TeamUp: Form Best Project Teams

Dawei Yin, Sha Hu, and Yuting Zhang(✉)

Boston University, Boston, MA 02215, USA
{davidyin,baymax,danazh}@bu.edu

Abstract. Team formation is a critical task in many contexts such as business, sports, healthcare, research, education, and more. In the academic settings, students working in teams on group projects is proven to be a very effective learning methodology. Our software engineering course features a semester-long team project as the key component of this course. However, team assignments are complex and nontrivial tasks, necessitating a careful assessment of many different factors to ensure optimal performance of the team as well as individual participants' satisfaction. Usually a predefined set of questions are used to understand participant's capabilities and preferences, and teams are formed either manually or automatically based on the results of those questions. In this paper, we propose a generalized question definition by associating each question with three factors: multiple choice/multiple answer, similarity or diversity, and option valuation, in order to consider various types of factors, provide flexibility and capture each individual's characteristics. We then propose two team performance score functions that differentiate similarity questions from diversity questions and capture each of their team formation objectives. A heuristic team formation algorithm, TeamUp, is proposed, attempting to maximize the team performance as well as participants' preferences. Through the initial evaluation we show our proposed algorithm can perform well for different team size and type of questions.

Keywords: team formation · team project · project-based learning

1 Introduction

Team formation is a critical task in many contexts such as business, sports, healthcare, research, education, and more. Whether in academic settings or professional environments, the effective team composition can significantly enhance the success and outcomes of the tasks at hand. However, team assignments can be complex and nontrivial tasks, necessitating a careful assessment of many different factors to ensure optimal performance of the team as well as individual participants' satisfaction. Usually a predefined set of questions are used

T. Zlateva and G. Tuparov (Eds.): CSECS 2023, LNICST 514, pp. 372–384, 2023.
https://doi.org/10.1007/978-3-031-44668-9_29

to understand participant's capabilities and preferences, and teams are formed either manually or automatically based on the results of those questions. While the question content is context dependent, we can categorize possible question types to capture different situations. In this paper, we propose the consideration of three factors and define 6 different questions types. We also propose a team formation algorithm that can take into consideration all those factors and this diverse set of question types.

Students working in teams on group projects is proven to be a very effective learning methodology in the academic setting. Particularly, in the software engineering field, almost all software is developed by teams of developers collaboratively. In order to help students experience this collaboration in real world settings, our software engineering course features a semester-long group project as the key component of this course. As our students are a mix of full-time and part-time graduate students with very diverse backgrounds, schedules, and even time zones (for the online course), and most students don't know each other before taking the class, team assignments are very challenging for the instructors. Usually the instructor and the TAs need to spend quite an amount of time on this task and there is no guarantee that the teams are formed well. The proposed algorithm can help automate this complex process and enhance possible team performance in this course and many other courses that have similar team project components.

In order to facilitate the creation of well-balanced and high-performing project teams, we need to first have a thorough understanding of the diverse preferences, skills, and characteristics of each participant. The data is usually collected through a set of predefined survey questions that participants need to fill out. Multi-choice questions are commonly used and the team formation criteria are usually to try to group individuals with the same choice for those questions. For example, the question "What technology stack do you want to use in your project?" can be used in the Software Engineering course, however, it doesn't provide flexibility for students who can use multiple technology stacks and it also doesn't capture the individual skills and capability of students in that technology stack. Multi-answer questions may be used here to allow students to choose multiple technology stacks, however, it still doesn't reflect students' preferences and capability in each of the technology stacks they choose. Furthermore, while we want to team up students that want to use the same technology stack to improve the team performance, we also want to diversify other factors in a team such as gender and work roles preferences. In our Software Engineering course, each team has different roles such as team leader, requirement leader, design and implementation leader, QA leader, and etc. We would better team up students with preferences and suitability in different roles rather than the same roles. To consider all the above issues, we associate each question with three factors: the number of choices each participant can choose, the similarity or diversity of participants' choices that should be in a team, as well as the capability or preferences value of each choice. Through these three factors we can represent a wide range of questions with a uniform definition of question, and our algorithms aim to enhance both individual preferences and team performance by considering all those three factors, in order to generate optimized team formation to promote collaboration, complimentary, and ultimately, successful project outcomes.

