FERN: Fair Team Formation for Mutually Beneficial Collaborative Learning

Maria Kalantzi, Agoritsa Polyzou, and George Karypis

Abstract—Automated *Team Formation* is becoming increasingly important for a plethora of applications in open source community projects, remote working platforms, as well as online educational systems. The latter case, in particular, poses significant challenges that are specific to the educational domain. Indeed, teaming students aims to accomplish far more than the successful completion of a specific task. It needs to ensure that all members in the team benefit from the collaborative work, while also ensuring that the participants are not discriminated with respect to their protected attributes, such as race and gender. Towards achieving these goals, this work introduces FERN, a fair team formation approach that promotes mutually beneficial peer learning, dictated by protected group fairness as *equality of opportunity* in collaborative learning. We formulate the problem as a multi-objective discrete optimization problem. We show this problem to be NP-hard and propose a heuristic hill-climbing algorithm. Extensive experiments on both synthetic and real-world datasets against well-known team formation techniques show the effectiveness of the proposed method.

Index Terms—Team formation, Group fairness, Collaborative learning, Hill-climbing, Crowd-sourcing, Clustering, Partitioning.

I. INTRODUCTION

EXTENSIVE research has unveiled the benefits of collaborative learning through participation in group educational activities (e.g., study groups, team projects, etc.) [1], [2], [3], [4], [5], [6], especially when the members of the teams can learn something from their peers (peer learning) [6], [7], [5], [4]. This stands for traditional brick-and-mortar educational institutions, as well as online learning environments and massive open online courses. To facilitate the creation of such collaborative learning groups, *team formation* algorithms are used to group students based on different factors and constraints. Since students have the opportunity to benefit through peer learning, it is important for the team formation algorithms to ensure that the students are treated fairly with respect to benefit from team work, irrespective of any protected attributes, such as gender and race.

There has been considerable research in developing methods to form teams in the educational domain, some of which try to account for different notions of fairness. Some works implicitly address fairness when they aim to produce balanced teams with respect to the skills of their participants (inter-team homogeneity) [8], [9], [10], while others also account for

M. Kalantzi and G. Karypis are with the Department of Computer Science and Engineering, University of Minnesota, Twin Cities, MN, USA. E-mail: {kalan028, karypis}@umn.edu.

A. Polyzou is with the Massive Data Institute and Computer Science Department, Georgetown University, Washington, D.C., USA. E-mail: ap1744@georgetown.edu.

diversity in personality and gender [8]. Outside the educational domain, recent work on fair clustering methods considers equal representation of the protected groups across the clusters [11], [12], [13].

In this work, we are motivated by the *equality of educational opportunity* [14], where every student should have equal educational opportunities irrespective of race, gender, socioeconomic class, sexuality or religion. We transfer this ideal of *equality of opportunity* in the context of collaborative learning and we strive to ensure that all students benefit equally from peer learning, regardless any protected attributes. Our goal is to create teams comprising students with the necessary set of skills for performing the target task, while maximizing the peer learning opportunities at both *individual* and *protected group* level. We consider cases where individuals can belong to different groups, based on a protected attribute such as race, demographics and socioeconomic status, and we focus on alleviating potential discrimination against the team members. Our approach is formalized as a multi-objective discrete optimization problem that captures the aforementioned objectives. We formally prove its NP-hardness, and we propose FERN; a heuristic hill-climbing greedy algorithm to tackle it. We experimentally evaluate the performance of FERN on both synthetic datasets with different characteristics, and real-world datasets, against well-known team formation techniques. The results show the effectiveness of the proposed method on creating fair and beneficial teams. To the best of our knowledge, this is the first work in the team formation domain that addresses group fairness in the form of equality of opportunity in collaborative learning.

II. PROBLEM STATEMENT

Given a set $\mathbb{S}$ of N students and a target task with specific skill requirements, we are interested in assigning the students to teams such that, in addition to completing the target task, every member in a team benefits. Following existing conventions [6], [7], [15], [16], we consider a student to benefit from participating in a team if there is at least one skill they can learn from their higher ability peers. In this case, we can create teams such that even high performing members could benefit, as long as there is a peer who has higher ability in at least one skill. In our setting, the number of teams and their size are not fixed and depend on the input data and the target task. We assume that a task requires multiple skills in order to be completed successfully. For example, different lower level courses could be required to complete a project in a higher level course.

Fig. 1: Example of the benefit of students A, B, C if we team them in two possible ways.

Skill requirements. A given target task with k required skills is represented by a k-dimensional skill requirements vector, $\mathbf{r} = (r_1, \ldots, r_k)$, whose p-th component, r_p, with $p \in \{1, \ldots, k\}$, is the threshold that the students from each team need to reach collectively, in order to complete the task with respect to skill p. Note here, that the requirements can be different. Large value for r_p indicates that skill p is of higher importance or that it requires higher ability from the students in order to complete the target task. Student i is represented by a k-dimensional vector, $\mathbf{s}^i = (s_1^i, \ldots, s_k^i)$, whose p-th element, s_p^i, with $p \in \{1, \ldots, k\}$, represents the student's ability for skill p. We consider a team l to complete the task successfully when the aggregated ability of its members is greater than or equal to the corresponding skill threshold value, for all the k skills, i.e.:

$$\sum_{i \in \mathbb{T}_l} s_p^i \geq r_p, \forall p = 1, \ldots, k,$$

where $\mathbb{T}_l$ is the set of students that belong to team l.

Individual benefit. A student i benefits from a student j when j has higher ability in at least one skill. We formally define this as follows:

Definition 1. *A student i with skill vector $\mathbf{s}^i$ **benefits** from a student j with skill vector $\mathbf{s}^j$, when there is at least one skill, $p \in \{1, \ldots, k\}$, such that $s_p^j - s_p^i > \varepsilon$.*

We use a threshold ε to distinguish among *significant* and *non-significant* differences in the skill values of the students. Next, we define the benefit matrix, $\mathbf{B} \in \mathbb{R}^{N \times N}$, with $B_{ij} = 1$ if student i benefits from student j, and $B_{ij} = 0$, otherwise. Fig. 1 presents a toy example which highlights the definition of benefit. There are three students A, B, C, and their corresponding performance in three skills. If we group B with A then, A will learn skills 1 and 2 from B, and B will learn skill 3 from A. On the contrary, if we group B with C, student B will not benefit from C as B is performing better in every skill.

We then define the *individual benefit* of student i from their team, as the fraction of their teammates that i benefits from, i.e.:

$$IndBen(i) = \frac{1}{|\mathbb{T}^i|} \sum_{j \in \mathbb{T}^i} B_{ij}, \tag{1}$$

where $\mathbb{T}^i$ is the set of students that belong to the same team as i, excluding i. According to this definition, in order to increase the individual benefit of student i, we have to increase the number of teammates that i benefits from. In other words, we aim to create teams in which the members benefit from as many teammates as possible.

Group benefit. We assume that the students belong to m different groups based on a protected attribute, such as race, demographics and socioeconomic status. The benefit of a protected group q, with $q \in \{1, \ldots, m\}$, is the average individual benefit of its members, i.e.:

$$GBen(q) = \frac{1}{|\mathbb{G}_q|} \sum_{i \in \mathbb{G}_q} IndBen(i), \tag{2}$$

where $\mathbb{G}_q$ is the set of students that belong to protected group q. We define **group fairness** in the context of equality of opportunity in collaborative learning and we consider a team solution to be fair across the protected groups, if the different protected groups have approximately the same relative group benefit. In particular, for two protected groups q_1 and q_2, a team solution is fair when:

$$GBen(q_1) \approx GBen(q_2)$$

If the students that tend to benefit the least disproportionately belong to one protected group, then this corresponds to an unfair team solution.

