







An Integer Linear Programming Model for Team Formation in the Classroom with Constraints

Gonzalo Candel¹, Victor Sánchez-Anguix²(✉) , Juan M. Alberola³ ,
Vicente Julián^{3,4} , and Vicent Botti^{3,4} 

¹ Universitat Politècnica de València, Camino de Vera, s/n, 46022 Valencia, Spain
goncanpe@inf.upv.es

² Instituto Tecnológico de Informática, Grupo de Sistemas de Optimización Aplicada, Ciudad Politécnica de la Innovación, Edificio 8g, Universitat Politècnica de València, Camino de Vera s/n, 46022 Valencia, Spain
vicsana1@upv.es

³ Valencian Research Institute for Artificial Intelligence (VRAIN), Universitat Politècnica de València, Camino de Vera, s/n, 46022 Valencia, Spain
{jalberola,vjulian,vbotti}@upv.es

⁴ ValgrAI (Valencian Graduate School and Research Network of Artificial Intelligence), Universitat Politècnica de València, Camino de Vera, s/n, 46022 Valencia, Spain

Abstract. Teamwork is essential in many industries to tackle complex projects. Thus, the development of teamwork skills is crucial in higher education. In the classroom, the formation of teams must be fostered throughout all phases to promote the development of these skills. Several criteria for forming teams in the classroom have been proposed, including Belbin's role taxonomy or Myers-Briggs type indicator. However, finding optimal teams or partitions of members into teams is a highly combinatorial problem that requires optimization techniques. This paper presents an integer linear programming model for team formation in the classroom that includes constraints requested by lecturers and allows for the incorporation of different team evaluation heuristics. We study the performance and the scalability of the model using different solvers, conditions, and problem instance types.

Keywords: Teamwork · integer linear programming · team evaluation heuristics · team formation · optimization

1 Introduction

Increasingly, teamwork prevails in industries to address complex projects that can consume many resources and require extensive planning and management. In these scenarios, large-scale projects that cannot be carried out individually often have to be tackled, requiring the participation of individuals working together. Therefore, the demand for skills related to teamwork such as communication, task allocation, leadership, and conflict resolution, is constant.

In this sense, it is crucial that competencies related to teamwork can be developed in higher education degrees. In fact, the importance of future professionals relies on their technical and soft skills, among which teamwork is included [22]. Teamwork requires skills related to the ability to communicate to express and defend ideas in front of a group of peers, leadership, or efficient time management, among many others [15].

To promote the personal development of team members in a classroom, teams must be structured in a way that allows for a satisfactory experience. In the literature, several approaches for forming teams in the classroom have been proposed, ranging from simple ones such as grouping students according to their average grade, or the time they have available for work, to more complex approaches based on personality and behavior such as the Belbin's role taxonomy [5], the Myers-Briggs type indicator [17] (MBTI) or the Big five inventory [13].

The problem of finding optimal teams or partitions of members into teams is known as the team formation problem. Finding an optimal partition of students into teams is a highly combinatorial problem that is both difficult to be solved manually and using algorithms [14]. In fact, many variants of the team formation problem are NP-hard problems [14]. Thus, the use of optimization techniques is necessary to tackle this problem appropriately. Typically, the team formation problem relies on a function that is capable of estimating team performance prior to carrying out the task at hand. This function can be named as the team evaluation heuristic, as it approximates the performance of the team by relying on different criteria.

In the literature, several authors have proposed the use of heuristic and metaheuristic algorithms to tackle the team formation problem in the classroom. For instance, Yannibelli et al. [29] proposed a genetic algorithm based on crowding to group students based on Belbin's role taxonomy. In another study, Andrejczuk et al. [3] proposed an anytime heuristic to tackle large instances of team formation based on personality traits, congeniality, and competences. The authors in [4, 26] propose a genetic algorithm for dividing students into teams based again on Belbin's role taxonomy. The article [11] proposes a multi-objective genetic algorithm that aims to foster homogeneity across groups and heterogeneity within groups.