The contributions of this paper is summarized as follows:

1. Propose a generalized question definition by associating each question with three factors: multiple choice/multiple answer, similarity or diversity, and option valuation. In particular, by allowing participants to assign values to options, it can capture the relative preferences or expertise of participants associated with each option. This enables more precise and tailored team compositions, allowing that participants with specific skills or preferences are appropriately assigned to project groups.
2. Propose two team performance score functions that differentiate similarity questions from diversity questions and capture each of their team formation objectives.
3. Propose a heuristic team formation algorithm, TeamUp, that attempts to maximize the team performance as well as participants' preferences. The initial evaluation shows its efficiency and effectiveness.

The remaining sections are organized as follows. Section 2 details how our research relates and differs from some other related work. Section 3 provides a detailed description of the problem, in particular defines the supported questions that consider three refactors mentioned above, and the algorithm TeamUp is described in Sect. 4. Then we used a generated synthetic data to evaluate the algorithms and show its efficiency and effectiveness in Sect. 5. Finally we conclude our paper and list some future work in Sect. 6.

2 Related Work

The TeamMaker system, developed by Layton, et al. [1] is the most similar work to ours. It forms teams based on instructor-defined criteria using a max-min heuristic. TeamMaker supports various types of questions, including multiple-choice and multiple-answer questions. While they have defined different score functions for multiple-choice and multiple-answer questions, both functions primarily rely on the number of options chosen by team members. A higher value indicates greater diversity. They differentiate similarity and diversity by assigning negative weights to similarity questions and positive weights to diversity questions. However, we argue that this score definition fails to capture the importance of measuring the degree of similarity based on the number of team members choosing the same option. For instance, in a team of 10 members, if 9 students choose one option and 1 student chooses another option, the score would be the same as a scenario where 5 students choose one option and 5 students choose another option, since both cases have two options chosen. In contrast, our algorithm defines distinct score functions for similarity and diversity questions, effectively capturing the dynamics of each question type. Additionally, we empower participants to assign values to options, allowing us to incorporate their capabilities and preferences in team formation, thereby enhancing team performance and meeting the participants' preferences.

Meulbroek, Ferguson, Ohland, and Berry [2] proposed an extension to'TeamMaker" by using the Gale-Shapley algorithm to form initial teams. However, their approach requires preference list from both students and instructors, which does not align with the general case defined in our algorithm. Thus, for team initialization, we opted simple random assignment for initial team formation, similar to the example in [9]. This choice was made to diminish the complexity of the subsequent team member swaps.

Several work only underscored the importance of heterogeneity in group dynamics and leveraged genetic algorithms to optimize team heterogeneity [3,6], [7,8]. Additionally, [4] focuses on role assignment within the group, and [5] studies optimization of team assignments in emergency departments. Our goal, instead, is to form the groups based on various factors , while taking consideration of role assignments through diversity questions.

3 Problem Definition

Suppose we have the participant set as $P = \{p_i | i = 0, ..., N\}$, and we need to form M teams $T = \{T_i | i = 1, ..., M\}$. We represent the participant p_i assigned to the team t_j as $team(p_i) = T_j$. The size of the team T_j is $|T_j|$, which is the number of participants in that team.

The team formation criteria is defined through a set of K survey questions $Q = \{q_i | i = 1, ..., K\}$, with each question q_i associated with a weight w_i. The default value of w_i is 1. We differentiate two types of questions in the formation criteria, namely similarity questions and diversity questions. Similarity questions aim to identify commonalities or preferences for the same or similar options among team members, while diversity questions focus on differences or preferences for diverse perspectives within each group. Therefore, it is intended to group participants with same/similar responses to the similarity questions and group participants with different responses to the diversity questions.