In summary, we aim to form teams that fulfill the skill requirements of the target task, maximize the individual benefit and impose group fairness. This problem is a combinatorial optimization problem, with a large discrete configuration search space. Next, we show that this is an NP-hard problem.

Theorem 1. *The problem is NP-hard.*

Proof. In order to prove that our problem is NP-hard, we will reduce the Dual Bin Packing Problem (DBPP) [17] to a much simpler version of our problem. We consider the case where we have only one skill for each student and the goal is to assign the students to teams, such that, the number of teams which fulfill or exceed the skill requirements is maximized. We then map the bins to teams and the items to students. The skill s^i of student i corresponds to the size α_i of item i and the bin threshold t corresponds to the skill requirement threshold r. We want to find the maximum number of teams, L, and a partition of the students to teams, $\mathbb{T}_1 \cup \cdots \cup \mathbb{T}_L$, such that $\sum_{i \in \mathbb{T}_l} s^i \geq r, \forall l = 1, \ldots, L$. Thus, the DBPP is reduced to our problem and if we can find a solution in polynomial time to our problem, then we can find a solution to DBPP as well. But we know this is a contradiction since DBPP is NP-hard [17]. □

III. FERN: A Fair Team Formation Approach

We formulate the problem of fair team formation as a multi-objective discrete optimization problem and we present a heuristic algorithm to solve it. We will use the name FERN to refer to the approach and associated algorithm. FERN starts by computing an initial solution and then refines this solution to gradually improve the objective function.

A. Objective Function

Our goal is threefold: create teams such that 1) the collective ability of each team is adequate to successfully complete the target task, 2) the number of students that benefit from the team work is maximized, and 3) impose fairness among the protected groups. Next, we explain how we proceeded in order to achieve each goal.

Average skill deficiency. A team completes the target task successfully, if its collective ability reaches or exceeds the skill requirements threshold. In order to tackle this goal, we measure the *average skill deficiency* of a solution, which is given by:

$$\mathcal{X} = \frac{1}{Lk} \sum_{l=1}^{L} \sum_{p=1}^{k} \left(r_p - \min\left(r_p, \sum_{i \in \mathbb{T}_l} s_p^i \right) \right)^2, \qquad (3)$$

where L is the number of teams of the solution.

Average individual benefit. In order to tackle our second goal and maximize the number of students that benefit from collaborating with others in the same team, we measure the *average individual benefit* over the whole set of students $\mathbb{S}$:

$$\mathcal{Y} = \frac{1}{|\mathbb{S}|} \sum_{i \in \mathbb{S}} IndBen(i), \qquad (4)$$

Variance in group benefit. In order to achieve group fairness with respect to benefit among the m protected groups, we measure the *variance in group benefit*, which is given by:

$$\mathcal{Z} = \frac{1}{m} \sum_{q=1}^{m} \left(GBen(q) - \overline{GBen} \right)^2,$$
$$\text{where } \overline{GBen} = \frac{1}{m} \sum_{q=1}^{m} GBen(q). \qquad (5)$$

Combined objective function. At last, we combine Eq. 3, 4, 5 in a single multi-objective function and we solve the following minimization problem:

$$\mathcal{F} = \mathcal{X} - \gamma \mathcal{Y} + \delta \mathcal{Z}, \qquad (6)$$

where we want to minimize $\mathcal{X}$, $\mathcal{Z}$ and maximize $\mathcal{Y}$. We use parameters γ and δ to control the relative importance of individual benefit and group fairness, respectively.

The group benefit is the most novel component of our multi-objective function. The motivation behind our definition of group benefit (Eq. 2) and the way we measure group fairness (Eq. 5) is the following. The second term of the objective function (average individual benefit, $\mathcal{Y}$) maximizes the individual benefit of each student and implicitly ensures that the number of students that benefit is maximized. This means that the group benefit of each protected group will also increase. Then, the third term of the objective function (variance in group benefit, $\mathcal{Z}$) additionally takes care of the balance in group benefit across the protected groups. This corresponds to improving the least benefited group, since the second term will penalize any decrease in the benefit. Ultimately, our combined formulation is similar to the one of maximizing the group benefit of the least benefited protected group. To back up the above arguments, we present experimental results in Subsection VI-C7.

B. Initial Assignment Algorithms

We developed four different algorithms to compute the initial assignment of students to teams, which are used as the starting point for the refinement process.

1) Global Most-Benefit First (`GMBF`): This method creates one team at a time; it starts with the first team and keeps on adding students until the skill requirement threshold is reached for each skill, or there are no students left to select. Every time, it chooses student i who has not yet been selected and has the highest individual benefit if all the students were teamed up in a single team, i.e., $i = \arg\max_i \sum_{j \in \mathbb{S}} B_{ij}$ - *global* choice. This algorithm optimizes the average skill deficiency (the first term of the multi-objective function) and ensures that every team, except for probably the last one, will reach the threshold in order to complete the target task.

2) Local Most-Benefit First (`LMBF`): This method is similar to `GMBF`, with the difference now, that it chooses the student with the highest individual benefit based on the current team it creates - *local* choice. The first student of each team is selected to be the one with the global lowest individual benefit out of those not yet selected. This algorithm accounts for the first two terms of the multi-objective function at every local choice it makes.

3) Local Most-Benefit Fair First (`LMBFF`): This method extends the previous two and takes care of all terms of the multi-objective function. The difference compared to the two aforementioned algorithms is that, now the algorithm chooses to add to the current team, that student who optimizes the weighted summation of the average individual benefit and the variance in group benefit. Again, the first student of each team is selected to be the one with the global lowest individual benefit out of those not yet selected.

4) Random: This corresponds to a random assignment and serves as a baseline to compare against the above more sophisticated proposed initial algorithms. It randomly assigns equal number of students to each team. If this is not possible, the remaining students are equally split in the first N `modulo` L teams, where the number of teams L, is determined by `GMBF`.

C. Refinement Algorithms

We developed two hill climbing greedy approaches to perform the refinement process. They are best-first iterative algorithms that start from an initial sub-optimal solution, and attempt to find a better solution by making an incremental change at a time to the current solution. This is done repeatedly, until no further improvement can be found.

The change consists of moving student x from their current team T_s to team T_d; at each step, we choose the triplet (x, T_s, T_d) that gives the highest gain (optimal available move). The gain $\mathcal{G}_{x,T_s,T_d}$ of a move (x, T_s, T_d) is the reduction on the objective if we move student x from their current team T_s to team T_d, i.e., $\mathcal{G}_{x,T_s,T_d} = \mathcal{F}_{x \in T_s} - \mathcal{F}_{x \in T_d}$. The gain is positive if the move improves the value of the objective function, negative if the move deteriorates the solution, and equal to zero if the gain does not change.

The first algorithm `SAHC`, is a greedy steepest-ascent hill climbing, that searches the whole space of possible moves and performs the best one which leads to a *strictly positive* gain; if no such move exists, it terminates. The pseudo code is presented in Algorithm 1.