While heuristics and metaheuristics are necessary to tackle larger instances, they may not be advisable for conducting experiments to compare the effectiveness of several team evaluation heuristics in the classroom. As mentioned, team evaluation heuristics are functions that estimate team performance based on several criteria. Due to the small number of teams that one may form in the classroom, one should reduce variability in the study to make the most of available data. Heuristics and metaheuristics may introduce noise into the study due to their approximate nature, and exact methods such as mathematical programming may be preferred. Ideally, exact methods should be as scalable as possible to ensure their use in large classrooms.

Algorithms for integer linear programming (ILP) are well-suited for obtaining optimal solutions to complex problems due to their ability to handle discrete decision variables [12]. These types of algorithms have been successfully applied

to various similar optimization problems, such as resource allocation [28], matching of students to supervisors [24], task assignment to agents based on their capabilities [8], project allocation to individuals according to their skills [7, 23], or grouping of students for peer assessment [27]. These algorithms allow us to obtain optimal solutions to problems, making them suitable for situations where one wants to conduct experiments to compare several team evaluation heuristics.

In this work, we propose a linear integer programming model for team formation. The model allows for the incorporation of various constraints (e.g., team size, members who should be placed together or not, etc.) that commonly arise in a classroom setting. In addition to this, the model is generic and it can incorporate several team evaluation heuristics. In this article, we study the performance of the model under two team evaluation heuristics: one based on Belbin's role taxonomy [5] and another based on Myer-Briggs Type Indicator [17]. We focus on these two criteria as they have been widely employed in the literature [2–4, 9, 26]. We present experiments to compare the performance and scalability of implementations of this model under different non-commercial solvers.

The rest of the paper is organized as follows. Section 2 briefly formalizes the team formation problem in the classroom and introduces two team evaluation heuristics employed in this study for creating problem instances. Then, Sect. 3 presents the mathematical model for team formation. Section 4 provides the evaluation of the model by comparing different solvers. Finally, Sect. 5 highlights some concluding remarks and draws some future work lines.

2 Team Formation Problem in the Classroom

The team formation problem in the classroom typically aims to partition a set of students into disjoint teams. Let us briefly formalize the problem.

Given a set of students $S = \{s_1, \dots, s_p\}$, we aim to form teams whose sizes are in the set $\mathcal{L} = \{l_1, \dots, l_r\}$.

We define $N \subset S \times S$ as the set of pairs of students who should not be in the same team for pedagogical issues. Similarly, we define $C \subset S \times S$ as the set of pairs of students who should be in the same team. We define $T = \{t_1, \dots, t_q\}$ as the set of feasible teams. A team t_i will be feasible if and only if $l \in \mathcal{L}$ and it is satisfied that $\forall s_j, s_k \in t_i, (s_j, s_k) \notin N$.

The goal of team formation problems in the classroom is finding a partition of students into teams that is optimal. Typically, the optimality of a team is linked to its performance. However, exactly knowing a team's performance prior to the execution of their tasks is not possible. Therefore, team formation problems typically employ a heuristic to estimate or serve as a proxy function for team performance (i.e., the team evaluation heuristic). One of the common heuristics found in the literature is using team heterogeneity as an approximation to future team performance. Heuristics based on the Belbin taxonomy and the Myer-Briggs indicator have been a common approach due to their foundations on management and psychology theories. In this article, we will employ two team evaluation heuristics based on both studies to analyze the performance of our ILP model.

2.1 Belbin Team Evaluation Heuristic

Belbin's role taxonomy defines one of the most important theories regarding successful team dynamics [5]. In this theory, Belbin identifies eight behavioral patterns that are required for a successful team. He called these behavioral patterns as roles. A team member could play different roles within the team as a result of the emergence of different behaviors at different times. That is, there is no limitation on the number of roles that may be played by an individual. Belbin stated that a good balance in the distribution of roles within a team showed more satisfactory team-level results than teams with overrepresented or lacking roles. To obtain the predominant roles of each member, the Belbin Self-Perception Inventory is used [16]. This inventory calculates a numerical score for each of the roles for each team member.

The heuristic defined for this theory considers that a team member has a predominant role when they obtain a *high* or *very high* score associated with this role, according to the salience level defined by Partington and Harris [18].