For each question q_j, we define given options as a set of options $O^j = \{o^j_i | i = 1, ...|O^j|\}$, where $|O^j|$ is the total number of given options for question q_j. In order to provide participants more flexibility and capture participants' preferences or capabilities on the given options, we define the response of each participant p_i to Question q_j as

$$R^i_j = (index(o^j_m), value(o^j_m)) | m = 1...maxO^j \tag{1}$$

where $index(o^j_m)$ is the index of the chosen option and $value(o^j_m)$ is the value associated with that option. By default, the value is between 1 to 5, with 5 representing the highest capability or preferences of the participant to that option. If the participant doesn't specify the value, a default value of 5 will be used.

$maxO^j$ is the maximum number of options that a participant can choose for question q_j. It can also be customized. A multiple choice question restricts that number to be 1 as each participant can only choose one option, while a multi-answer question enables the participants to choose more than 1 option, possibly all options.

In summary, our problem definition consider the following factors that can capture the intricacies of team dynamics and affect the team formation:

1. A weight for each question to specify the importance of the question in the team formation
2. A type (S or D) for each question to differentiate the team formation criteria in similarity questions from diversity questions
3. The maximum number of options a participant can choose to provide participants more flexibility in their response
4. The capability or preference value of the chosen option in participant's response to reflect each individual characteristic.

The goal is to form teams that can maximize the team performance as well as individual preferences.

4 Algorithm

4.1 Objective Function

In order to maximize the team performance and participants' preferences, we define the team performance score for each team T_z as the weighted summation of the team score in each question.

$$Score(T_z) = \sum_{i=0}^{K} Score_i^z * w_i \tag{2}$$

where $score_i^z$ is the score of team T_z for question q_i, and w_i is the weight for question q_i and K is the total number of questions.

As the team criteria is different in similarity and diversity questions, we define two different score functions for them:

Similarity Question: The performance score of a team T_z for a similarity question q_m is defined as

$$Score\_similar_z^m = \frac{1}{|T_z|} \max_{i=1}^{|O_m|} \left(\sum_{j=0}^{|T_z|} val_j^{i,m} \right) \tag{3}$$

where $|T_z|$ is the size of the team T_z, that is the number of participants in the team. $|O_m|$ is the number of given options in question q_m. $val_j^{i,m}$ is the jth participant's specified value for question q_m's the ith option o_i^m. If a participant doesn't choose that option, that value is 0. If the question doesn't need the participant to specify the value, a default value of 5 will be used instead.

For similarity questions, we would like to team up participants who choose the same options and maximize that similarity. In addition, we consider each individual participant's capability or preference value and attempt to maximize that value for each option as well. Therefore, we define each option's similarity performance score as the summation of each participant's value for that option. This value increases if more participants choose this option, or participants set higher option value to this option, indicating higher capability or preferences for this option. To maximize that similarity, we choose the highest option score as the question score, as the option with the highest score will be mostly the favored option for the team. If all team members choose the same option and set the option value as 5, then the team score for the question will be the highest value 5.

This can be better understood with the following example. Suppose in the Software Engineering course survey, we have the following similarity question: "Which programming languages would you like to use in your project and rate your proficiency in each programming language that you choose, with 5 being the highest proficiency." Assume that we have the response of each team member in Table 1:

Table 1. Similarity Question Example

Team Member	Java	Python	C++	R
Participant 1	5			
Participant 2		4		
Participant 3	3			
Participant 4			3	
Participant 5		3		
Summarize option value	8	7	3	0

After we summarize all participant's option value for each option, we could find the highest sum is for option "Java", which may be the favorite programming language to be used by the team. The team performance score is therefore set as

$$S = \frac{1}{5} * val_{Java} = \frac{1}{5} * 8 = 1.6 \tag{4}$$

Diversity Question: On the contrary, the performance score of a team T_z for a diversity question q_m is defined as

$$S\_diversity_z^m = \frac{1}{|O_m|}(\sum_{i=1}^{|O_m|} \max_{j=1}^{|T_z|}(val_j^{i,m})) \tag{5}$$

All annotations are the same as in the previous similarity case, where $|T_z|$ is the size of the team T_z, that is the number of participants in the team. $|O_m|$ is the number of given options in question q_m. $val_j^{i,m}$ is the jth participant's specified value for question q_m's the ith option o_i^m. If a participant doesn't choose that option, that value is 0. If the question doesn't need the participant to specify the value, a default value of 5 will be used instead.