Algorithm 1 Steepest Ascent Hill Climbing (SAHC)

Require: initial assignment T, initial number of teams L.
1: $pq \leftarrow$ compute gains   ▷ priority queue with gain values for all possible moves
2: **while** 1 **do**
3:    $gain, student, team_s, team_d \leftarrow pq.pop()$ ▷ move with the maximal gain
4:    **if** $gain > 0$ **then**
5:       Perform the move and update T
6:       Update gains $\forall student_i \in S$
7:    **else**
8:       **break**
9:    **end if**
10: **end while**
11: Remove empty teams from L. Check for teams with only one student and find the best other team to assign this student to.
12: **return** T, L

The second algorithm is based on the Fiduccia-Mattheyses (FM) algorithm [18]; an iterative heuristic for hyper-graph bi-partitioning. This FM-based hill climbing, FMHC, is different from the steepest-ascent as it allows for uphill moves, i.e., moves that do not improve the value of the objective function, and even worsen it. It may perform *bad* moves with the goal of escaping from *local minima*. In particular, at each pass, FMHC examines all possible moves and greedily chooses the one with the optimal gain (it could be a negative one), until there are no more students to move. After it makes a move, it *locks* the student and does not consider them again in the same pass. It considers again the locked students in the next pass, and only then, it updates their gains. Next, FMHC finds the number of subsequent moves z, from the beginning of the current pass, that maximizes the sum of the gains of the z moves, $gain_{max}$. This translates to identifying a set of z concurrent moves (including potential non-beneficial single moves) that altogether will lead to a better solution. If $gain_{max}$ is larger than ϵ, then it accepts the z moves, updates the assignment of students to teams and improves the solution. If $gain_{max}$ is smaller than or equal to ϵ, then the refinement terminates. The pseudo code is presented in Algorithm 2.

For both algorithms, we implemented a max-heap priority queue to store the gains efficiently. During the refinement, the algorithms are allowed to empty a team and consequently, reduce the number of teams, if this leads to a better solution. At the end of the refinement process, we perform a post-processing step to eliminate teams with only one student; we assign the student to the best team, based on the value of the objective function.

Time Complexity Analysis. The most computationally expensive part in both the refinement algorithms is the update of the gains in the priority queue after we make a move. In the priority queue, we store in total $N \times (L-1)$ moves for the N students and L teams. The update of all the gains costs $O(NL\log(NL))$. Thus, the time complexity of SAHC for each iteration of the while loop is $O(NL\log(NL))$. For FMHC,

Algorithm 2 FM-based Hill Climbing (FMHC)

Require: initial assignment T, initial number of teams L.
1: $pq \leftarrow$ [ ] ▷ priority queue that stores the gain values for all possible moves
2: **while** 1 **do**
3:    $pq \leftarrow$ compute gains
4:    $locked \leftarrow set(), T_s \leftarrow T, moves \leftarrow$ [ ]
5:    **while** pq **do**   ▷ while there are students to move
6:       $gain, student, team_s, team_d \leftarrow pq.pop()$ ▷ move with the maximal gain
7:       $locked.add(student)$
8:       Remove from pq all entries that correspond to $student$
9:       Update T_s as if we make the move
10:       Update gains $\forall student_i \in (S - locked)$
11:       $moves.append([[student, source, dest], gain])$
12:    **end while**
13:    $z \leftarrow 0, gain_{max} \leftarrow -\inf$
14:    Find z that maximizes $gain_{max}$, the sum of $moves[1][1], \ldots, moves[z][1]$
15:    **if** $gain_{max} > \epsilon$ **then**
16:       Update T with the z moves
17:    **else**
18:       **break**
19:    **end if**
20: **end while**
21: Remove empty teams from L. Check for teams with only one student and find the best other team to assign this student to.
22: **return** T, L

---

the overall time complexity of one pass is $O(N^2L\log(NL))$, since the inner while loop (line 5) is executed N times.

IV. RELATED WORK

Agrawal et al., [15] were the first to introduce a computational approach for team formation in the educational setting. They form teams of *leaders* and *followers* (those with ability above and below the team average, respectively) to maximize the total gain. They measure the gain as the number of students that can increase their ability by interacting with leaders. Another work by Agrawal et al., [16] is built upon the intuition that every student can increase their benefit up to a reference score and extends the previous work [15] which assumes that only students below the team mean can benefit. In such formulation, every student could potentially increase their gain except for the best performing students of each team. Both these works consider one-dimensional skill vectors, do not incorporate the notion of the skill requirements and are limited to maximize the benefit of a subset of students.

Bandyopadhyay et al., [9] propose a method based on the Modularity Optimization technique, used in community detection to group a collection of objects. Their goal is to maximize the intra-team diversity and minimize the inter-team diversity, by maximizing the pairwise distance between the individuals. Liu et al., [10] propose a method to extract features for

students' skill proficiency by performing a cognitive analysis, and use these features to create the students' vectors. They also propose the *uniform k-means* method for creating a fixed number of teams K, that maximizes collaborative learning. This approach uses a variant of k-means clustering method to create k equal-sized clusters based on the students' vectors and then, randomly distribute the students of each cluster to K teams. The goal is to create heterogeneous teams by maximizing an objective based on dissimilarities. Methods like these, which promote the intra-team distances, could end up hurting both the individual-level and potentially the protected group-level benefit (unfair solution): the high ability students will be teamed up with lower ones, and the former will not benefit. Our objective function is entirely different from the aforementioned techniques, and the proposed approaches cannot be directly applied to solve our problem.

Bahargam et al., [19] defined the Guided Team-partitioning problem; given a set of points, they want to partition them into teams such that the centroid of each team is as close as possible to a specific target vector. They consider multiple target tasks with different requirements. This work is different from ours in that there are multiple target tasks and the team size and number of teams are fixed. The major difference though is that it allows for the removal of students out of the assignment solution. This is not allowed in our problem and cannot be applied in real-world educational settings.

Other works take into account the social network of the members in addition to the skill values, such that the communication cost is minimized and they are interested in forming a single team for the target task [20], [21], [22]. Andrejczuk et al., [8] consider the problem of forming *synergistic* teams, where each team is balanced in terms of ability, personality, and gender. The final equal-sized teams are diverse (equal representation) and of equal performance. One of the proposed methods is a local search algorithm that is similar to our hill-climbing method, but with significant differences: 1) their approach is not greedy, 2) it performs different kind of moves than ours, and 3) it allows for only a limited number of consecutive bad moves. Finally, they optimize a different objective function as they do not account for skill requirements or fairness in the form of equality of opportunity (we point out that the latter stands for all the related work), which makes this method unsuitable to solve our problem. Machado et al., [23] also account for fair teams, but in the notion again of forming teams of equal performance: the best-skilled members are equally distributed across the teams. Another difference with our work is that, even though they too build their problem upon members' skills and project requirements, they do not use numerical values; instead, they use text tags and assume that a single member can fulfill one skill. Last, they target multiple tasks and assign the best team to each task.

Genetic algorithms have also been proposed for solving the team formation problem [24], [21], [22], [25]. They are used for optimization problems and they incorporate the idea of evolution. They model the solutions as chromosomes and apply recombination operations (crossover and mutation) in order to preserve only the necessary information that will lead to a better solution [24].

Crowd-sourcing. Another related field is that of the standard crowd-sourcing. Within that context, given a set of workers with specific skills and cost, the goal is to form a single team for a target task. Mishra et al., [25] study a variation of the problem which does not include any cost on the workers' side. They use the notion of *dominance* to form a ranking of workers out of a pool of available workers, in order to choose the best ones for the target task, that are non-dominated with each other, i.e., they are equally preferable. Their definition of dominance is related to our definition of benefit; in our context, we want to minimize the dominance in order to increase the benefit. However, we are interested in allocating all the individuals to teams, so that the task's skill requirements is reached. Yadav et al., [26] study the problem of splitting the workers into teams that satisfy some requirements, but each team is assigned a different task. The minimization objective is related to each worker's cost.

