

Journal of the Operational Research Society

ISSN: 0160-5682 (Print) 1476-9360 (Online) Journal homepage: www.tandfonline.com/journals/tjor20

A two-phase binary-goal programming-based approach for optimal project-team formation

Fatih Cavdur , Asli Sebatli , Merve Kose-Kucuk & Cagla Rodoplu

To cite this article: Fatih Cavdur , Asli Sebatli , Merve Kose-Kucuk & Cagla Rodoplu (2019) A two-phase binary-goal programming-based approach for optimal project-team formation, Journal of the Operational Research Society, 70:4, 689-706, DOI: 10.1080/01605682.2018.1457480

To link to this article: https://doi.org/10.1080/01605682.2018.1457480

Published online: 25 Apr 2018.

Submit your article to this journal

Article views: 550

View related articles

View Crossmark data

Citing articles: 4 View citing articles

Full Terms & Conditions of access and use can be found at
https://www.tandfonline.com/action/journalInformation?journalCode=tjor20

JOURNAL OF THE OPERATIONAL RESEARCH SOCIETY
2019, VOL. 70, NO. 4, 689–706
https://doi.org/10.1080/01605682.2018.1457480

THE OPERATIONAL RESEARCH SOCIETY
Taylor & Francis
Taylor & Francis Group

ORIGINAL ARTICLE

Check for updates

A two-phase binary-goal programming-based approach for optimal project-team formation

Fatih Cavdur, Asli Sebatli, Merve Kose-Kucuk and Cagla Rodoplu

Department of Industrial Engineering, Uludag University, Bursa, Turkey

ABSTRACT

This study presents a two-phase binary-goal programming-based approach for solving a novel system design project-team formation problem which involves several restrictions and requirements as well as the preferences of the potential team members. The problem instance considered in this study basically includes two types of allocations, as the allocations of students and academic advisers, which are performed in the first and second phase of the proposed solution approach, respectively. Although it represents a particular case as represented in this study, it can be easily generalised to be used for solving similar project-team formation problems. We implement our methodology on a real-life problem in an academic institution and compare our solutions to the real-life allocations performed manually. It is noted that, in terms of satisfying the goals of the problem, our approach significantly outperforms the real-life allocations. In addition, computational results show our model's ability to solve similar-sized real-life problems in reasonable time periods on an average personal computer, implying its potential for significant savings in terms of the human resources available.

ARTICLE HISTORY
Received 13 December 2016
Accepted 15 March 2018

KEYWORDS
Project-team formation; student-project allocation; integer programming; goal programming; multi-criteria optimisation

1. Introduction

An increasing number of undergraduate programs, especially the colleges of engineering, require their students to take classes in which the students work on some projects as teams before they graduate. The aim of this type of classes, sometimes referred as senior projects or (integrated) system design courses, is to provide students an opportunity to work on a real-life problem where they can not only utilise all the capabilities they had gained during their undergraduate educations, but also get some more experience about working in teams.

Such a project is usually performed as a team of several students and academic advisers. Sometimes some other people might also be in the team, such as the advisers from the organisations where the projects are to be implemented. We thus note that since such a problem involves formation of project teams, it can also be considered as a team formation problem of a specific type, which, according to the best of our knowledge, is not considered in previous studies in that context. In a general team formation problem, several subsets of different groups of people with various qualifications are combined together to form the best teams possible. In the simplest case, it is required that each team includes the minimum necessary qualifications. It is, however, usually necessary to consider additional constraints to form project teams as homogeneous as possible, especially if the desired requirements for each project are not explicitly defined. We can optimise these constraints with an appropriate objective function to minimise the occurrences of some undesired situations (i.e., unbalanced groups etc.). Note that, these qualifications might be further classified as discrete (categorical) or continuous (hierarchical) qualifications for which the corresponding expertise, knowledge or skill levels are represented with binary or continuous parameters, respectively. In the first case (discrete or categorical qualifications), people are assumed to be either no qualified at all (zero) or perfectly qualified (one), whereas in the second case (continuous or hierarchical qualifications), some values in between are possible (i.e., 50%) to represent the corresponding expertise, knowledge or skill level. We also note that it is quite likely to cause infeasiblities in such a team formation problem by increasing the number of qualifications or considering additional constraints, in which case, a good solution can be obtained by allowing some relaxations on the constraints.

In this study, we consider a conceptually similar problem to form the system design course project teams of students and academic advisers (from now on, referred as advisers only). Since the real-life team formation is performed in two phases, in which the allocations of students and advisers are performed, respectively, we accordingly design our solution approach as a two-phase allocation process to perform the corresponding allocations.