For diversity questions, we would like to team up participants who choose different options and maximize that diversity. In addition, we consider each individual participant's capability or preference value and attempt to maximize the overall value of different options. Therefore, the team members who choose the same options do not add the additional value to diversity. Each option's value is only determined by the highest option value of all team members who choose that option. To maximize the diversity, we define the performance score as the summation of all option's value. This score increases if more participants choose different options or various participants set higher value for different options. In the most diverse case, where all options are chosen by the team with highest value assigned to each option, the value will also be 5, the highest possible value.

This can be better understood with the following example. Suppose in the Software Engineering course survey, we have the following diversity question: "Which leader role would you like to take in your project and rate your preferences in each role that you choose, with 5 being the highest preference." Assume that we have the response of each team member in Table 2:

Table 2. Diversity Question Example

Team member	Team leader	Design leader	Security leader	Testing leader
Participant 1	4			
Participant 2		5		
Participant 3	3			
Participant 4			3	
Participant 5		3		
highest option value	4	5	3	0

After we get the highest score of all participants' value for each option, we could calculate the performance score for the question as the average score of all option values

$$S = \frac{1}{4} * sum(val) = \frac{1}{4} * (4 + 5 + 3) = \frac{1}{4} * 12 = 3 \tag{6}$$

4.2 Find Best Team Formation

In order to maximize overall team performance score and minimize the worst performance team defined above, we adopt a heuristic swapping algorithm similar to the one defined in [1] by swapping team members from two different teams if this swap can improve the performance.

First, we perform an initial random team assignments. Then we initiate an iterative process encompassing pairs of two teams to perform the swap. The swap algorithm in [1] consider all possible pairs of team, which has a n^2 complexity, where n is the total number of participants. This cause the algorithm extremely slow. To reduce the complexity and enhance the efficiency, we will consider the swap of only adjacent pairs of teams. Starting with the first adjacent pair, we probe potential member exchanges between the two teams under consideration. Upon identifying an exchange that can result in the enhancement of the minimum team score within that adjacent pair, we implement this exchange. Consequently, we then shift our focus to the subsequent pair of adjacent teams. This systematic approach of exploring and actualizing advantageous member exchanges perpetuates sequentially through all pairs of adjacent teams as described in the Algorithm 1.

In every "Pass", an iterative process takes place which can include a number of iterations until a round is completed without any team member swaps. Each Pass essentially constitutes a thorough traversal of the initial team, with the aim of achieving an optimal arrangement of team members. Given that the initial team assignments are random, we can enhance the significance of our results by setting up additional initial teams. This gives us multiple possible Pass results, and we select the pass that gives the highest minimum team score to be our final team assignment outcome.

Algorithm 1. One-time Iteration Algorithm

Input: Initial Team assignments and corresponding team scores
Output: New team assignments and team scores after swapping

```
Initialize pointer p ← 0
while p is not at the last team index do
    swap_status ← False
    for each member i in team[p] do
        for each member j in team[p + 1] do
            if swap(i, j) leads to improvement in the minimum team score then
                Perform swap(i, j)
                swap_status ← True
                break
            end if
        end for
        if swap_status then
            break
        end if
    end for
    p ← p + 1
end while
return TeamScores
```

Table 3. Results Of the Four Passes

Pass Min. Score	Team Scores						
	Te. 1	Te. 2	Te. 3	Te. 4	Te. 5	Te. 6	Te. 7
Pass 1 1.2	3.9	7.6	1.2	6.5	8.3	5.7	9.1
Pass 2 1.5	4.6	8.5	2.3	3.8	7.9	9.2	1.5
Pass 3 3.7	9.2	6.3	3.7	7.1	5.9	8.4	4.8
Pass 4 2.9	5.1	9.8	2.9	4.2	6.7	8.1	3.6

For example, divide the class into 7 teams and complete 4 rounds. Suppose the score in each pass is shown in Table 3. According to results of the four passes, the maximum minimum score achieved is 3.7. Therefore, based on the grouping result, pass 3 is our final grouping decision. This can be seen more clearly in Table 3.