If $b_{j,k}$ is the score obtained by student s_j in role k , and β_k is the threshold for which it is considered *high*, then we say that the team acquires a positive score in role k if any of the members surpass or equalize threshold β_k . We formalize this score as $f_k(t_i)$. The total score of the team $f(t_i)$ is the sum of the scores obtained for each of the roles normalized by the maximum to be achieved. Formally, both are defined as follows:

$$f_k(t_i) = \begin{cases} 1 & \text{if } \exists s_j \in t_i, b_{j,k} \geq \beta_k \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

$$f(t_i) = \frac{1}{8} \times \sum_{k=1}^8 f_k(t_i)$$

2.2 MBTI Team Evaluation Heuristic

The Myers-Briggs Type Indicator (MBTI) is an instrument that focuses on identifying an individual's personality in four different dimensions [17]. Each of these dimensions is formed by 2 traits that are opposite to each other, and whose combination defines 16 different personalities. The MBTI inventory allows to obtain a score for each of these four dimensions, determining the personality trait assigned to an specific team member in these dimensions.

The heuristic used is a normalized version of the one presented by Pieterse et al. [21]. In this heuristic, we will assume that k is one of the four dimensions (introversion-extraversion, sensation-intuition, thinking-feeling, judging-perceiving) and k_1 and k_2 are the traits in that dimension (e.g., introversion and extraversion for the introversion-extraversion dimension). In addition, $\gamma_k(s_j, k_l)$ is a function that returns 1 if student s_j shows the trait k_l in dimension k . The heuristic applied to a specific dimension and team will return 0 if all members show the same personality trait; 1 if at least one member presents a different trait compared to the other members; and 2 otherwise. The final score of a team

is represented as the sum of the scores obtained for each dimension normalized by the maximum score to be achieved:

$$f_k(t_i) = \begin{cases} 0 & \text{if } \exists k_l, \sum_{s_j \in t_i} \gamma_k(s_j, k_l) = |t_i| \\ 1 & \text{if } \exists k_l, \sum_{s_j \in t_i} \gamma_k(s_j, k_l) = 1 \\ 2 & \text{otherwise} \end{cases} \quad (2)$$

$$f(t_i) = \frac{1}{8} \times \sum_{k=1}^4 f_k(t_i)$$

3 Integer Linear Programming Model

In this section, we describe the formulation of the ILP model that seeks to maximize the sum of scores of teams formed in the classroom. Apart from constraints that ensure that students are partitioned into disjoint teams, the model is based on some common constraints that lecturers and teachers want to employ when forming teams in the classroom:

- Constraints to put two or several students together in the same team.
- Constraints to avoid two or several students being placed into the same team.
- Constraints to control the size of the teams.
- Constraints to control the number of teams formed for each team size.

The model assumes that all feasible teams $t_i \in T$ have been generated, as it creates a binary decision variable for each of the teams that can be formed in the classroom. In addition to this, and due to the fact that decision variables represent the choice of specific teams, the model can incorporate any team evaluation heuristic into the objective function. Let $\mathcal{L} = \{l_1, \dots, l_r\}$ define the set of allowed team sizes for the team formation. The ILP model is defined as follows:

$$\max \sum_{t_i \in T} f(t_i) \times \delta_i \quad (3a)$$

s.t.

$$\sum_{t_i \in T, s_j \in t_i} \delta_i = 1 \quad \forall s_j \in S \quad (3b)$$

$$\sum_{t_i \in T, s_j, s_k \in t_i} \delta_i = 1 \quad \forall (s_j, s_k) \in C \quad (3c)$$

$$m_l \leq \sum_{t_i \in T, |t_i|=l} \delta_i \leq M_l \quad \forall l \in \mathcal{L} \quad (3d)$$

where, on the one hand, δ_i is a binary decision variable that indicates whether feasible team t_i is chosen or not for the solution. Specifically, its value will be 1 when the team participates in the team structure and 0 otherwise. On the other hand, constraint 3b ensures that a student is assigned to precisely one team, ensuring the partition into disjoint teams. Next, constraint 3c ensures

that students that should go in the same team are assigned to the same team. Finally, constraint 3d limits allowed team sizes, and the number of teams for each allowed team size with lower and upper bounds m_l and M_l respectively. Finally, as shown in 3a, $f(\cdot)$ is a function that numerically estimates the performance or quality of a team. For instance, it could represent any of the team evaluation heuristics presented in Sect. 2. With respect to the constraint that precludes students from being placed on the same team, this can be easily implemented without any formal constraint by not generating decision variables that represent teams with incompatible members.