In the first phase of our approach, students are allocated to project teams. One of the most important criteria considered in the allocations is the preferences

CONTACT Fatih Cavdur fatihcavdur@gmail.com, fatihcavdur@uludag.edu.tr

© Operational Research Society 2018

of the students. Consideration of student preferences is very important since the students are expected to be more successful if they work on the projects they are interested. In addition to the preferences of students, there are also other criteria considered for the allocation process, such as having an evenly distributed average grade point average (GPA) across the teams, the desired number of students and the other necessary qualifications in each team. It is particularly important to have balanced teams in terms of student qualifications for the sake of fairness. In other words, since all teams are expected to work on projects requiring approximately similar experience, expertise or knowledge levels (undergraduate senior projects), it is crucial that all necessary qualifications are included in each team. Note, however, that it might not always be possible to do so due to the shortages of the necessary qualification of potential team members. In such cases, it is aimed at including all necessary qualifications in each team as much as possible by distributing these qualifications homogeneously among teams. Even if potential team members are all qualified, using such a procedure results in perfect satisfaction of all qualifications by all teams.

We perform the allocations of advisers to project teams in the second phase in a similar fashion; that is, considering the preferences of the advisers as well as some other criteria details of which are explained in the following pages. The overall solution approach is illustrated in Figure 1. The problem data include the list of projects with their short descriptions, preferences of students and advisers for the projects, and some student qualifications. Before we implement our solution approach, problem data are pre-processed to organise the preferences of students and advisers. This pre-processing, as detailed in the following pages, is necessary to obtain a consistent preference matrix. The organised data are then used as the inputs of our goal programming models of phase (i) and phase (ii) in which the allocations of students and advisers are performed, respectively. Note that, although we can model and solve the problem in a single phase only, solving it in two phases, as it is also performed in the real-life team formation process, is preferable in terms of computational issues, especially for larger-size problems. We illustrate our solution approach to form the system design course project teams in an academic institution and compare our solutions to the real-life allocations. Note that an experienced person (from now on, referred as the human expert) performs the real-life allocations manually by considering several problem criteria as explained in the Problem Context section of the manuscript in more detail.

One of the most important issues in such a project-team formation problem is about the fairness of allocations. Although it is possible to consider different

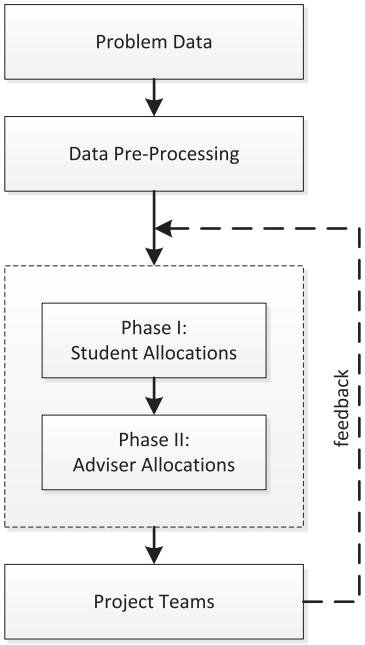


Figure 1. Overall solution approach.

approaches, we highlight two points in our team formation process to ensure that the allocations are fair to team members as the (i) homogeneity of teams and (ii) preferences of potential team members are considered. The homogeneity is important in such project-team formation problems since each team is supposed to be equally qualified to perform similar project tasks. In other words, it is not desired to form some project teams which are more qualified than the others. Similarly, consideration of the preferences of potential team members is also very important to help achieving fairness in the allocation process. The remainder of the paper is organised as follows. Section 2 summarises previous studies on similar problems and presents a brief introduction to goal programming. Section 3 gives the details of the problem considered in this study. Following Sections 4 and 5 defines the methodology and presents the results, respectively. Section 5 includes a brief discussion about solution efficiency, before we present the concluding remarks in the final section (Section 6) of our paper.

2. Literature review and research background

Although the project-team formation problem, as presented in this study, has not yet been considered in previous studies, according to the best of our knowledge, it has some correspondences in the related literature as shortly summarised in this section.

A simplified version of the problem considered in this study is referred as the Student-Project Allocation with Preferences (SPA-P) problem in which there is a set of available projects offered by a set of professors where each student has a preference over the projects

and each professor has a preference over the students. Sometimes lower-upper bounds for the number of students in a team and for the number of projects of a professor. This version of the problem might be considered as a matching (two-side matching) problem where the members of two disjoint sets are matched, such as the students and professors in our case, with its roots go back to the well-known Stable-Marriage Problem (see Gale & Shapley, 1962; Gusfield & Irwing, 1989 for details). Some efficient algorithms are presented for this version of the problem in the literature as in Abraham, Irwing and Manlove (2007). For instance, finding a maximum stable matching is shown to be APX-hard, however, a polynomial time approximation algorithm is presented by Manlove and O'Malley (2008), and an improvement for the lower and upper bounds for the algorithm is given by Iwama, Miyazaki and Yanagisawa (2012).

In addition to these studies, developing mathematical models for the simplified versions of the problem considered in this study goes decades back. Proll (1972), for instance, presents a binary-integer programming model to assign projects to students. Another study by Anwar and Bahaj (2003) extends the study of Proll (1972) by considering additional problem parameters, such as student disciplines. In a more recent study, Agustin-Blas et al. (2011) adapt the machine-part cell formation problem for team formation, develop a genetic algorithm to solve the problem and apply their approach on several problem instances one of which is a real-life teaching group formation problem in a university.