5 Evaluation

This section aims to evaluate the performance of our heuristic team formation algorithm. As our algorithm differentiates similarity from diversity questions, where the goal is to group participants with the same responses for similarity questions, and group participants with different responses for diversity questions, we define the following performance metrics for each case to evaluate the performance:

For similarity questions, we assume that the option that receives the highest value from all team members will be the favorite option for the team. More members choose that option, higher the similarity. On the other hand, more options are chosen by the team members, higher the diversity. Therefore, we considered the following **performance metrics**:

1. performance score: this is calculated based on the score function defined in the previous section.
2. similarity degree: average percentage of members choosing the favored option in a team.
3. diversity degree: average percentage of choices chosen by members.

Experiment Setup: The simulation data we generate for this study serves to recreate a situation where we have 200 individuals responding to a series of questions. These responses are then utilized to assemble groups of varying sizes. The group sizes considered in our study include groups of 2, 5, and 10 participants, representing small, medium, and large groups, respectively. This experiment is repeated 10 times to increase the robustness of our findings. Each participant is tasked with responding to a total of 8 questions, each question with 5 given options. To be applicable to a broad range of real-world scenarios, we consider 4 types of questions, with 2 questions in each category. They are similarity questions with variable option value (SimVal), similarity questions without option value (SimNoVal), diversity questions with variable option value (DivVal) and diversity questions without option value (DivNoVal).

Statistical Test: We compared our TeamUp algorithm with a basic random assignment algorithm and the team-maker algorithm respectively across different team sizes. In our analysis, we employ the t-test as our primary statistical test. The t-test is a popular statistical test that is used to determine if there is a significant difference between the means of two groups. In the context of our evaluation, the t-test is used to assess whether the performance metrics of our algorithm significantly differ from the other two approaches.

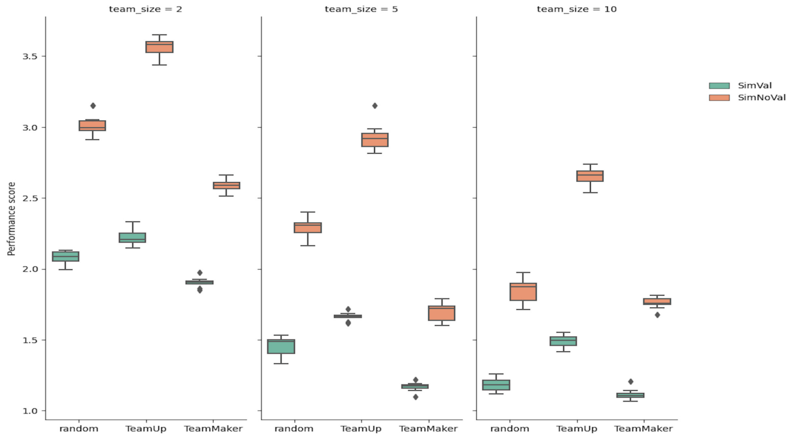
Performance Score. The team performance score are shown in Fig. 1a and 1b. It shows that TeamUp consistently outperforms the random assignments and Team-Maker no matter what the team size is. The difference in mean performance scores was statistically significant, as established by a paired t-test (t(df) = value, $p < .05$), underscoring the superior effectiveness of TeamUp in fostering high-performing teams.

Interestingly, the performance scores yielded by the Team-Maker system were not only lower than those of TeamUp but also underperformed relative to random group assignments. This finding suggests possible limitations in the Team-Maker, particularly in its ability to effectively incorporate participant response option values as grouping criteria. Such an inability may potentially impede its capacity to form optimal groupings that maximize performance scores.