4 Experiments

In this section, we show the different experiments that we conducted to study the scalability of the proposed model under different non-commercial solvers and conditions. The main goal of the experiments is to assess the scalability of the ILP model under different conditions. First, we introduce the solvers that will be employed in the experiments, then we describe the problem instances used in the evaluation, and, finally, we describe the results obtained in the experiments.

4.1 Solvers

The solvers that will be employed in the study are non-commercial solvers. The reason behind this is that team formation problems in the classroom are common in educational settings and, therefore, many educational institutions may not have access to commercial solvers. The solvers employed in the experiments are:

- SCIP [6] is one of the fastest academic solvers for mixed integer programming and mixed integer nonlinear programming.
- COIN-OR Branch and Cut (CBC) [10] is an open-source solver that allows solving linear programming and mixed integer programming problems, and is a variant of the branch and bound technique, which among its operations includes adding cutting planes to search for the solution more efficiently.
- CP-SAT [20] is a solver designed to solve integer programming problems that consists of a Lazy Clause Generation solver over a SAT solver. Lazy Clause Generation is a search technique in Constraint Programming (CP) that adds explanation and learning to a propagation-based solver, which is responsible for narrowing down the range taken by the decision variables.

The implementation of the mathematical model has been carried out using the Google ORTools library [20]. First, we are interested in verifying that all solvers find the same solution for the same problem. Second, we aim to validate the scalability of the solvers as the problem size increases. Finally, we want to test if the team evaluation heuristic influences the execution time.

Table 1. Average execution time and percentage of instances solved optimally by each solver for each combination of classroom size (20, 30, 60), team size (3 to 5) and team evaluation heuristic (Belbin and MBTI)

Sizes		Belbin						MBTI					
		CBC		CP-SAT		SCIP		CBC		CP-SAT		SCIP	
Classroom	Team	Time	%	Time	%	Time	%	Time	%	Time	%	Time	%
20	4	0.11	100	2.82	100	0.77	100	0.12	100	2.97	100	0.59	100
	5	0.49	100	4.84	100	3.34	100	0.53	100	5.01	100	3.30	100
30	3	0.08	100	1.71	100	0.77	100	0.08	100	1.79	100	0.74	100
	5	15.93	100	65.52	100	185.00	100	17.55	100	63.10	100	194.01	100
60	3	1.07	100	15.10	100	11.30	100	1.04	20	15.37	20	14.19	20
	4	92.79	100	443.16	100	2081.971	100	103.22	20	491.34	20	1987.56	20
	5	-	0	-	0	-	0	-	0	-	0	-	0

4.2 Problem Instances

To compare the performance of different solvers, several problem instances have been created. For this, an original dataset was used containing a total of 260 anonymous students from the Tourism degree at the Polytechnic University of Valencia. The dataset includes the results of their Belbin and MBTI tests.

Next, we generated synthetic classrooms by sampling this dataset. This allowed us to create different problem instances with characteristics similar to those that would be found in real classrooms. These instances have been used to carry out experiments and evaluate different solutions under a variety of conditions.

Specifically, the tests have been performed on 30 randomly generated instances for 3 different classroom sizes: 20, 30, and 60 students. Thus, the performance of the different solvers will be observed on a total of 120 different instances in terms of execution time and the solution values obtained. Each of the 120 generated instances can be solved with different constraints regarding the team size. However, it was defined that the possible teams to be formed will have a minimum of 3 students and a maximum number of 5. In particular, experiments have been carried out by just allowing team sizes of 3, 4, and 5 students.

4.3 Results

The experiments were carried out on a machine with 4 cores and 8 GB of RAM. Each type of solver has been executed a total of 5 times on each instance to capture statistical differences in the execution time of the different solvers.