The project-team formation problem presented in this study can be considered as an extended version of such simpler problems, which, in addition to the preferences of students and professors, requires the consideration of several other criteria details of which are explained in the following pages. Since such a problem involves the formation of project teams, it can also be considered as a team formation problem of a specific type. A review of several general team formation methods are summarised by Bacon et al. (2001), such as random assignments and self-selection. In the problem considered in this study, however, it is usually not possible to form project teams which perfectly satisfy all (probably) conflicting criteria of the problem. It is thus desired to find a solution which is feasible (does not violate the hard constraints) and also fair to everyone (satisfy the soft constraints as much as possible).

A final remark for the student-project team formation problem can be made from the perspective of educational sciences. We can refer to the studies of the field for a more detailed discussion on how to form effective student teams. The study of Oakley et al. (2004) defines some guidelines on the topic. More recently, Borrego et al. (2013) present a review where the authors consider the applying the team effectiveness theory from industrial and organisational psychology for engineering student project teams. As we note in most team formation problems, it might not be sufficient to consider only quantitative qualifications to form successful project teams. In other words, it might be necessary to take some psychological criteria into account, such as student characters, to ensure that they can successfully work together (see Blowers, 2003; Huxham & Land, 2000 for details). Some studies also utilise social network analysis for team formation as in Wi, Oh, Mun and Jung (2009a) and Wi, Mun, Oh and Jung (2009b). Even if all qualifications are quantitative, we can still form project teams differently with respect to different criteria, such as maximally diverse or evenly skilled teams as in Hubscher (2010), which might constitute another research problem for educational sciences.

From a more general perspective, we note that skilled-workforce planning and scheduling studies are also related to our problem since it involves the formation of teams each of which should include people with particular skills or qualifications. The first type of these problems require the corresponding skills to be included in a team. In the problems of second type, although it is not a requirement, it is desired to have these skills in a team. In some other cases, a team of specialised workforce is necessary, such as the personnel planning problems in the airline and railway industries. (see Bengtsson et al., 2007; Yan & Chang, 2002 for some examples). Another well-known problem in the health sector is the nurse scheduling or nurse rostering problem is about finding an optimal schedule subject to some hard and soft constraints. A review on the nurse rostering problem is presented in the study of Cheang et al. (2003). The problem considered in this study has some similarities to different types of skilled-workforce problems. Some skills are required in each team (i.e., the min-max number of students in each team), whereas some others are desired (i.e., language or programming skills) which are employed via the hard and soft constraints, respectively. It is moreover necessary to form a team of specialised workforce (i.e., a team of students and advisers with necessary skills or qualifications). The interested reader can refer to the studies of Ernst et al. (2004) and Van den Bergh, Belien, De Bruecker, Demeulemeester and De Boeck (2013) for comprehensive reviews of skilled-workforce planning and scheduling problems.

We note that, for solving such problems, it is usually necessary to utilise some multi-criteria decision-making (MCDM) methodology as the problems involve more than one criterion. The problem considered in this study, for instance, include several criteria, such as satisfying the preferences of students and advisers, including necessary qualifications in each team and ensuring some level of fairness for students and advis-

ers. In addition to its extensive use in many different type of problems, we also note some studies utilising MCDM methodologies for project-team formation with some similarities to the problem presented in this study. Baykasoglu et al. (2007), for instance, consider a project-team selection problem in business where a fuzzy multi-objective model is used with two objectives as sustainability and team size. In another study, Badri, Davis and Davis (2001) present a binary-goal program for project selection with six goals which are related to the total benefits, costs (hardware, software and other), risks and preferences. In another study, Meskens, Duvivier and Hanset (2013) consider a multi-objective operating room scheduling problem with three objectives as makespan, overtime and affinities where the authors use constraint programming for solving the problem. In the study of Azaiez and Al Sharif (2005), a binary-goal programming model is presented for nurse scheduling where the authors consider three type of objectives which are classified as hospital objectives, ergonomic considerations and nurse preferences.

We note that, just as the project-team formation problem considered in this study, goal programming is widely used for solving such MCDM problems with its roots go back to the studies of Charnes et al. (1955) and Charnes and Cooper (1957). Its extensive use in different problems are presented in comprehensive reviews of Ignizio (1978), Tamiz, Jones and Romero (1998) and Aouni and Kettani (2001). A generic goal programming formulation has an objective (achievement) function to minimise,

$$\min z = f(\mathbf{n}, \mathbf{p}), \tag{1}$$

where $\mathbf{n}$ and $\mathbf{p}$ are vectors of negative and positive deviational variables, respectively. The objective (achievement) function given by Equation 1 is minimised subject to the goal constraints, where each goal $q \in Q$ has an achieved value of $g(\mathbf{x})$ on its underlying criterion,