V. EXPERIMENTAL EVALUATION

A. Datasets

We evaluated the performance of FERN on two sets of datasets. The first is a set of synthetically generated datasets of different size, complexity, and solution difficulty. The second is a set of datasets that we derived from actual students' course registration and performance data. Details on how we generated the synthetic and real-world datasets are provided next.

1) Synthetic Datasets: We developed a synthetic dataset generation method with five different parameters: (i) the number of students N, (ii) the dimension of the skill vectors k, (iii) the number of protected groups m, (iv) a vector of length m specifying how the N students are distributed across the m groups and, (v) m sets of parameters that control how the skill vectors for students of each protected group are generated. Next, we describe the last type of parameters, (v).

We assume that the completion of a target task requires knowledge of specific courses and the ability of a student to complete the task is represented by the corresponding grades (skill values). We generate the data in such a way that the students fall into four *grade buckets* reflecting their overall average performance: A, B, C, and D. Based on our generation model, for each protected group, we follow two steps: first, we distribute the students to the grade buckets, and second, we generate the skill vectors of each grade bucket.

In the first step, we want to create the grade buckets' distributions. Our goal is to generate datasets of progressively increased difficulty. The shape of these distributions affects the difficulty of balancing the generated dataset with respect to benefit. When the distribution of the students to the grade buckets is different across the protected groups, then students from different groups have different performance. For example, in the case of two protected groups, most of the students of the first group could belong to bucket A and most of the students of the second group to bucket C. In such cases, it is harder for the algorithm to impose group fairness. To this end, we sample from a beta distribution and we create multiple datasets of increasing difficulty by varying the values of α and

TABLE I: Statistics of the synthetic datasets for two protected groups.

Dataset	Group 1						Group 2					
	A	B	C	D	α	β	A	B	C	D	α	β
D1	16.2	57.0	25.7	1.1	6.0	4.0	16.2	57.0	25.7	1.1	6.0	4.0
D2	42.3	51.0	6.6	0.1	8.0	3.2	7.4	59.4	32.2	1.0	7.0	5.5
D3	87.7	11.8	0.5	0.0	7.5	1.0	0.0	0.6	11.1	88.3	1.0	7.5

A, B, C, D are the grade buckets. The values of each bucket correspond to average percentages (%) of the students that belong to each particular bucket. α, β are the parameters of the beta distribution used for each dataset. The two protected groups are of equal size.

(a) MATH   (b) CSCI   (c) MENG

Fig. 3: Real data - distribution of the students to the four grade buckets based on their average performance in two courses ($k = 2$).
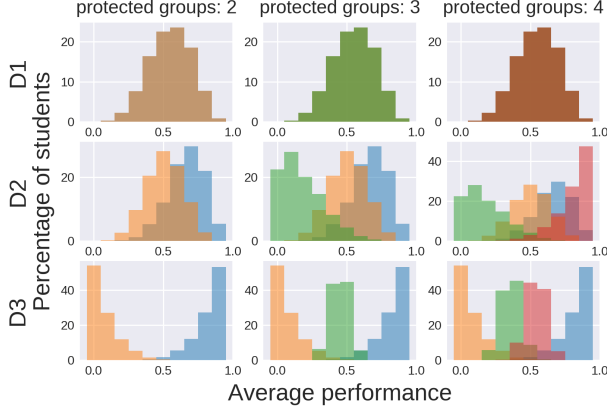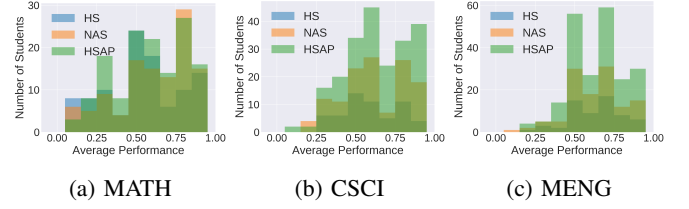

Fig. 2: Synthetic data - distribution of the students to the four grade buckets based on their average performance.

β parameters. We chose beta distribution as it is able to model grade distributions [27], can produce several different shapes, and is suitable to model random percentages. By definition, the beta distribution is defined on the interval $[0, 1]$; in order to correspond the sampled values to the four grade buckets, we discretize $[0, 1]$ to four bins. Table I summarizes the statistics of the generated data for two protected groups. Fig. 2 presents the shapes of the distributions for the synthetic data when the number of protected groups is equal to $2, 3, 4$. The different protected groups are of equal size. Dataset D1 is considered an easy dataset, as the different protected groups have the same bucket distributions, i.e., the students of different protected groups have similar performance. Dataset D2 is of middle difficulty, as the bucket distributions of the two groups are slightly different from each other. Finally, dataset D3 is the hardest one and represents an extreme case, where the two distributions are opposite with each other. In this case, it will be very difficult to impose group fairness, as there will be many high-skilled (bucket A) members of one protected group who are difficult to benefit from the lower-skilled (bucket D) members of the other group.

In the second step of the data generation process, we generate the students' skill vectors of each grade bucket. Since the skill values correspond to course grades, we map the letter grades $A - D$ to numeric ones. Then, we sample values from a normal distribution and we choose the mean to be the middle point of the corresponding numeric grade set of each grade bucket: 3.85 for A, 3.0 for B, 2.0 for C and 1.15 for D. In

all cases, we set the variance equal to 0.1. Last, we normalize the generated skill vectors, so that the values belong to the interval $[0, 1]$.

2) Real-world Datasets: We collected data from three departments of a large public university: Mathematics (MATH), Computer Science (CSCI), and Mechanical Engineering (MENG). The data include the grades of undergraduate students and span a period of 12 years. We selected the students who had received an A-D grade in the k most frequently taken courses from each major. We used the corresponding grades as the skill values for each student. Next, we transformed the letter grades to numerical values and normalized them in the range $[0, 1]$. We used 2, 4, and 6 courses, i.e., $k = 2, 4, 6$. In order to be able to explore what is the effect of the different values of k on the solution, we had to make sure that the set of students will be the same as we change k. Thus, we selected the students obtained when $k = 6$ (which is a subset of the students when $k = 2, 4$) and used those for the other values of k as well. For CSCI and MENG, where we had a large number of students, we selected the 400 most recent students, out of those that had taken the 6 most popular courses. For MATH and $k = 6$, we had 355 students.

The available data did not include any protected attributes. In order to assign the students to protected groups, we used the entry registration status and the number of credits transferred to simulate the *socioeconomic status* of the students. Our motivation came from the following findings. High school students can take Advanced Placement (AP) courses and transfer the credits earned to their undergraduate program. Minorities and low-income students are underrepresented in AP classes, and only a low percentage of them actually take and pass the AP exams every year [28], [29]. Regarding students coming from other institutions, we did not have information about the institution they transferred from. However, reports statistically show that almost half of these students come from 2-year colleges [30], which are considered a major access point to 4-year institutions for minority and low-income students [31]. Finally, our analysis led to three protected groups: high school students with less than 15 credits transferred (HS), high school students with more than 15 credits transferred (HSAP), and those coming from other institutions/colleges (NAS). Table II presents the statistics of the datasets for two courses ($k = 2$). Fig 3 presents the distribution of the students to the four grade buckets based on their average performance in two courses.

TABLE II: Statistics of the real-world datasets.