The methodology used to obtain the fastest solver follows a multiproblem analysis methodology, which is common in the field of optimization with meta-heuristics [19]. More specifically, we have carried out an analysis for each family of problems. In this context, a family of problems combines classroom size, allowed team size, and team evaluation heuristic. First, we obtain the average execution

time that each solver has taken to solve each prospective instance. Each family of problems consists of 30 instances. Therefore, for each subproblem and solver we obtain 30 measures (i.e., the average execution time of the solver for each of the 30 instances). Then, we employ a non-parametric test to compare the execution time of the three solvers for each of the subproblems. More specifically, we employ the Friedman test. The Friedman test is the extension for more than two populations of the Wilcoxon signed-rank test, and it either assumes that the distributions are identical (null hypothesis) or at least two solver execution times are different from each other. The Wilcoxon signed-rank test with corrected p-values is employed in case of rejecting the null hypothesis to detect pairs of execution times that are different from each other.

Table 1 shows the experiments' results. The table shows the average execution time, the percentage of problem instances solved optimally for each solver, and a combination of classroom size, team size, and team evaluation heuristic. For each family of problems, we have underlined the statistically better results than the rest using the methodology described above.

First, it is important to emphasize that the solutions obtained by different solvers are the same for a given problem instance and the same combination of classroom size, team size, and team evaluation heuristic. Therefore, we consider that all solvers have achieved the optimal solution whenever the problem could be solved.

As observed, CBC is generally the fastest solver of the three solvers for the instances that we tested. In fact, the results suggest that CBC is orders of magnitude faster than SCIP and CP-SAT. The difference between CBC and the other two solvers is also statistically significant, as suggested by the Friedman and Wilcoxon signed-ranked posthoc tests. Of course, the fact that CBC is faster applies only to the type of problems and the model employed in the experiments. Other solvers may provide better results for other types of problems or models. A fact that seems to stem from the experiments is the necessity to test a model for the team formation problem with different solvers, as there seem to be significant and large differences among different solvers.

As the reader may have observed, the number of decision variables in the proposed model directly depends on the number of teams that can be formed. If no other constraint is provided, this depends on the classroom size and the number of allowed team sizes. We carried out some additional experiments with extra team sizes to analyze how the computation time of the model grows with the number of decision variables. The results of this experiment can be found in Fig. 1. More specifically, the figure shows the average computation time for CBC and the 95% confidence interval for instances with a varying number of decision variables. Please, note that the number of decision variables in the figure is expressed in thousands. The figure shows that the model's execution time seems to grow exponentially with the number of decision variables. This is an expected behavior, as many team formation problem variants are NP-hard problems [14]. In the results, some solvers could not solve some of the larger instances, as it stems from the percentage of instances solved in Table 1. In fact, the problems

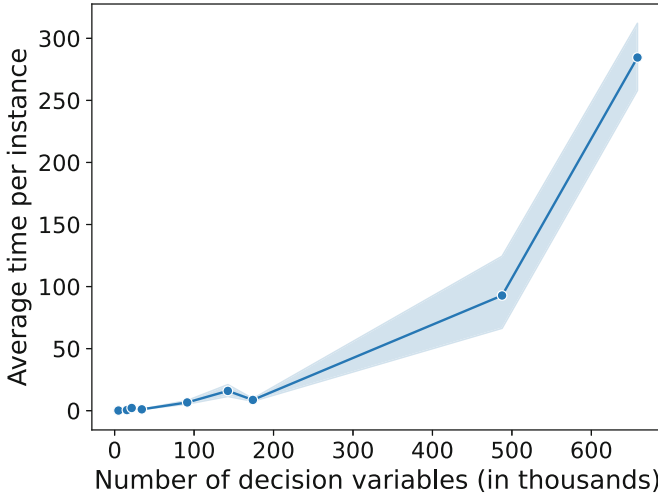


Fig. 1. Average computation time per instance for CBC and 95% confidence interval depending on the number of decision variables

with 60 students and teams of size 5 could not be solved by any of the solvers, given the computational resources available.

This exponential behavior may make it challenging to solve larger problem instances in a reasonable time with common hardware. For instance, instances with 80 to 100 students, which may be found in some Spanish university classrooms. This limitation is in line with this type of models, as mentioned by other authors [1,25]. That opens the room to study alternative ILP formulations that can scale better for larger instances.

Another insight provided by the experiments is related to the two types of team evaluation heuristics employed. While the execution time seems similar for both heuristics in smaller instances, some instances for MBTI seem harder to solve for large instances. For instance, only 20% of the problem instances could be solved optimally with the given resources for 60 students and team sizes of 3 and 4 students, respectively. This result may indicate that the distribution of traits in students may influence the resolution time of models. Thus, the distribution of traits may also influence the scalability of the model and the most appropriate solver for solving the problem.