$$g_q(\mathbf{x}) + n_q - p_q = b_q, \quad \forall q \in Q \tag{2}$$

the other constraints to ensure the feasibility of the solution represented by feasible space F

$$\mathbf{x} \in F \tag{3}$$

and the non-negativity of deviational variables,

$$n_q, p_q \geq 0, \quad \forall q \in Q \tag{4}$$

We can define some variants of goal programming with respect to distance metrics, such as the weighted (Archimedean or non-pre-emptive), lexicographic (pre-emptive) and min-max (Chebyshev) goal programming. In a weighted or non-pre-emptive goal program, the relative importance of each goal is determined with its weight, as a result of which it can be solved as a single objective mathematical program. A lexicographic or pre-emptive goal programming model, on the other hand, is more suitable for the cases where it is not possible (or desired) to determine the relative importance of each goal, yielding *extremely* more or less important goals to be considered. To solve a lexicographic goal program, we can sequentially solve mathematical programs each with a single-objective starting with that of the most important goal. The optimal objective function value is then used as a constraint in the solution of the mathematical program with the next most important goal. The solution procedure continues until all objectives are optimised in the order of their importance. The min–max or Chebyshev variant considers the worst satisfied goal by minimising the maximal deviation from the goals. Some other variants can also be defined, such as the fuzzy, integer/binary and fractional goal programming. As their name suggest, these variants are based on the variables and goals of program rather than the distance metrics considered in the model. For more details on these classifications, we can refer to Jones & Tamiz (2010). Some new variants of goal programming (i.e., extended, multi-choice, multi-segment, meta, etc.) are also proposed more recently by Romero (2001), Uria, Caballero, Ruiz and Romero (2002), Chang (2007a, 2007b, 2008), Liao (2009), Lin, Nagalingam and Lin (2009) and Chang (2011), increasing the potential applicability of goal programming in future studies.

3. Problem context

In this section, we present the details of the problem considered in this study. Although it seems to represent a case of an academic institution as presented in this paper, it can be modified easily for a more general implementation as well, since similar problems with minor differences are quite common in most academic institutions.

The problem considered in this study involves the formation of the system design course project teams of students and advisers. In particular, we explain the case of an academic institution since it allows us to illustrate the approach proposed in this study for a comprehensive allocation process.

As depicted in Figure 2, we have two different inputs for the problem; a set of available projects with their requirements and a set of different types of persons (i.e., students and advisers) with their qualifications and preferences. Ideally, it is desired to have perfect information to define the problem inputs (i.e., requirements and qualifications) clearly as much as possible. In other words, in an ideal setting, all projects have clearly defined requirements which are perfectly satisfied again with clearly defined qualifications. We can then match these requirements and qualifications for the allocations assuming that all requirements are satisfied with

the qualifications of the available persons. In addition to the requirements and qualifications, persons might have preferences over the projects, another important issue to be considered during the team formation process.

Although it is conceptually possible to clearly define project requirements and corresponding qualifications, in real-life it is not always very easy or practical to do so, especially due to the lacking of a perfect matching between the requirements and qualifications. As a result, by defining somehow more general requirements and qualifications, the problem is usually transformed to an easier or more practical form. In some cases, it is simply desired to have all qualification in all teams. It is also the case for the system design course project team formation problem considered in this study.

Since it is not practical to define the requirements for all projects and the corresponding qualifications for all students, a more general project requirement-student qualification setting is used by the institution as shown in Figure 3 where all project teams are expected to have all qualifications. We note that such a setting is particularly applicable in cases where the projects and persons represent a somehow homogeneous structures with respect to their requirements and qualifications, respectively. In our case, since all projects are system design course projects and all students are senior-level engineering students, the setting applied by the institution as presented in Figure 3 seems applicable.

Although the human expert uses experience rather than a particularly defined method, the setting shown in Figure 3 represents the criteria considered by the human expert to perform the real-life allocations. In fact, Figure 3 simply defines the relationships between the skill determinants and skill consequences in our case. As seen in Figure 3, there are three general project requirements implying that each team expected to have the capability of performing tasks about (i) engineering applications, (ii) computer applications, (iii) business communication and literature review which are assumed to be related to (i) GPA and computer programming, (ii) computer programming, general software and database and (iii) language skills, respectively.

As the real-life team formation is performed in two phases, in which the allocations of students and advisers are performed, respectively, we accordingly design our solution approach as a two-phase allocation process to perform the corresponding allocations.

In the first phase of the process, students are allocated to project teams. There are several criteria considered in the allocations some of which are related to the skills corresponding to the project requirements as depicted in Figure 3 (i.e., GPA, computer programming, general software, database and language skills). There are also additional criteria representing the other general problem parameters such as the student preferences and the number of students in each team.

Table 1. Problem data.