Dataset	N	HS					NAS					HSAP				
		N_1	A	B	C	D	N_2	A	B	C	D	N_3	A	B	C	D
MATH	355	102	30	42	22	8	113	57	32	18	6	140	57	46	34	3
CSCI	400	57	20	25	12	0	128	51	50	27	0	215	96	79	38	2
MENG	400	61	19	36	5	1	119	43	63	12	1	220	87	110	23	0

N is the total number of students and N_1, N_2, N_3 the number of students that belong to groups HS, NAS, and HSAP, respectively. A, B, C, D are the grade buckets.

B. Baseline Algorithms

We found that none of the available team formation algorithms that we listed in Section IV can be applied directly to solve our problem. This is because none of them accounts for the equality of opportunity in collaborative learning, as we do. As a result, they optimize an entirely different objective function. To this end, we implemented the following baseline approaches to compare against our proposed method FERN.

Random. This method is described in Subsection III-B4 and corresponds to a random assignment of the students to teams.

Uniform k-means (Umeans) [10]. This method (described in Section IV) creates heterogeneous equally-sized teams by maximizing a dissimilarity-based objective. We create $\lceil N/L \rceil$ clusters, where L is the number of teams generated by GMBF. This approach does not account for group fairness.

Genetic algorithm (GA). This approach is motivated by the genetic algorithms proposed for the team formation problem and described in Section IV, and it is adapted to directly optimize our multi-objective function, Eq. 6. We model the chromosome/solution as a vector of size N, i.e., the number of genes is N. We set the following parameters: mutation probability $= 0.1$, population size $= 200$, number of generations $= 300$, uniform crossover and swap mutation. For the number of teams L, we use the one that FERN identifies.

GMBF. This algorithm is described in Subsection III-B1 and is one of the algorithms that we experimented with to get an initial assignment. We do not use refinement process afterwards; this corresponds to a method like FERN without the refinement. This is an algorithm that optimizes the average skill deficiency (first term of the objective) and ensures that every team will reach the threshold, except potentially for the last one.

C. Experimental Setting

We try the following values for the importance parameters γ and δ: $\{0, 0.2, 0.5, 1, 2, 3\}$ The gain threshold that we use as the convergence criterion for FMHC is set to $\epsilon = 10^{-4}$. We also tried smaller values of ϵ, i.e., $\epsilon = 10^{-5}$ and $\epsilon = 10^{-6}$, but the gains in performance were negligible, whereas the running time was significantly increased, as the algorithm performed significantly more work until convergence. We run experiments for different number of students, i.e., $N = 50, 100, 200, 400$ for the synthetic datasets D1, D2, and D3.

Unless otherwise stated, we use the following set of parameter values: number of dimensions for the skill vectors $k = 2$, skill requirements threshold vector $\mathbf{r} = \mathbf{2}$, $\gamma = \delta = 1$ which corresponds to equal importance among the three terms of the

(a) D1 - SAHC

(b) D2 - SAHC

(c) D3 - SAHC

(d) D1 - FMHC
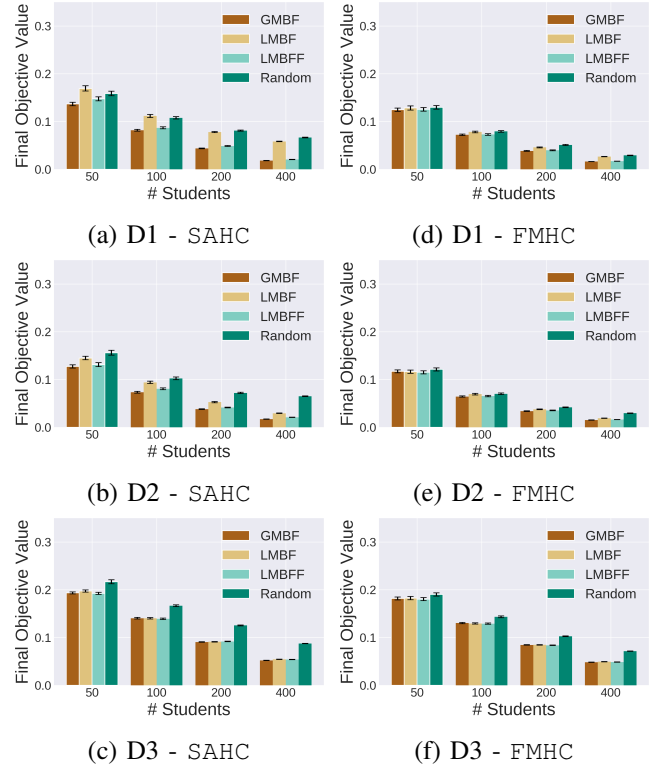
(e) D2 - FMHC

(f) D3 - FMHC

Fig. 4: Final value of the objective function starting from all four different initial assignments for both refinement algorithms SAHC and FMHC, in all three synthetic datasets. The smallest value the better.

objective function, $\varepsilon = 0$ for the benefit threshold, number of protected groups $m = 2$ for D1, D2, D3 and $m = 3$ for MATH, CSCI, MENG. For Umeans and GA, we run each experiment 10 times and we report the average performance. Finally, for each synthetic dataset, we run 50 experiments with different sampled data for the beta and normal distributions, generated by different 50 fixed for each particular dataset seeds, and we report the mean and standard error values.

VI. RESULTS

A. FERN's Optimization Algorithms

In order to explore how the different initial assignments, as well as the different refinement algorithms affect the final solution, we run experiments for both the refinement algorithms, starting from all four different initial algorithms presented in Subsection III-B. We evaluate the performance of the algorithms based on the final value of the objective function. The results are presented in Fig. 4.

Regarding the different initial assignment algorithms, Random performs the worst and should not be used; among the rest, there is little variation in performance. Regarding the different refinement algorithms, FMHC which allows for uphill moves, is more robust to the initial assignment and performs better than SAHC which gets easily trapped to local minima. Moreover, we see that as the number of students increases, we get better solutions. This is because, the more students we have, the higher flexibility we get to arrange them into teams

of high benefit. Another observation is that D3 is the most difficult one, and D1 the easiest one, verifying our assumptions on how to generate datasets of increasing difficulty. For the subsequent experiments, we use the `GMBF` initial algorithm and `FMHC` refinement and this combination refers to `FERN`.

B. Comparison Against Other Approaches

We compare the performance of `FERN` against that of competing approaches described in Subsection V-B, on the different datasets. The goal of this experiment is to evaluate how the different methods performed with respect to our threefold goal. In order to evaluate the performance with respect to the average skill deficiency, we report the percentage of teams that fulfilled the skill requirements. To evaluate the individual benefit of the final solutions, we report the average individual benefit (Eq. 4) percentage, which measures the percentage of teammates that a student benefits from on average. Last, in order to evaluate the group fairness, we report the variance in group benefit (Eq. 5). For D1, D2 and D3, we report the results for $N = 100$. Table III presents the results.

We can see that our method `FERN` is the best performing with respect to individual benefit in all datasets, as well as with respect to group fairness except for D1 in which `Umeans` and `GA` perform better. Regarding group fairness, we observe that `Umeans` is worse even than `Random`, except for D1 and MATH. This means that `Umeans` not only does not account for group fairness, but it additionally hurts it. Furthermore, D3 is the most challenging with respect to group fairness, and D2 is the second most challenging. Regarding the performance with respect to individual benefit, for the most difficult dataset D3, we see that on average, in `FERN`'s solution, every student benefits from 87% of their teammates, compared to only 56% in the case of `Umeans`. `GA` is better than `Umeans` in both individual and group benefit. However, `Umeans` is among the best performing with respect to skill deficiency, as the teams it forms are compact and concentrate adequate collective ability for the skill requirements. Last but not least, by comparing the performance between `FERN` and `GMBF`, we see that the refinement further improves the solution with respect to benefit. Overall, we see that `FERN` outperforms every other method with respect to benefit and this does not come with a significant cost for the skill requirements goal.