5 Conclusions

In this paper, we have proposed an integer linear programming model for solving the team formation problem in the classroom. The model incorporates several common constraints in the classroom: disjoint teams, allowed team sizes, students that should be paired together, and students that should be placed in different teams. The objective function of the problem is general, and it can be

extended to take into consideration different criteria. In the experiments carried out in this paper, we employ two team evaluation heuristics that foster heterogeneity of Belbin's role taxonomy and MBTI.

We have conducted experiments with problem instances generated from a real dataset of students' traits regarding Belbin's role taxonomy and MBTI. More specifically, we have created several instances with different classroom sizes. Each instance has been solved employing each of the two team evaluation heuristics, different allowed team sizes, and different non-commercial solvers. The results point out several insights.

Regarding execution time, there may be significant differences between available solvers, making it important to study and identify the most appropriate solver for the variant of team formation in the classroom problem. Another insight comes from the fact that the model's execution time rapidly grows with the problem size. Specifically, based on the resources used in our experiments, no solver was able to solve problem instances with a class size of 60 and a team size of 5. This indicates that it is necessary to study other models that scale better with larger instances. Exact methods like the one proposed in this paper are necessary as they obtain the optimal solution, allowing us to compare various team evaluation heuristics. Thus, it is necessary to study alternative formulations that scale better with the size of the problem for specific team formation in the classroom problems. This is especially the case for large classroom sizes like the ones found in many Spanish universities.

The experiments also suggest that despite the fact that both team evaluation heuristics aim for heterogeneity, the distribution of traits among students may influence the hardness of the problem instance. Therefore, different solvers and models may perform differently under different team evaluation heuristics that foster diversity. In future work, we also plan to propose several integer linear problems for team formation in the classroom problem and study their appropriateness for different problem types.

Acknowledgement. This work was partially supported with grant DIGITAL-2022 CLOUD-AI-02 funded by the European Commission and grant PID2021-123673OB-C31 funded by MCIN/AEI/10.13039/501100011033 and by "ERDF A way of making Europe".

References

1. Aguiar, A., Pinheiro, P.R., Coelho, A.L.V., Nepomuceno, N., Neto, Á., Cunha, R.P.P.: Scalability analysis of a novel integer programming model to deal with energy consumption in heterogeneous wireless sensor networks. In: Le Thi, H.A., Bouvry, P., Pham Dinh, T. (eds.) MCO 2008. CCIS, vol. 14, pp. 11–20. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-87477-5_2
2. Alberola, J.M., Del Val, E., Sanchez-Anguix, V., Palomares, A., Teruel, M.D.: An artificial intelligence tool for heterogeneous team formation in the classroom. *Knowl.-Based Syst.* **101**, 1–14 (2016)