Data-set	Number of students	Number of advisers	Number of projects
1	78	11	23
2	50	11	18
3	70	12	25

After students are allocated to projects, adviser allocations are performed in the second phase. Since all projects are performed as part of an undergraduate (senior) level system design course, it seems reasonable to assume that all advisers are capable of administrating all projects. As a result, there are no skill determinant–skill consequence relationships defined for advisers and projects (i.e., research topics, classes etc.). We use several other criteria for adviser allocations such as their preferences and the number of teams each adviser administrates. Note that the aforementioned problem criteria of both phases used in developing the proposed solution approach are exactly the ones considered by the human expert in performing the real-life allocations.

We use three different data-sets each of which contains the problem data for a particular year. A summary on these data-sets are shown in Table 1. The problem data include the list of projects with a short description and the required student skills for each project (obtained from the organisations where the project is to be implemented), the preferences of students and advisers for the projects and student qualifications (obtained from student and adviser questionnaires and student transcripts). Before we implement our solution approach, problem data are pre-processed to organise the preferences of students and advisers. This pre-processing is necessary to obtain a consistent preference matrix. The (i, j)th entry of the preference matrix of students (and also advisers), let it be $p_{ij}^{(s)}$ (and $p_{ij}^{(a)}$ for advisers), simply shows the preference of student (adviser) i for project j, a smaller number implying a higher preference. Consider a student (say, student 1), for instance, who indicates three preferences as $p_{11}^{(s)} = 1$, $p_{13}^{(s)} = 2$ and $p_{15}^{(s)} = 3$; meaning that the student mostly wants to work on the first project (a preference entry of 1 for project 1), then the third project (a preference entry of 2 for project 3), and then the fifth project (a preference entry of 3 for project 5).

Note that, to be meaningful, none of the elements of a preference matrix can be null, and yet, there are null entries in the matrices because of the following two reasons. The first one is about the different number of preferences. Although all students and advisers are asked to indicate a particular number of preferences, say, from 1 to 5; some students and/or advisers might not indicate that particular number of preferences of

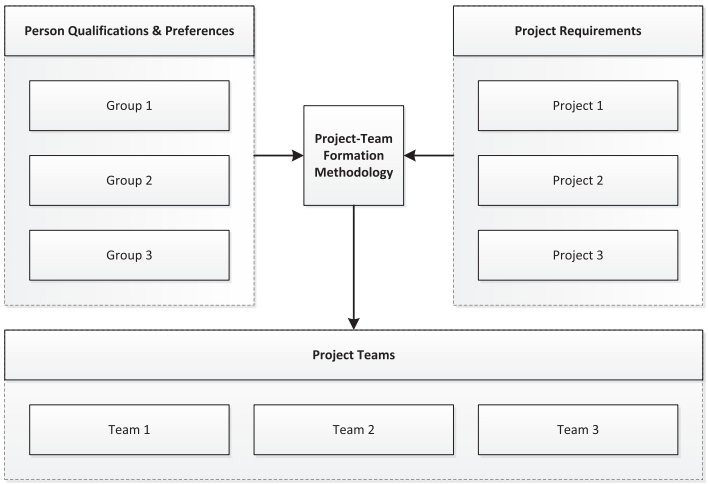


Figure 2. Project-team formation approach.

theirs. That is, while a student might enter her preferences from 1 to 5, for instance, another one might enter hers from 1 to 3 or from 1 to 10. Secondly, it is not very practical to ask all students and advisers to indicate their preferences for all projects from 1 to the number of projects, especially when the number of projects is relatively high. As a result, even if all students and advisers indicate the correct number of preferences of theirs, there are some null entries in the preference matrices. To address this issue about the inconsistencies in the preference matrices, we propose the following expressions to update all of the null entries in the matrices. In particular, by defining $p_i^{s\text{-max}}$ as

$$p_i^{s\text{-max}} = \max_j \left\{ p_{ij}^{(s)} \right\}, \quad \forall i \tag{5}$$

we update the null entries in the matrix as

$$p_{ij}^{(s)} := p_i^{s\text{-max}} + 1, \quad \forall i, j : p_{ij}^{(s)} = \varnothing \tag{6}$$

for students, and similarly, by defining $p_i^{a\text{-max}}$ as

$$p_i^{a\text{-max}} = \max_j \left\{ p_{ij}^{(a)} \right\}, \quad \forall i \tag{7}$$

we update the null entries in the matrix as

$$p_{ij}^{(a)} := p_i^{a\text{-max}} + 1, \quad \forall i, j : p_{ij}^{(a)} = \varnothing \tag{8}$$

for advisers. Note that, according to the expression in Equation 6 and Equation 8, all the null entries of a student or adviser are set as one plus the maximum entry value for that particular student or adviser, considering the fact that the preference for a null entry should be greater than the maximum entry value for a person, or in other words, implying that *the person's motivation for being in the team of the corresponding project should be less than for the projects the person indicates any preference*. After updating the preference