C. Sensitivity Analysis

In this Section, we use `FERN` and we explore how the problem and the solution change as we scale the values of the problem's features. We mainly focus on the synthetic data, because we are interested in testing datasets of varying difficulty with respect to benefit.

1) Distribution of the Students across the Different Protected Groups: With this set of experiments, our goal is to determine the impact of the relative sizes of the protected groups to the difficulty of the problem. We keep everything else fixed, except for the distribution of the students to the two protected groups. We test the following percentages: 50-50, 30-70 and 10-90. We present the results for D3 in Fig. 5 and we report the value of the objective function.

TABLE III: Comparison of the baselines with our method `FERN` on the different datasets.

Dataset	Method	Teams met skills	$\mathcal{Y}$	$\mathcal{Z}$
D1	Random	73.68	59.23	0.03
	Umeans	95.21	56.41	**0.00**
	GA	72.05	70.00	**0.00**
	GMBF	**98.21**	86.01	0.01
	FERN	80.43	**93.00**	0.01
D2	Random	77.98	58.85	164.48
	Umeans	**99.82**	55.65	228.22
	GA	75.05	68.79	35.74
	GMBF	98.61	86.36	2.05
	FERN	83.71	**93.77**	**1.47**
D3	Random	75.85	58.33	645.47
	Umeans	96.20	56.45	898.41
	GA	78.75	65.51	229.13
	GMBF	**98.74**	80.62	43.48
	FERN	85.09	**87.33**	**20.61**
MATH	Random	75.00	61.17	30.80
	Umeans	82.01	83.83	0.73
	GA	65.79	69.11	1.67
	GMBF	**100.00**	96.22	2.23
	FERN	96.43	**99.41**	**0.02**
CSCI	Random	76.70	59.92	4.16
	Umeans	99.10	58.43	12.38
	GA	65.79	69.11	1.67
	GMBF	**100.00**	97.64	0.01
	FERN	99.03	**99.82**	**0.00**
MENG	Random	76.92	64.13	0.63
	Umeans	98.50	64.78	2.55
	GA	68.21	75.98	0.87
	GMBF	**100.00**	98.18	0.95
	FERN	**100.00**	**99.75**	**0.00**

The third column (teams met skills) is the percentage of teams that fulfill the skill requirements. The fourth column ($\mathcal{Y}$) is the average percentage of teammates that a student benefits from. The last column ($\mathcal{Z}$) is the variance in group benefit (the values for group benefit correspond to percentages %). We used $\gamma = \delta = 1$ and $k = 2$. The best performance is marked in bold.

We observe that there is not significant difference in the quality of the final solution for the different sizes of the two protected groups. We only show the results for D3 as the trends are similar for D1 and D2. The only exception is for D3 and $N = 50$, where there is a slight variation among the different splits. A possible explanation is that 50 students is a relatively small input size for the algorithm to find stable solutions and is sensitive to the underlying characteristics of the data. In conclusion, the relative sizes of the protected groups do not seem to significantly affect the difficulty of the problem. To this end, for the remaining experiments, we use equally sized protected groups.

2) Effect of the Number of Protected Groups on the Group Benefit: In this section, we study how the group benefit varies as we increase the number of protected groups. Fig. 6 shows the results on D1, D2 and D3 for 2, 3, and, 4 protected groups, when we account for the group benefit ($\delta = 1$) and when we completely ignore it ($\delta = 0$). If we look at the results by column, we observe the following. First, as the number of

groups increases, the group fairness gets more difficult to fix (with the exception of D3 for $N = 50, 100$ when $\delta = 0$, most likely because of the extremity of the grade distribution for the 2 groups; see Fig. 2). Second, as expected, the final solutions become unfair when we do not explicitly account for the group fairness (second column). This stands especially for D2 and D3, in which cases the students of each group have different grade performance and as such, it is more challenging to balance the benefit across the protected groups.

3) Effect of the Dimensionality of the Skill Vectors: We experiment with different number of dimensions for the skill vectors, $k = 2, 4, 6$, and we evaluate the solution based on the final value of the objective function. The remaining parameters are set to the default values. We present the results for D3 in Fig. 7. We can see that the problem gets easier as we increase the number of dimensions. This happens because, for higher values of k, the degrees of freedom for the skill values is higher. As a result, it gets easier to maximize the benefit by arranging the students properly and the algorithm is able to find a very good and fair solution. The same results stand for the other datasets as well.

4) Effect of the Skill Requirements Threshold: We explore how the skill requirements vector $\mathbf{r}$ affects the difficulty of the problem and the number of teams. We run experiments for $\mathbf{r} = 2, 3, 4$ on D3 and we set the remaining parameters to the default values. In Fig. 8a, we plot the final value of the objective function with respect to $\mathbf{r}$. The higher the threshold, the more difficult the problem gets, as more students are needed in each team to fulfill the skill requirements and thus, it is harder to maximize the individual benefit. In Fig. 8b, we plot the number of teams the algorithm generates with respect to $\mathbf{r}$. As we increase $\mathbf{r}$, the number of teams decreases because the average number of students per team increases. The trends are exactly the same for all the datasets.

5) Effect of each Benefit Term on the Final Solution: How does each term of the multi-objective function affect the final solution? We answer this question by removing each term and we evaluate with the total final objective value. Fig. 9 presents the results. We see that when we exclude the individual benefit ($\gamma = 0, \delta = 1$), the final objective value is equal to zero (optimal solution). However, this is not the case when we exclude the group benefit term ($\gamma = 1, \delta = 0$). This indicates that the individual benefit term is the hardest to optimize.
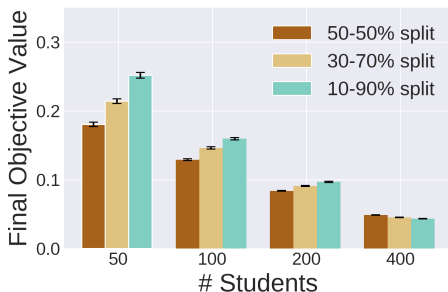

Fig. 5: Final value of the objective function for the different relative sizes of the protected groups, on D3.

(a) D1, $\delta = 1$ (d) D1, $\delta = 0$

(b) D2, $\delta = 1$ (e) D2, $\delta = 0$

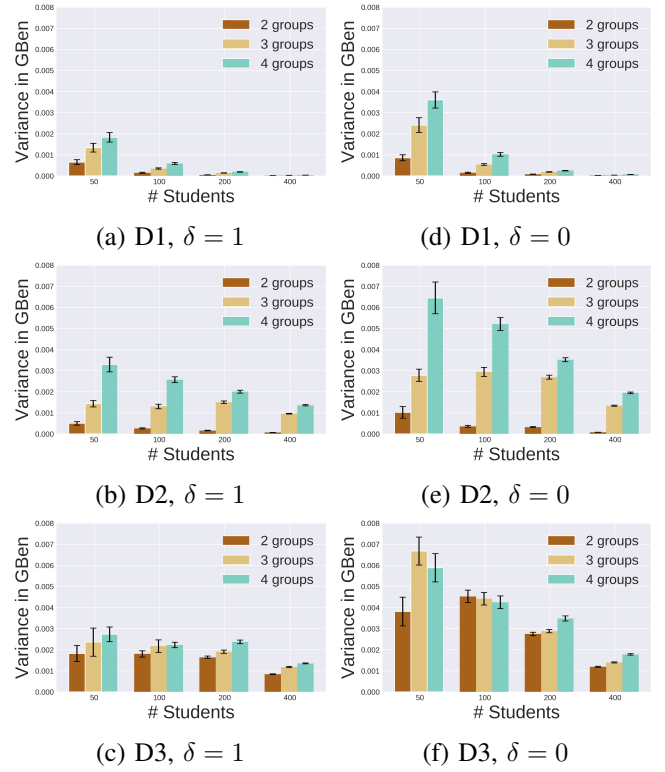(c) D3, $\delta = 1$ (f) D3, $\delta = 0$