3. Andrejczuk, E., Bistaffa, F., Blum, C., Rodríguez-Aguilar, J.A., Sierra, C.: Synergistic team composition: a computational approach to foster diversity in teams. *Knowl.-Based Syst.* **182**, 104799 (2019)
4. Aranzabal, A., Epelde, E., Artetxe, M.: Team formation on the basis of Belbin's roles to enhance students' performance in project based learning. *Educ. Chem. Eng.* **38**, 22–37 (2022)
5. Belbin, R.M.: *Team Roles at Work*. Routledge (2012)
6. Bestuzheva, K., et al.: The SCIP optimization suite 8.0. arXiv preprint [arXiv:2112.08872](https://arxiv.org/abs/2112.08872) (2021)
7. Campêlo, M., Figueiredo, T.F.: Integer programming approaches to the multiple team formation problem. *Comput. Oper. Res.* **133**, 105354 (2021)
8. Crawford, C., Rahaman, Z., Sen, S.: Evaluating the efficiency of robust team formation algorithms. In: Osman, N., Sierra, C. (eds.) *AAMAS 2016. LNCS (LNAI)*, vol. 10002, pp. 14–29. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46882-2_2
9. Farhangian, M., Purvis, M., Purvis, M., Savarimuthu, B.T.R.: Modeling team formation in self-assembling software development teams. In: *Proceedings of the 2016 International Conference on Autonomous Agents & Multiagent Systems*, pp. 1319–1320 (2016)
10. Forrest, J., Lougee-Heimer, R.: CBC user guide. In: *Emerging Theory, Methods, and Applications*, pp. 257–277. INFORMS (2005)
11. Garshasbi, S., Mohammadi, Y., Graf, S., Garshasbi, S., Shen, J.: Optimal learning group formation: a multi-objective heuristic search strategy for enhancing inter-group homogeneity and intra-group heterogeneity. *Expert Syst. Appl.* **118**, 506–521 (2019)
12. Genova, K., Guliashki, V.: Linear integer programming methods and approaches - a survey. *J. Cybern. Inf. Technol.* **11**(1), 3–25 (2011)
13. John, O.P., Donahue, E.M., Kentle, R.L.: *The Big Five Inventory – Versions 4a and 5*. University of California, Berkeley/Institute of Personality and Social Research, Berkeley (1991)
14. Juárez, J., Santos, C., Brizuela, C.A.: A comprehensive review and a taxonomy proposal of team formation problems. *ACM Comput. Surv. (CSUR)* **54**(7), 1–33 (2021)
15. Mellon, C.: *What are the benefits of group work?* (2015)
16. Meredith, B.R.: *Management teams: why they succeed or fail*. *Hum. Resour. Manage. Int. Digest* (2010)
17. Myers, I.B.: *The Myers-Briggs type indicator: manual* (1962)
18. Partington, D., Harris, H.: Team role balance and team performance: an empirical study. *J. Manage. Dev.* **18**, 694–705 (1999)
19. Peres, F., Castelli, M.: Combinatorial optimization problems and metaheuristics: review, challenges, design, and development. *Appl. Sci.* **11**(14), 6449 (2021)
20. Perron, L.: *Operations research and constraint programming at Google*. In: Lee, J. (ed.) *CP 2011. LNCS*, vol. 6876, pp. 2–2. Springer, Heidelberg (2011). https://doi.org/10.1007/978-3-642-23786-7_2
21. Pieterse, V., Kourie, D.G., Sonnekus, I.P.: Software engineering team diversity and performance. In: *2006 Annual Conference of the South African Institute of Computer Scientists and Information Technologists on IT Research in Developing Countries*, pp. 180–186 (2006)
22. Prada, E.D., Mareque, M., Pino-Juste, M.: Teamwork skills in higher education: is university training contributing to their mastery? *Psicol.: Reflexão e Crítica* **35**(5), 1–13 (2022)

23. Ramchurn, S.D., Polukarov, M., Farinelli, A., Jennings, N., Trong, C.: Coalition formation with spatial and temporal constraints (2010)
24. Sanchez-Anguix, V., Chalumuri, R., Aydođan, R., Julian, V.: A near pareto optimal approach to student-supervisor allocation with two sided preferences and workload balance. *Appl. Soft Comput.* **76**, 1–15 (2019)
25. Tarplee, K.M., Friese, R., Maciejewski, A.A., Siegel, H.J.: Scalable linear programming based resource allocation for makespan minimization in heterogeneous computing systems. *J. Parallel Distrib. Comput.* **84**, 76–86 (2015)
26. Ugarte, N., Aranzabal, A., Arruarte, A., Larrañaga, M.: Using the behavioural tendency of students in a team environment for team formation. In: 2022 IEEE Frontiers in Education Conference (FIE), pp. 1–9. IEEE (2022)
27. Uto, M., Nguyen, D.T., Ueno, M.: Group optimization to maximize peer assessment accuracy using item response theory and integer programming. *IEEE Trans. Learn. Technol.* **13**(1), 91–106 (2019)
28. Vecina, M.Á., et al.: A decision support tool for the static allocation of emergency vehicles to stations. In: García Bringas, P., et al. (eds.) HAIS 2022. LNCS, vol. 13469, pp. 141–152. Springer, Cham (2022). https://doi.org/10.1007/978-3-031-15471-3_13
29. Yannibelli, V., Amandi, A.: A deterministic crowding evolutionary algorithm to form learning teams in a collaborative learning context. *Expert Syst. Appl.* **39**(10), 8584–8592 (2012)