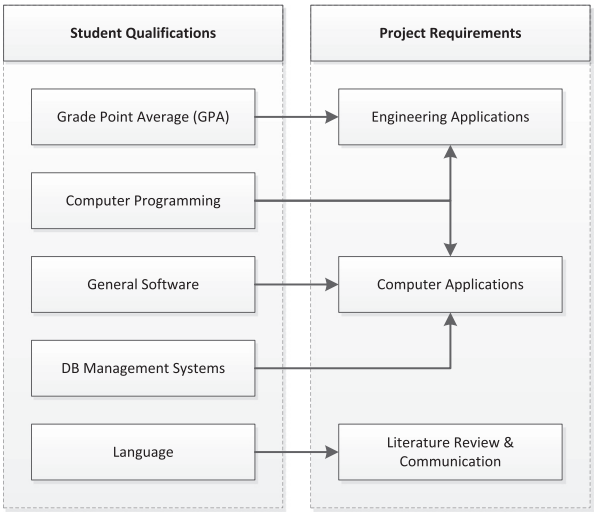


Figure 3. Student qualifications – Project requirements relationships.

matrices and organising the other data (i.e. tabulating student skills etc.), they are used as the inputs for the models defined in the following subsections.

4. Methodology

We present the details of the proposed solution approach in this section. Since the overall real-life problem involves a two-phase sequential-allocation process, the problem considered in this study is also solved in two phases in which the allocations of students and advisers are performed, respectively, as detailed in the following sub-sections.

4.1. Phase I: Student allocations

We have the following definitions for the first problem.

- i: student index, $i = 1, \ldots, I_S$.
- j: project index, $j = 1, \ldots, J$.

- k: preference index, $k = 1, \ldots, K_S \leq J$.
- $p_{ij}^{(s)}$: preference, (i,j)th entry of the matrix represents the preference of student i for project j.
- $K_{(i,j)}$: a subset of the elements of index k whose each element represents the preference of student i for project j, $K_{(i,j)} = \left\{k \mid k = p_{ij}^{(s)}\right\}$.
- g_i: GPA, ith entry of the vector represents the GPA of student i.
- l_i: language skills, ith entry of the vector represents the language grade of student i.
- q_i^1: computer programming skills, ith entry of the binary-vector equals 1 if student i has necessary computer programming skills and 0 otherwise.
- q_i^2: general software skills, ith entry of the binary-vector equals 1 if student i has necessary general software skills 0 otherwise.
- q_i^3: database skills, ith entry of the binary-vector equals 1 if student i has necessary database skills and 0 otherwise.
- N_S: desired number of students in a project team.
- $N_{S\text{-min}}$ ($N_{S\text{-max}}$): allowed lower (upper) bound on the number of students in a project team.
- G_T: desired average GPA in a project team.
- $G_{T\text{-min}}$ ($G_{T\text{-max}}$): allowed lower (upper) bound on the average GPA in a project team.
- L_T: desired average language grade in a project team.

We finally define the decision variables by letting $x_{ijk}^{(s)}$ equal 1 if student i is allocated to project j that is his/her kth choice and 0 otherwise. Our first set of hard constraints requires that each student is allocated to a project, and one project only.

$$\sum_{j=1}^{J}\sum_{k=1}^{K_S} x_{ijk}^{(s)} = 1, \ \forall i. \tag{9}$$

After ensuring that each student is allocated to a project, and one project only, we want to satisfy the preferences of the students as much as possible via the following set of soft constraints.

$$\sum_{j=1}^{J_S}\sum_{k\in K_{(i,j)}} k x_{ijk}^{(s)} \approx 1, \ \forall i. \tag{10}$$

By defining the corresponding deviational variables we can write

$$\sum_{j=1}^{J}\sum_{k\in K_{(i,j)}} k x_{ijk}^{(s)} + d_i^{1-} - d_i^{1+} = 1, \ \forall i \tag{11}$$

where only the positive deviational variables are included in the objective (achievement) function to maximise the satisfaction of the student preferences goal. The number of students allocated to a project team should be N_S as expressed by the following set of constraints.

$$\sum_{i=1}^{I_S}\sum_{k=1}^{K_S} x_{ijk}^{(s)} \approx N_S, \ \forall j. \tag{12}$$

By adding the corresponding deviational variables, it is rewritten as

$$\sum_{i=1}^{I_S}\sum_{k=1}^{K_S} x_{ijk}^{(s)} + d_j^{2-} - d_j^{2+} = N_S, \ \forall j, \tag{13}$$

where both the positive and negative deviational variables are included in the objective (achievement) function to maximise the satisfaction of the number of students per team goal. Note that the above soft constraints might not prevent the construction of the teams with much less or much more members than N_s. To do so, we can write the following auxiliary set of hard constraints to make sure that the number of students in a team is between $N_{S\text{-min}}$ and $N_{S\text{-max}}$.

$$N_{S\text{-min}} \leq \sum_{i=1}^{I_S}\sum_{k=1}^{K_S} x_{ijk}^{(s)} \leq N_{S\text{-max}}, \ \forall j \tag{14}$$

In addition to the soft constraints about the preferences and the number of students in a team, it is also important to consider the distributions of the students in terms of their qualifications so that either the qualification levels of teams are not much different from each other or each team is desired to have the minimum qualification levels. One of these qualifications is GPA. It is desired to have an average GPA of G_T for each team, that is approximately the average GPA of the class.

