groups that optimize the team-assembly criteria. The choice between teams is determined according to a combination of all fairness criteria. This algorithm is not as cheap as selecting the best candidates incrementally, and it is not as expensive as testing all possible groups that can be formed. Instead, the proposed algorithm filters the best candidates among the ones that fulfill the project requirements, and it forms several random groups containing these candidates. When selecting a group that is already formed one can directly access the fairness metrics, and it is easier for the algorithm to minimize a given criterion or a set of criteria.

2 Related Work

Team-Assembly. Research on computational team-assembly is diverse, partly due to the variety of application areas and computational approaches. [10] proposes a team recommender that groups individuals within a social network based on pre-defined skill requirements. [14] presents an approach to form and recommend emergent teams based on how software artifacts are changed by developers, while [17] proposes building teams based on the personality of the team members using a classifier to predict the performance of the constructed teams. [16] frames team-assembly as a group recommendation, where it forms a team of users, each of whom has specific constraints, and recommends items to that team.

Bias and Fairness in Team-Assembly. Research on fair machine learning developed various ways for identifying and addressing unfair bias. In most works, fairness is framed as a local resource allocation problem where a given good should distribute efficiently without violating some pre-defined fairness constraint(s). Examples of metrics include Statistical Parity [6], which requires that the distribution of positive outcomes is statistically independent of so-called protected attributes (e.g., gender), and Equalized Odds [8], which requires parity in group-relative error rates. Different techniques can be applied throughout the system pipeline to mitigate bias in data, algorithms, or output distributions [15]. While research on fairness specifically in computational team-assembly contexts remains scarce, there are some notable exceptions. For example, [4] formulates the task of team-formation as an instance of fair allocation: a procedure for assigning students to projects should involve fair division, which is defined in terms of balanced workloads and tasks in the resulting teams. Another example is [2], which examines the fair team-formation problem in an online labour marketplace. To the best of our knowledge, [12] presents the most similar setting to our work, exploring a problem where teams have multidisciplinary requirements and the selection of members is based on the match of their skills and the requirements. For assembling multiple teams and allocating the best members in a fair way between the teams, it suggests a heuristic incremental method as a solution to create team recommendations for multidisciplinary projects.

3 Motivation

Team Assembly as One-Shot Subset Selection. Our approach is designed for subset selection cases where a team is formed by choosing an optimal set of individuals from a larger set of candidates, where project-to-team fit is evaluated by considering project requirements and candidates' skills. Our motivation is that, subset selection has received comparably less attention in research on fairness in algorithmic decision-making (see, however, [5]). Also, existing approaches to fairness-aware team assembly have largely focused on an *incremental* approach to selecting candidates, which can undesirable or suboptimal in certain cases since the overall composition of the team can be known only by selecting all candidates. Hence, we address a gap in the research literature by focusing on subset selection in an *one-shot team assembly* setting.

Multi-Objective Fairness in Team Assembly. We approach fairness-aware team assembly from the perspective of multi-objective optimization, observing the limitations of previous works that employ a single fairness metric. In particular, using a single measure does not allow the decision-maker to evaluate resulting team-compositions from a holistic evaluative perspective nor to identify trade-offs that may arise between their (un)desirable properties [11]. Our approach takes these notions into account, and recognizes that team assembly procedures can be multi-faceted in terms of the values they should promote and the goods and opportunities that are distributed therein. For example, in real-life contexts of team assembly, the decision-maker is not only distributing access to the team, but also allocating tasks and responsibilities between accepted team-members. Our notion of *multi-objective fairness* captures this idea, and we use it to denote the general sentiment that multiple goods and opportunities should be distributed fairly with due regard also for the overall utility generated.

For fairness-aware team assembly, we apply 4 objectives: (a) *Fair Representation*: The distribution of protected attributes within a team should be fair in terms of being as equitable as possible. (b) *Fair Workload Distribution*: The distribution of tasks within a team should be fair in terms of being as equal as possible. (c) *Fair Expertise Distribution*: The distribution of skills within a team should be fair in terms of being as equal as possible. (d) *Fair Cost Distribution*: The distribution of the cost within a team should be as fair as possible considering the protected attributes associated with candidates. Each objective equalize some benefit, or resource that many consider important in team assembly.