Fig. 6: Value of the variance in group benefit for the different numbers of protected groups, on D1, D2 and D3.
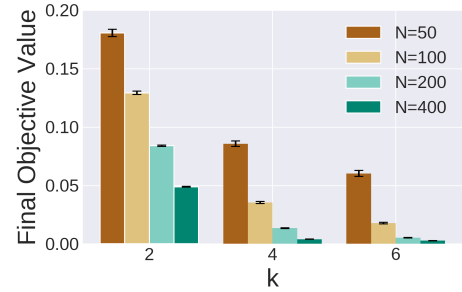

Fig. 7: Final value of the objective function w.r.t. increasing number of dimensions for the skill vectors on D3.

6) Trade-off between Individual Benefit and Group Fairness: Next, we study the relationship between the individual benefit and the group fairness terms, as we increase the relative importance of each one. In particular, while keeping the importance of the group term fixed ($\delta = 1$), we explore how it is affected as we increase γ, i.e., as we increase the weight of the individual benefit. Looking at it the other way, this experiment also shows how the group fairness impacts the individual benefit. Fig. 10 presents the results on the most difficult dataset, D3.

Note that we aim to maximize the individual benefit and minimize the variance in group benefit. We can see that when we aim to optimize the individual benefit, we end up hurting the group fairness. In particular, for $\gamma > 1$ (Fig. 10a), the variance in group benefit deteriorates and at the same time
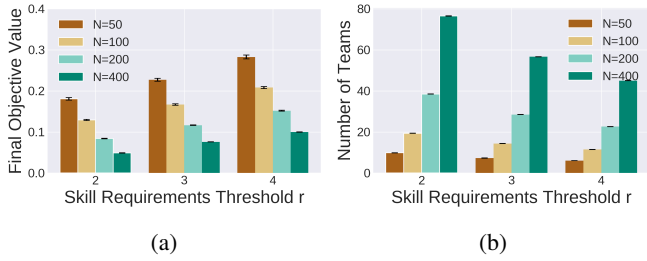
Fig. 8: Results for different values of the skill requirement threshold **r**, for the different number of students in D3. (a) Final value of the objective function with respect to **r**. (b) Number of teams with respect to **r**.

TABLE IV: Effect of the individual benefit and the group fairness terms to the group benefit for two protected groups on D3.

	Umeans			FERN		
	$GBen(q_1)$	$GBen(q_2)$	$\mathcal{Z}$	$GBen(q_1)$	$GBen(q_2)$	$\mathcal{Z}$
$\gamma = 0, \delta = 1$	26.64	86.23	887.80	77.01	77.06	0.00
$\gamma = 1, \delta = 0$	26.64	86.23	887.80	98.69	91.95	11.35
$\gamma = 1, \delta = 1$	26.64	86.23	887.80	98.16	92.50	8.01

The Umeans results correspond to a group fairness unaware solution. $GBen(q_1)$ and $GBen(q_2)$ is the group benefit of protected group q_1 and q_2, respectively. $\mathcal{Z}$ is the variance in $GBen$. The values for $GBen$ represent percentages (%).
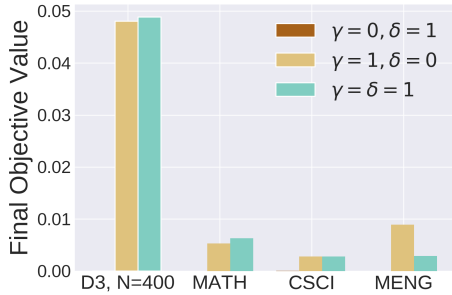

Fig. 9: Final value of the objective function for the different datasets, when we exclude the individual benefit term ($\gamma = 0$), when we exclude the group benefit term ($\delta = 0$) and when both terms have equal contribution ($\gamma = \delta = 1$).

the average individual benefit does not improve significantly. However, this is not the case when we increase the importance of the group term (Fig. 10b). We observe that as we increase δ, the variance in group benefit improves and the average individual benefit is almost constant. This means that we should use higher value for δ than that of γ for the most challenging datasets D2 and D3 in order to further decrease the variance in group benefit. For the remaining datasets, a value of $\delta > 1$ does not further improve the group fairness.

7) Group Benefit of the Final Solution: In this subsection, we support our choice for the definition of group benefit (Eq. 2) and the way we measure group fairness (Eq. 5) by showing how the group benefit changes as we change the relative importance of $\mathcal{Y}$ and $\mathcal{Z}$ in Eq. 6. We compare the solution given by FERN against the one given by a group fairness unaware method, i.e., the solution by Umeans, on D3 and $N = 400$. Table IV presents the results.

According to the Umeans results, we can see that q_1 is the least benefited protected group, as its members benefit from only 26.64% of their teammates, compared to 86.23% of their teammates for members of q_2. In the first case where we ignore the individual benefit ($\gamma = 0$), FERN increases the group benefit of the least benefited protected group q_1, and decreases the group benefit of q_2 in order to perfectly balance the group benefit and impose group fairness. In the other two cases ($\gamma = 1$), the group benefit of both q_1 and q_2 increases, and $GBen(q_1)$ in particular, increases from 26.64% to about 98%. Now, the difference is that when we ignore

the group fairness ($\delta = 0$), the gap between $GBen(q_1)$ and $GBen(q_2)$ is larger than the gap when $\delta = 1$. The findings are similar for the other datasets as well. The second term of the objective maximizes both the individual benefit and the group benefit of each protected group. Then, the third term additionally balances the difference in group benefit.

VII. CONCLUSION

In this work we study the fair team formation problem where we maximize the benefit coming from peer learning. Given a set of students and a target task with specific skill requirements, we aim at assigning the students to teams of flexible size such that: 1) the collective ability of each team is adequate to successfully complete the target task, 2) the number of students that benefit from the team work is maximized and, 3) the different protected groups are treated fairly with respect to benefit. To the best of our knowledge, this is the first work that introduces fairness as *equal opportunity* in collaborative learning for both the individual and protected group level. We formulate the problem as a multi-objective function and we propose FERN; a hill-climbing heuristic. Experimental results in both synthetic and real-world data show the effectiveness of the proposed method.