$$\sum_{i=1}^{I_S}\sum_{k=1}^{K_S} g_i x_{ijk}^{(s)} - \sum_{i=1}^{I_S}\sum_{k=1}^{K_S} G_T x_{ijk}^{(s)} \approx 0, \ \forall j. \tag{15}$$

By adding the corresponding deviational variables, we have

$$\sum_{i=1}^{I_S}\sum_{k=1}^{K_S} g_i x_{ijk}^{(s)} - \sum_{i=1}^{I_S}\sum_{k=1}^{K_S} G_T x_{ijk}^{(s)} + d_j^{3-} - d_j^{3+} = 0, \ \forall j, \tag{16}$$

where both the positive and negative deviational variables are included in the objective (achievement) function to maximise the satisfaction of the corresponding student qualification (average GPA per team) goal. As in case for the number of students in each team, we add the supporting set of hard constraints to set limits for

the average GPA as well.

$$\sum_{i=1}^{I_S}\sum_{k=1}^{K_S} G_{T\text{-min}} x_{ijk}^{(s)} \le \sum_{i=1}^{I_S}\sum_{k=1}^{K_S} g_i x_{ijk}^{(s)} \le \sum_{i=1}^{I_S}\sum_{k=1}^{K_S} G_{T\text{-max}} x_{ijk}^{(s)}, \quad \forall j. \tag{17}$$

Similarly, it is desired to have an average language grade that is greater than or equal to L_T for each team.

$$\sum_{i=1}^{I_S}\sum_{k=1}^{K_S} l_i x_{ijk}^{(s)} - \sum_{i=1}^{I_S}\sum_{k=1}^{K_S} L_T x_{ijk}^{(s)} \approx 0, \quad \forall j. \tag{18}$$

Adding the corresponding deviational variables yields

$$\sum_{i=1}^{I_S}\sum_{k=1}^{K_S} l_i^{x_{ijk}(s)} - \sum_{i=1}^{I_S}\sum_{k=1}^{K_S} L_T x_{ijk}^{(s)} + d_j^{4-} - d_j^{4+} = 0, \quad \forall j, \tag{19}$$

where only the negative deviational variables are included in the objective (achievement) function to maximise the satisfaction of the corresponding student qualification (average language grade per team) goal. Note that the average language grade per team is treated differently from the average GPA per team goal since the decision maker wants to balance the teams in terms of their average GPA, whereas a minimum average language grade is desired for each team. Similarly, we can define other student qualifications expressed as computer programming, general software and database skills represented by the following sets of soft constraints. Note that, differently from GPA and language grade (which are modelled as continuous or hierarchical skills), a binary or categorical skill-level representation is used for these qualifications. It is desired that each team has the corresponding qualifications.

$$\sum_{i=1}^{I_S}\sum_{k=1}^{K_S} q_i^1 x_{ijk}^{(s)} \approx 1, \quad \forall j, \tag{20}$$

$$\sum_{i=1}^{I_S}\sum_{k=1}^{K_S} q_i^2 x_{ijk}^{(s)} \approx 1, \quad \forall j, \tag{21}$$

$$\sum_{i=1}^{I_S}\sum_{k=1}^{K_S} q_i^3 x_{ijk}^{(s)} \approx 1, \quad \forall j. \tag{22}$$

By adding the corresponding deviational variables, we obtain the following sets of constraints

$$\sum_{i=1}^{I_S}\sum_{k=1}^{K_S} q_i^1 x_{ijk}^{(s)} + d_j^{5-} - d_j^{5+} = 1, \quad \forall j, \tag{23}$$

$$\sum_{i=1}^{I_S}\sum_{k=1}^{K_S} q_i^2 x_{ijk}^{(s)} + d_j^{6-} - d_j^{6+} = 1, \quad \forall j, \tag{24}$$

$$\sum_{i=1}^{I_S}\sum_{k=1}^{K_S} q_i^3 x_{ijk}^{(s)} + d_j^{7-} - d_j^{7+} = 1, \quad \forall j, \tag{25}$$

where only the negative deviational variables are included in the objective (achievement) function to maximise the satisfaction of the corresponding student qualifications (computer programming, general software and database skills) goals. We finally have

$$d_i^{1\pm} \ge 0, \quad \forall i, \tag{26}$$

$$d_j^{2\pm}, \ldots, d_j^{7\pm} \ge 0, \quad \forall j. \tag{27}$$

The terms of the objective (achievement) function representing the goals of the first phase are given by