4 Problem Formulation

Let $S = \{s_1, \ldots, s_m\}$ be a set of skills, A be a binary sensitive attribute that can assume values A_0 or A_1, $U = \{u_1, \ldots, u_k\}$ be a set of individuals, i.e., the candidate pool, and P be a set of requirements for a project, i.e., a subset of the skill set ($P \subset S$). An individual $u \in U$ is represented as a combination of a cost profile (u^S) containing the hiring cost associated with their skills, and a value (u^A) associated with a sensitive attribute. The cost profile is obtained through a function θ that returns the cost of a certain skill, for example, $u^S = (\theta(s_1, u), \theta(s_2, u), \ldots, \theta(s_m, u))$ represents user u according to their cost in skills $\{s_1, s_2, \ldots, s_m\}$. We assume a user has a certain skill as long as the cost associated with that skill is greater than 0.

Any set of more than 2 and less than $|U|$ individuals is considered a team T. And the number of project requirements that are fulfilled by a team is referred to as *coverage*. The aim of the team-assembly method is to select among all teams that cover the project requirements, the team that minimizes five objectives: team cost, workload uneven distribution, expertise uneven distribution, representation parity, and cost difference. These objectives are described next.

Team Cost. The total cost of hiring a team for a project is defined as:

$$Cost(T, P) = \sum_{u \in T} \sum_{j=1}^{|S \cap P|} \theta(s_j, u). \tag{1}$$

It can be described as the summation of the cost associated with each team member's skills that match the project requirements. It is worth mentioning that one candidate can contribute to more than one task in the project, in the case when their skills coincide with more than one project requirement.

Workload Uneven Distribution is calculated as the standard deviation of the cost associated with each member of the team:

$$Workload(T, P) = \sqrt{\frac{1}{|T|}\sum\nolimits_{u \in T}\left(\sum\nolimits_{j=1}^{|S \cap P|}\theta(s_j, u) - \frac{Cost(T)}{|T|}\right)^2}. \tag{2}$$

A team in which the total cost is well distributed among members (low variance) is considered fair, whereas a team in which the cost is concentrated among a few members (high variance) is considered unfair.

Expertise Uneven Distribution. It is important to ensure that not just the costs are well distributed among candidates, but that the costs are also well distributed among project requirements. The unevenness of expertise distribution is calculated as:

$$Expertise(T, P) = \sqrt{\frac{1}{|P|}\sum\nolimits_{j=1}^{|S \cap P|}\left(\sum\nolimits_{u \in T}\theta(s_j, u) - \frac{Cost(T)}{|P|}\right)^2}. \tag{3}$$

In a similar fashion to the workload distribution, this objective measures the standard deviation of the cost associated with each project requirement. A fair team is expected to distribute their cost among requirements as even as possible, thus resulting in a low standard deviation value.

Representation Parity. It measures the difference between the occurrences of A_0 and A_1 within a team as potential values for a sensitive attribute A. The objective is calculated as:

$$Representation(T, A) = \frac{\sqrt{(|f(T, A_0)| - |f(T, A_1)|)^2}}{|T|}, \tag{4}$$

where function $f(T, A_0)$ returns a set containing the members of T associated with sensitive attributes A_0, as well as in the case of attribute A_1. A low Representation Parity indicates a fair distribution of attribute A whereas a high value indicates a majority of members associated with one of the classes, A_0 or A_1.

Cost Difference. It measures the difference between the cost allocated to two categories, A_0 and A_1, within a team. The total cost of team members associated with a certain sensitive attribute, named Cost Attribute (CA), is calculated as:

$$CA(T, P, A_0) = \sum\nolimits_{u \in f(T, A_0)}\sum\nolimits_{j=1}^{|S \cap P|}\theta(s_j, u), \tag{5}$$

in the case of attribute A_0. And the Cost Difference objective is calculated as:

$$CostDiff(T, A, P) = \frac{\sqrt{(CA(T, P, A_0) - CA(T, P, A_1))^2}}{Cost(T, P)}. \tag{6}$$

As mentioned before, our goal is to select a team T that fulfils the project P requirements and that minimizes the multi-objective condition:

$$\text{argmin}_T(Cost(T,P), Workload(T,P), Expertise(T,P), Representation(T,A),$$
$$CostDiff(T,A)).$$