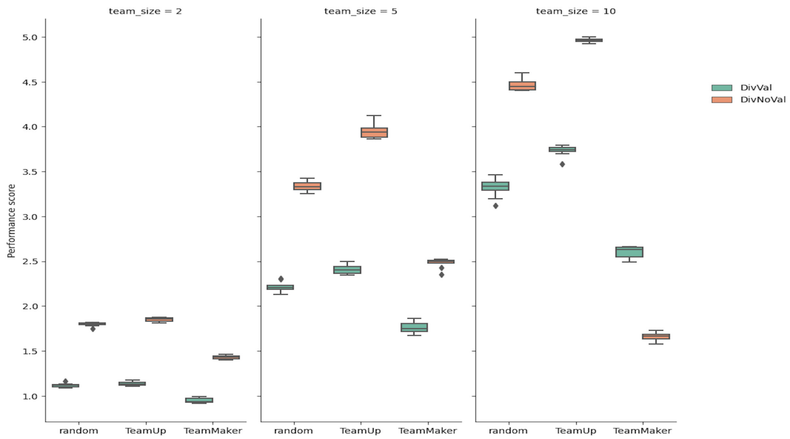
Similarity Degree. When evaluating the average percentage of team members selecting their preferred option, TeamUp consistently demonstrates superior performance compared to other methods. This superior performance is maintained across a variety of team sizes and for both fixed and variable option value questions. This suggests that TeamUp more effectively aligns teams according to individual participant preferences, which could facilitate improved team harmony and effectiveness.

Interestingly, as team size increases, there is a general trend across all methods indicating a decrease in the likelihood of individual team members selecting their preferred options. This is likely due to the natural dilution effect of larger teams, where individual preferences have less influence. However, despite this trend, TeamUp continues to outperform other methods, further underscoring its utility in team formation, even as team size scales up (Fig. 2).

Diversity Degree. In addition, TeamUp significantly outperforms both the random allocation method and the team maker tool when considering the average percentage of distinct options chosen by team members. This suggests that, on average, TeamUp is more effective at assembling teams wherein members favor a variety of options, in both fixed and variable choice value situations.



(a) Similarity question performance score comparison



(b) Diversity question performance score Comparison

Fig. 1. Comparison of performance scores

Notably, as the team size increases under fixed option conditions, all grouping methods tend to facilitate a higher degree of diversity. This could be attributed to the increased possibility of each option being chosen as team size grows, thus maximizing the representation of different preferences within a team. This diversity is a potential asset, promoting a broader range of perspectives and skills within each team (Fig. 3).