REFERENCES

[1] A. Ashman and R. Gillies, *Cooperative learning: The social and intellectual outcomes of learning in groups*. Routledge, 2003.
[2] R. E. Slavin, "Cooperative learning and student achievement," *Educational Leadership*, vol. 46, no. 2, pp. 31–33, 1988.
[3] R. Hertz-Lazarowitz, S. Kagan, S. Sharan, R. Slavin, and C. Webb, *Learning to cooperate, cooperating to learn*. Springer Science & Business Media, 2013.
[4] S. Freeman, S. L. Eddy, M. McDonough, M. K. Smith, N. Okoroafor, H. Jordt, and M. P. Wenderoth, "Active learning increases student performance in science, engineering, and mathematics," *Proceedings of the National Academy of Sciences*, vol. 111, no. 23, pp. 8410–8415, 2014.
[5] J. A. Bargh and Y. Schul, "On the cognitive benefits of teaching." *Journal of Educational Psychology*, vol. 72, no. 5, p. 593, 1980.
[6] R. Hertz-Lazarowitz and N. Miller, *Interaction in cooperative groups: The theoretical anatomy of group learning*. Cambridge University Press, 1995.
[7] L. S. Vygotsky, *Mind in society: The development of higher psychological processes*. Harvard university press, 1980.
[8] E. Andrejczuk, F. Bistaffa, C. Blum, J. A. Rodríguez-Aguilar, and C. Sierra, "Synergistic team composition: A computational approach to foster diversity in teams," *Knowledge-Based Systems*, vol. 182, p. 104799, 2019.
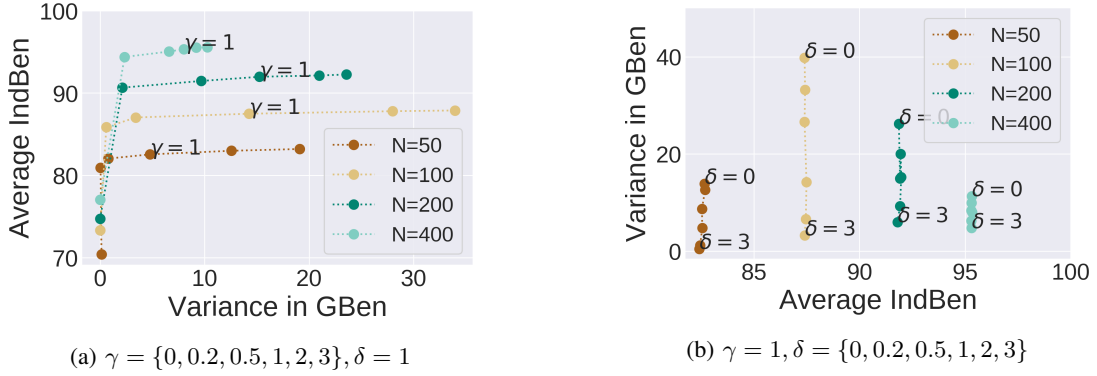
(a) $\gamma = \{0, 0.2, 0.5, 1, 2, 3\}, \delta = 1$ (b) $\gamma = 1, \delta = \{0, 0.2, 0.5, 1, 2, 3\}$

Fig. 10: Trade-off between the individual and the group fairness on D3, as we increase γ (a) and as we increase δ (b). The values for *Average IndBen* and *GBen* correspond to percentages (%).

[9] S. Bandyopadhyay, S. Nandanwar, R. Deshmukh, and M. Murty, "Div-group: a diversified approach to divide collection of patterns into uniform groups," in *Int. Conference on Pattern Recognition (ICPR)*. IEEE, 2018, pp. 964–969.

[10] Y. Liu, Q. Liu, R. Wu, E. Chen, Y. Su, Z. Chen, and G. Hu, "Collaborative learning team formation: a cognitive modeling perspective," in *Int. Conf. on Database Sys. for Advanced Applications*. Springer, 2016, pp. 383–400.

[11] F. Chierichetti, R. Kumar, S. Lattanzi, and S. Vassilvitskii, "Fair clustering through fairlets," in *Adv. in Neural Inf. Processing Sys.*, 2017, pp. 5029–5037.

[12] S. Bera, D. Chakrabarty, N. Flores, and M. Negahbani, "Fair algorithms for clustering," in *Adv. in Neural Inf. Processing Sys.*, 2019, pp. 4955–4966.

[13] I. M. Ziko, E. Granger, J. Yuan, and I. B. Ayed, "Clustering with fairness constraints: A flexible and scalable approach," *arXiv preprint arXiv:1906.08207*, 2019.

[14] L. Shields, A. Newman, and D. Satz, "Equality of educational opportunity," in *The Stanford Encyclopedia of Philosophy*, summer 2017 ed., E. N. Zalta, Ed. Metaphysics Research Lab, Stanford University, 2017.

[15] R. Agrawal, B. Golshan, and E. Terzi, "Grouping students in educational settings," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2014, pp. 1017–1026.

[16] R. Agrawal, S. Nandanwar, and M. Murty, "Grouping students for maximizing learning from peers." *Int. Educational Data Mining Society*, 2017.

[17] S. F. Assmann, D. S. Johnson, D. J. Kleitman, and J.-T. Leung, "On a dual version of the one-dimensional bin packing problem," *Journal of algorithms*, vol. 5, no. 4, pp. 502–525, 1984.

[18] C. M. Fiduccia and R. M. Mattheyses, "A linear-time heuristic for improving network partitions," in *Design Automation Conference*. IEEE, 1982, pp. 175–181.

[19] S. Bahargam, T. Lappas, and E. Terzi, "The guided team-partitioning problem: Definition, complexity, and algorithm," *arXiv preprint arXiv:1905.03037*, 2019.

[20] T. Lappas, K. Liu, and E. Terzi, "Finding a team of experts in social networks," in *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009, pp. 467–476.

[21] E. Scarlat and I. Maries, "A genetic algorithm for community formation based on collective intelligence capacity," in *KES International Symposium on Agent and Multi-Agent Systems: Technologies and Applications*. Springer, 2011, pp. 271–279.

[22] M. Niveditha, G. Swetha, U. Poornima, and R. Senthilkumar, "A genetic approach for tri-objective optimization in team formation," in *2016 Eighth International Conference on Advanced Computing (ICoAC)*. IEEE, 2017, pp. 123–130.

[23] L. Machado and K. Stefanidis, "Fair team recommendations for multi-disciplinary projects," in *2019 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. IEEE, 2019, pp. 293–297.

[24] J. Moreno, D. A. Ovalle, and R. M. Vicari, "A genetic algorithm approach for group formation in collaborative learning considering multiple student characteristics," *Computers & Education*, vol. 58, no. 1, pp. 560–569, 2012.

[25] S. Mishra, A. Yadav, and A. S. Sairam, "Worker selection in crowd-sourced platforms using non-dominated sorting," in *IEEE Region 10 Conference (TENCON)*. IEEE, 2019, pp. 41–45.

[26] A. Yadav, A. S. Sairam, and A. Kumar, "Concurrent team formation for multiple tasks in crowdsourcing platform," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–7.

[27] N. Arthurs, B. Stenhaug, S. Karayev, and C. Piech, "Grades are not normal: Improving exam score models using the logit-normal distribution." *International Educational Data Mining Society*, 2019.

[28] A. Tugend, "Who benefits from the expansion of ap classes," *The New York Times Magazine*, vol. 7, 2017.

[29] G. W. Whiting and D. Y. Ford, "Multicultural issues: Black students and advanced placement classes: Summary, concerns, and recommendations," *Gifted Child Today*, vol. 32, no. 1, pp. 23–26, 2009. [Online]. Available: https://doi.org/10.4219/gct-2009-840

[30] D. Shapiro, A. Dundar, F. Huie, P. K. Wakhungu, X. Yuan, A. Nathan, and Y. Hwang, "Tracking transfer," *NSC Research Center*, 2019.

[31] G. Crisp and A.-M. Nuñez, "Understanding the racial transfer gap: Modeling underrepresented minority and nonminority students' pathways from two-to four-year institutions," *Review of Higher Education*, vol. 37, no. 3, pp. 291–320, 2014.