4.1 Multi-Objective Fairness in Team Assembly

In this section, we propose a method designed for assembling teams with multiple fairness constraints. The method assumes a pool of candidates from which the team will be selected, a project and a sensitive attribute associated with each of the candidates. The method formulates fairness-aware team-assembly as a multi-objective optimization problem that is performed in two stages: first, project requirements are considered as objectives, and the best candidates are selected for the next phase. Second, multiple teams are formed with these candidates and fairness constraints are calculated for each of the teams. This time the fairness constraints are assumed as objectives, and the team that minimizes these constraints while fulfilling the project requirements is selected as the fairer.

Given a candidate pool (U) and set of project requirements (P), the first action is to filter candidates with at least one skill required by the project. The filtering process removes all users for which $|u \cap P| = 0$, and the remaining candidates are referred to as U_p. The candidates in U_p are then submitted to a multi-objective optimization step with the aim of selecting the best candidates for this specific project according to the Pareto dominance concept. According to this concept, a candidate dominates another if they perform better in at least one of the project requirements. A candidate is considered non-dominated if they are not dominated by any other candidate in the population, and the set of all non-dominated candidates compose the *Pareto candidates* subset.

At this point, the *paretoCandidates* subset contains the non-dominated candidates considered as the most suitable for the given project, but our notion of a fair team can only be assessed when having a formed team. The next step is to form a reasonable amount of teams containing a fixed number of Pareto candidates selected randomly. The number of random teams (N) as well as the size of these teams (M) are provided as parameters for the method. Once the teams are formed, it is possible to calculate their coverage, as well as their fairness objectives with Eqs. 1, 2, 3, 4 and 6. The teams that fulfil the project requirements are filtered out and considered in the following step. $Cost$, $Workload$, $Expertise$, $Representation$ and $CostDifference$ are then calculated, and each team end up being represented according to the values calculated for its objectives. A second round of multi-objective optimization is executed. This time, teams are being compared instead of candidates, with the same understanding as in from the first case. Given two arbitrary teams, two are the possibilities: (i) one team dominates the other if it has at least one objective with a lower value than the other team, or (ii) the two teams do not dominate each other. The non-dominated teams are referred to as *Pareto teams*. Finally, all objectives measured for Pareto teams are summed up as an indicator of an *overall unfairness*, and the method selects the team T with the lower unfairness value.

5 Experiments

We evaluated the proposed method in a dataset obtained from the *freelancer*[1] website, in which candidates register themselves to be hired as freelance workers. The dataset contains 1,211 candidates who self-declared their costs and their expertise in 175 skills [2]. In the information available in the dataset, users are associated with skills in a binary fashion, but no information is provided about the cost of each skill separately. We decided that the cost declared by the users is the same for every skill in which they have the expertise, meaning that if user u declared a cost c and they are hired for a project in which they will contribute with two skills, then the total cost associated with this user is $2 \times c$. [2] attributes a hypothetical binary sensitive attribute to each candidate, and generates several versions of the same dataset, associating candidates with this attribute in different proportions. We decided to use the dataset in which members are equally represented in the candidate pool (50/50), and the dataset in which members are more unevenly represented (10/90). The dataset contains also the requirements for 600 projects.

We applied two other team-assembly methods to the same task for the sake of comparison. The first method, named *Incremental*, selects the most suitable candidates incrementally until the project requirements are fulfilled. The second method, named *Fair Allocation*, operates in a similar fashion, but this time

Table 1. Cost, Workload, Expertise, Representation, Cost Difference, and number of formed teams for Incremental, Fair Allocation and Multi-Objective methods. Multi-Objective can optimize different criteria, and its results are presented according to seven objectives: Random, Top-Cost, Top-Workload, Top-Expertise, Top-Representation, Top-Cost Difference and Top-Sum. The best results for each objective are in boldface, and the second-best results are in underlining.