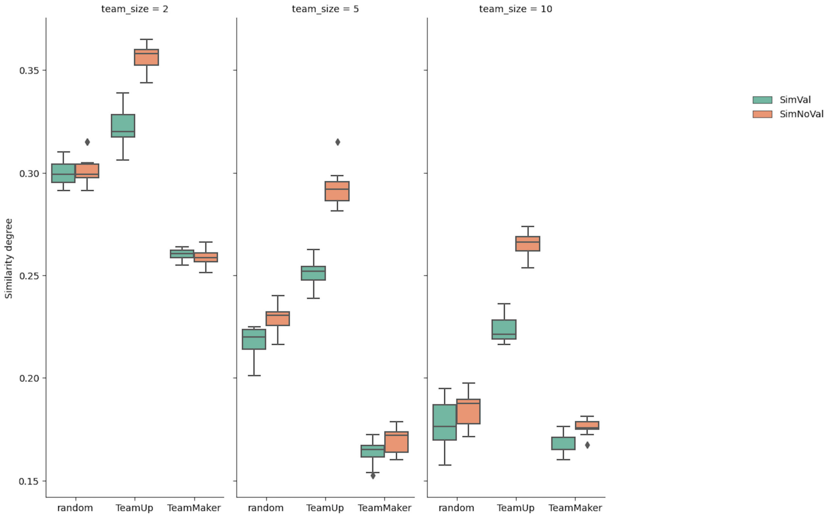


Fig. 2. Similarity Degree Comparison

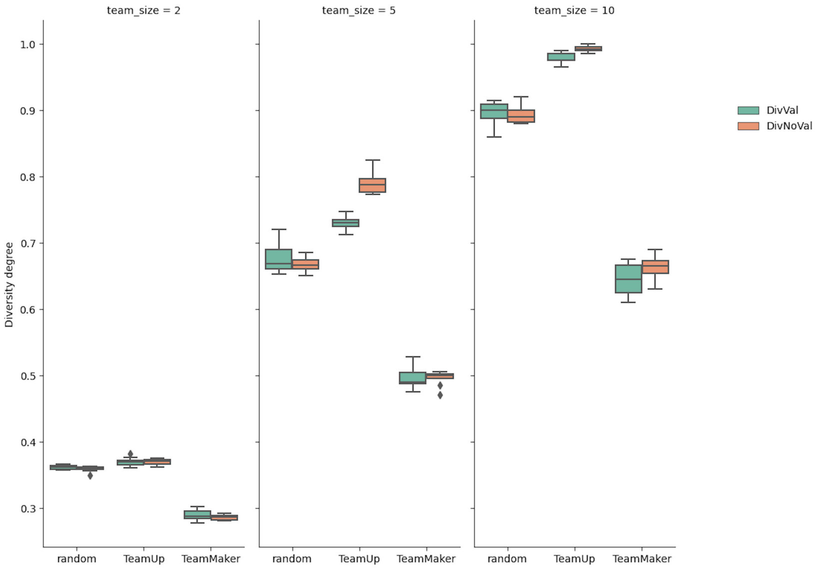


Fig. 3. Diversity Degree Comparison

6 Conclusion and Future Work

In this paper, we present TeamUp, a heuristic team formation algorithm based on survey results related to various types of questions. These questions encompass factors such as multiple-choice or multiple-answer options, similarity or diversity aspects, and the presence or absence of option valuation. Our algorithm aims to maximize team performance while accommodating the individual preferences of participants. Through initial evaluations, we demonstrate the efficacy of our proposed algorithm across different team sizes and question types. Possible improvements include implementing a more intelligent team initialization process, refining the swap method for enhanced effectiveness, and considering team size variations. Furthermore, we intend to conduct a comprehensive evaluation to further explore the algorithm's enhancements and applicability.

References

1. Layton, R.A., Loughry, M.L., Ohland, M.W., Ricco, G.D.: Design and validation of a web-based system for assigning members to teams using instructor-specified criteria. Adv. Eng. Educ. **2**, 1–28 (2010)
2. Meulbroek, D., Ferguson, D., Ohland, M., Berry, F.: Forming more effective teams using CATME teammaker and the gale-shapley algorithm. In: 2019 IEEE Frontiers in Education Conference (FIE), Covington, KY, USA, pp. 1–5, (2019). https://doi.org/10.1109/FIE43999.2019.9028552
3. Paredes, P., Ortigosa, A., Rodriguez, P.: A Method for Supporting Heterogeneous-Group Formation through Heuristics and Visualization. J. Univers. Comput. Sci. **16**, 2882–2901 (2010)
4. Zhu, H., Zhou, M., Alkins, R.: Group role assignment via a Kuhn-Munkres algorithm-based solution. IEEE Trans. Syst. Man Cybern. - Part A: Syst. Humans **42**(3), 739–750 (2012). https://doi.org/10.1109/TSMCA.2011.2170414
5. Patel, P.B., Vinson, D.R.: Team assignment system: expediting emergency department care. Ann. Emerg. Med. 46(6):499–506. Epub 2005 Sep 1. PMID: 16308063 (2005). https://doi.org/10.1016/j.annemergmed.2005.06.012
6. Assavakamhaenghan, N., et al.: Software team member configurations: a study of team effectiveness in moodle. In: 10th International Workshop on Empirical Software Engineering in Practice (IWESEP), Tokyo, Japan, (2019). https://doi.org/10.1109/IWESEP49350.2019.00012
7. Cavanaugh, R.M., Ellis, M.L., Layton, R.A., Ardis, M.A.: Automating the process of assigning students to cooperative learning teams. In: the Proceedings of the 2004 American Society for Engineering Education Annual Conference and Exposition (2004)
8. Imbrie, P.K., Agarwal, J., Raju, G.: Genetic algorithm optimization of teams for heterogeneity. In: 2020 IEEE Frontiers in Education Conference (FIE), Uppsala, Sweden, 2020, pp. 1–5, (2020). https://doi.org/10.1109/FIE44824.2020.9274243
9. Xiao, L., Huang, Q., Yank, V., Ma, J.: An easily accessible web-based minimization random allocation system for clinical trials. J. Med. Internet Res. **15**(7), e139 (2013). https://doi.org/10.2196/jmir.2392