Classes Dist.	Algorithm		Cost	Workload	Expertise	Representation	Cost Difference	Teams
50/50	Incremental		**23.311 (15.548)**	0.035 (0.034)	0.025 (0.026)	0.480 (0.378)	0.610 (0.345)	486
	Fair Allocation		29.201 (20.453)	0.042 (0.040)	0.032 (0.032)	0.238 (0.268)	0.447 (0.286)	486
	Multi-Obj.	Random	40.862 (23.391)	0.045 (0.045)	0.035 (0.036)	0.343 (0.339)	0.419 (0.351)	506
		Top-Cost	26.099 (16.876)	0.035 (0.038)	0.026 (0.027)	0.450 (0.360)	0.595 (0.333)	506
		Top-Workload	42.294 (27.074)	**0.018** (0.029)	0.032 (0.036)	0.434 (0.367)	0.471 (0.357)	506
		Top-Expertise	37.943 (23.646)	0.039 (0.039)	**0.011 (0.020)**	0.445 (0.363)	0.525 (0.359)	506
		Top-Repres.	28.503 (17.531)	0.036 (0.038)	0.028 (0.027)	**0.143 (0.170)**	0.393 (0.251)	505
		Top-Cost Diff.	41.458 (25.980)	0.040 (0.040)	0.037 (0.036)	0.188 (0.197)	**0.091 (0.196)**	506
		Top-Sum.	39.147 (22.969)	0.032 (0.035)	0.028 (0.029)	0.144 (0.174)	0.107 (0.206)	506
10/90	Incremental		**23.423 (15.693)**	0.035 (0.034)	0.025 (0.026)	0.758 (0.341)	0.875 (0.239)	486
	Fair Allocation		31.212 (20.802)	0.046 (0.040)	0.034 (0.030)	**0.413 (0.370)**	0.600 (0.328)	484
	Multi-Obj.	Random	40.939 (25.097)	0.044 (0.047)	0.035 (0.038)	0.722 (0.354)	0.748 (0.350)	506
		Top-Cost	26.159 (17.047)	0.035 (0.038)	0.026 (0.027)	0.729 (0.347)	0.843 (0.253)	506
		Top-Workload	42.525 (27.463)	**0.018 (0.029)**	0.033 (0.036)	0.828 (0.297)	0.836 (0.285)	505
		Top-Expertise	38.140 (23.855)	0.039 (0.040)	**0.011 (0.020)**	0.795 (0.327)	0.842 (0.298)	506
		Top-Repres.	30.231 (19.608)	0.039 (0.041)	0.030 (0.032)	0.435 (0.390)	0.594 (0.353)	506
		Top-Cost Diff.	34.961 (22.256)	0.040 (0.039)	0.032 (0.031)	0.470 (0.377)	**0.459 (0.439)**	506
		Top-Sum.	37.661 (23.700)	0.035 (0.038)	0.027 (0.030)	0.432 (0.389)	0.460 (0.436)	505

[1] https://www.freelancer.com/.

considering users associated with a sensitive attribute and forcing as much as possible that the formed team has a fair distribution of this attribute among its members. For more details, see at [3].

Multi-Objective, Incremental and Fair Allocation were evaluated for forming a team for each of the 600 projects in the freelancer dataset, and the average value obtained for each of the objectives presented in Sect. 4 were calculated. The results are reported in Table 1, separately for the two datasets containing different proportions of the sensitive attributes, 50/50 and 10/90. The results of Multi-Objective are presented according to five different optimization objectives, named configurations: Top-Cost, Top-Workload, Top-Expertise, Top-Representation and Top-Cost Difference, along with a Random selection variation. On average, the first round of multi-objective optimization reduced the number of candidates by 77%, meaning that the candidates dominated others with compatible skills represent 23% of the total. In the second round of multi-objective optimization, the teams were reduced by 98% on average. The teams that dominate the others represent only 2%. This reflects how much the teams formed randomly can be internally equivalent or redundant.

In general, the Incremental method was the most efficient in minimizing the total cost and the size of the formed teams. The Multi-Objective method was able to assemble a slightly higher number of teams than other methods, probably because of forming teams in one-shot instead of incrementally. When configured to minimize a specific objective, the Multi-Objective method was efficient in selecting the teams, except when the objective was the Cost, and when the objective was the Representation and the dataset contained an uneven distribution of classes (10/90). In the former case, Incremental was the most efficient method, and in the latter case, the Fair Allocation performed better.

In the context of Multi-Objective fairness, it is preferable that a team presents a good balance of objectives instead of an extremely low value for one objective despite the others. E.g., when configured to optimize the Expertise objective (Top-Expertise) in the uneven dataset (10/90), Multi-Objective selected, on average, teams with a fairly high Representation (0.795) and Cost Difference (0.842) values. The Top-Sum configuration, on the other hand, selected teams with the second-best average values (highlighted with underline in Table 1) for three out of five objectives, for both datasets. The two objectives in which the configuration did not perform well were Cost and Expertise, which leads us to the next step of analyzing potential conflicts between objectives in the results.

Tension Between Objectives. Conflicts between objectives might emerge in situations when one objective is minimized, and another (or a set) goes up as a side effect. We noticed that minimizing the Workload objective increases the total cost of teams in both datasets (see Top-Workload configuration of Multi-Objective in Table 1). These were the configurations in which the average cost of selected teams presented the highest values, 42.294 in the case of the 50/50 dataset, and 42.525 in the case of the 10/90 dataset. These values are 181.4% and 181.5% higher than the lowest cost obtained in the results, respectively. Workload and Expertise, however, are fairness objectives that do not take into account the sensitive attribute associated with candidates, differently from Representation and Cost Difference, which are both calculated according to how many team members belong to each of the classes derived from this attribute. Top-Sum configuration of Multi-Objective performed especially well regarding those two objectives, obtaining relatively close values to the lowest ones obtained in the experiments. In general, ensuring an equal distribution of costs among protected groups (Cost Difference) had a bigger impact on the total cost than ensuring that both groups are equally represented in the teams (Representation).

Impact of Class Distribution. The trade-offs between objectives as well as their absolute values can vary depending on how the classes derived from the sensitive attribute are distributed within the candidate pool. Workload and Expertise objectives were not impacted by the difference in the datasets, but Representation and Cost Difference, on the other hand, presented substantially different values depending on the proportion in which users are distributed in classes. The lowest average value calculated for the Representation objective increased from 0.143 in the 50/50 dataset, to 0.413 in the 10/90 dataset, an increase of approximately 289%, and the Cost Difference objective increased even more, its lower value went from 0.091 to 0.459, an increase of more than 500%. When focusing on Top-Sum configuration of the Multi-Objective team-assembly method, a significant decrease of approximately 82.5% (from 0.610 to 0.107) was observed in the Cost Difference if compared to the Incremental method, and of approximately 76% if compared to the Fair Allocation method, in the case of the 50/50 dataset. In the case of the 10/90 dataset, however, the differences were slightly different, the Cost Difference was reduced by approximately 23.4% (from 0.6 to 0.46) when compared to the Fair Allocation method, but the Representation went higher, increasing some 4.6% (from 0.413 to 0.432). If compared to the Incremental method, the Top-Sum configuration performed better on those objectives: the Cost Difference was reduced by 47% (from 0.875 to 0.460), and the Representation was reduced by 40% (from 0.758 to 0.432).

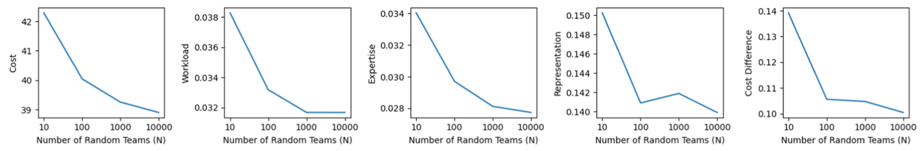

Fig. 1. The impact of N in cost, workload, expertise, representation and cost difference.

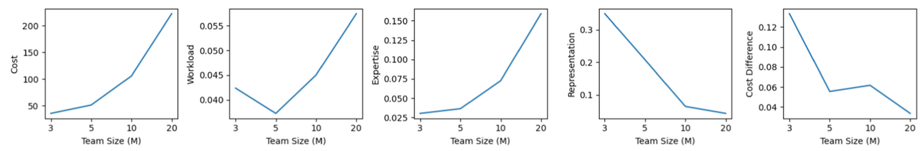

Fig. 2. The impact of M in cost, workload, expertise, representation and cost difference.

Impact of the Number of Random Teams. Multi-Objective receives N and M as parameters. First, N was set equal to 10, 100, 1,000 and 10,000, and the impact of these decisions on the team-assembly objectives can be seen in Fig. 1. One could expect that the number of random teams has a direct impact on the probability of the method forming a team that fulfils the team-assembly criteria, simply because when there are more teams there are more options to choose from. But this holds true to a certain limit, after which the improvement in the results is ordinary. We decided to configure the method to form 1,000 random teams once this is a point where all curves curve get more stable, as one can see in Fig. 1. M was then set equal to 3, 5, 10 and 20, and the impact of these decisions on the team-assembly objectives can be seen in Fig. 2. It is evident how the Cost, Workload and Expertise objectives increase as the teams get bigger, except for the workload distribution when teams are formed with 5 members. In this case, the workload gets slightly better distributed among team members than when teams are formed with 3 members. On the other hand, Representation and Cost Difference objectives present the opposite behaviour, they get lower as the teams get bigger, probably because it is more likely to get even distributions when there are more members to distribute among classes.

6 Conclusions

In this paper, we argued in favour of a wider notion of fairness in the context of team assembly by framing the task of forming teams as a multi-objective optimization procedure. We have also proposed an algorithm for assembling teams with multiple fairness constraints that assembled teams in a one-shot fashion, as opposed to incremental methods proposed previously in the literature. Our method is flexible enough that it can be applied to situations when one single objective needs to be minimized (or maximized), as well as in situations when all objectives need to be optimized jointly.

References

1. Anagnostopoulos, A., Becchetti, L., Castillo, C., Gionis, A., Leonardi, S.: Online team formation in social networks. In: WWW (2012)
2. Barnabò, G., Fazzone, A., Leonardi, S., Schwiegelshohn, C.: Algorithms for fair team formation in online labour marketplaces. In: Companion of WWW (2019)
3. Borges, R., Sahlgrens, O., Koivunen, S., Stefanidis, K., Olsson, T., Laitinen, A.: Computational team assembly with fairness constraints. CoRR abs/2306.07023 (2023)
4. Bulmer, J., Fritter, M., Gao, Y., Hui, B.: FASTT: team formation using fair division. In: Canadian AI (2020)
5. Cachel, K., Rundensteiner, E.: Fins auditing framework: group fairness for subset selections. In: AAAI/ACM Conference on AI, Ethics, and Society (2022)
6. Dwork, C., Hardt, M., Pitassi, T., Reingold, O., Zemel, R.: Fairness through awareness. In: Innovations in Theoretical Computer Science Conference (2012)

7. Friedler, S.A., Scheidegger, C., Venkatasubramanian, S.: The (im)possibility of fairness: different value systems require different mechanisms for fair decision making. Commun. ACM **64**(4), 136–143 (2021)
8. Hardt, M., Price, E., Srebro, N.: Equality of opportunity in supervised learning. In: NIPS (2016)
9. Harris, A.M., Gómez-Zará, D., DeChurch, L.A., Contractor, N.S.: Joining together online: the trajectory of CSCW scholarship on group formation. Proc. ACM Hum. Comput. Interact. **3**(CSCW), 1–27 (2019)
10. Lappas, T., Liu, K., Terzi, E.: Finding a team of experts in social networks. In: SIGKDD (2009)
11. Lee, M.S.A., Floridi, L.: Algorithmic fairness in mortgage lending: from absolute conditions to relational trade-offs. Mind. Mach. **31**(1), 165–191 (2021)
12. Machado, L., Stefanidis, K.: Fair team recommendations for multidisciplinary projects. In: WI (2019)
13. Mehrabi, N., Morstatter, F., Saxena, N., Lerman, K., Galstyan, A.: A survey on bias and fairness in machine learning. ACM Comput. Surv. **54**(6), 1–35 (2021)
14. Minto, S., Murphy, G.C.: Recommending emergent teams. In: MSR (2007)
15. Pitoura, E., Stefanidis, K., Koutrika, G.: Fairness in rankings and recommendations: an overview. VLDB J. **31**, 431–458 (2021). https://doi.org/10.1007/s00778-021-00697-y
16. Stefanidis, K., Pitoura, E.: Finding the right set of users: generalized constraints for group recommendations. In: PersDB (2012)
17. Yilmaz, M., Al-Taei, A., O'Connor, R.V.: A machine-based personality oriented team recommender for software development organizations. In: EuroSPI (2015)