$$\min z_1^{(s)} = \sum_{i=1}^{n_I} d_i^{1+} \tag{28}$$

$$\min z_2^{(s)} = \sum_{j=1}^{n_J} d_j^{2-} - d_j^{2+} \tag{29}$$

$$\min z_3^{(s)} = \sum_{j=1}^{n_J} d_j^{3-} - d_j^{3+} \tag{30}$$

$$\min z_4^{(s)} = \sum_{j=1}^{n_J} d_j^{4-} \tag{31}$$

$$\min z_5^{(s)} = \sum_{j=1}^{n_J} d_j^{5-} \tag{32}$$

$$\min z_6^{(s)} = \sum_{j=1}^{n_J} d_j^{6-} \tag{33}$$

$$\min z_7^{(s)} = \sum_{j=1}^{n_J} d_j^{7-} \tag{34}$$

The objective (achievement) function of the weighted goal program is then given by

$$\begin{aligned} \min z_w = {} & w_1^{(w)} \left(\sum_{i=1}^{I_S} d_i^{1+} \right) \\ & + w_2^{(w)} \left(\sum_{j=1}^{J} d_j^{2-} + d_j^{2+} \right) \\ & + w_3^{(w)} \left(\sum_{j=1}^{J} d_j^{3-} + d_j^{3+} \right) \\ & + w_4^{(w)} \left(\sum_{j=1}^{J} d_j^{4-} \right) + w_5^{(w)} \left(\sum_{j=1}^{J} d_j^{5-} \right) \\ & + w_6^{(w)} \left(\sum_{j=1}^{J} d_j^{6-} \right) + w_7^{(w)} \left(\sum_{j=1}^{J} d_j^{7-} \right), \end{aligned} \tag{35}$$

where $w_1^{(w)}, \ldots, w_7^{(w)}$ are the weights representing the relative importance of the corresponding goals whereas, for a particular lexicographic importance of goals, the objective (achievement) function of the lexicographic (pre-emptive) goal program is then written as

$$\text{lex min } z_p = \left\{ \begin{array}{c} \left(\sum_{i=1}^{I_S} d_i^{1+}\right) \\ \left(\sum_{j=1}^{J} d_j^{2-} + d_j^{2+}\right) \\ \left(\sum_{j=1}^{J} d_j^{3-} + d_j^{3+}\right) \\ \left(\sum_{j=1}^{J} d_j^{4-}\right) \\ \left(\sum_{j=1}^{J} d_j^{5-}\right) \\ \left(\sum_{j=1}^{J} d_j^{6-}\right) \\ \left(\sum_{j=1}^{J} d_j^{7-}\right) \end{array} \right\} \tag{36}$$

or equivalently as

$$\begin{aligned} \min z_p = {} & w_1^{(p)} \left(\sum_{i=1}^{I_S} d_i^{1+}\right) \\ & + w_2^{(p)} \left(\sum_{j=1}^{J} d_j^{2-} + d_j^{2+}\right) \\ & + w_3^{(p)} \left(\sum_{j=1}^{J} d_j^{3-} + d_j^{3+}\right) \\ & + w_4^{(p)} \left(\sum_{j=1}^{J} d_j^{4-}\right) + w_5^{(p)} \left(\sum_{j=1}^{J} d_j^{5-}\right) \\ & + w_6^{(p)} \left(\sum_{j=1}^{J} d_j^{6-}\right) + w_7^{(p)} \left(\sum_{j=1}^{J} d_j^{7-}\right), \end{aligned} \tag{37}$$

where $w_1^{(p)} \gg \ldots \gg w_7^{(p)}$ are the weights representing the lexicographic importance (priorities) of the corresponding goals.

4.2. Phase II: Adviser allocations

After allocating students to projects and forming teams, the next step is to allocate the required number of advisers to each project team. We have the following definitions for the problem. Note that, where appropriate, a similar notation as in the first phase is used to represent the common structures of the problems of both phases. Also note that since the project teams are now formed after the allocation of students and the advisers actually are allocated to these teams, in this phase, we use the term *project team* rather than just *project*, while referring the allocations as it might be considered more appropriate. We however use the same notation (i.e., J) to represent both the number of projects in the first phase and the number of project teams in the second phase since all projects are allocated in the first phase and the total number of projects allocated in the first phase simply represents the total number of project teams in the second phase.

- i: adviser, $i = 1, \ldots, I_A$.
- j: project team, $j = 1, \ldots, J$.
- k: preference, $k = 1, \ldots, K_A \leq J$.
- $p_{ij}^{(a)}$: preference, (i,j)th entry of the matrix represents the preference of adviser i for project team j.
- $K_{(i,j)}$: a subset of the elements of index k whose each element represents the preference of adviser i for project team j, $K_{(i,j)} = \left\{k \mid k = p_{ij}^{(a)}\right\}$.
- N_T: desired number of project teams allocated to an adviser.
- $N_{T\text{-min}}$ ($N_{T\text{-max}}$): allowed lower (upper) bound on the number of project teams an adviser allocated.
- N_A: required number of project teams allocated to an adviser.

We let $x_{ijk}^{(a)}$ equal 1 if adviser i is allocated to project team j that is his/her kth preference and 0 otherwise. We note that although each student *must* be allocated to a project and one project only, an adviser might be *desired* to be allocated to more than one project team. The following soft constraint set aims at allocating each adviser is allocated to N_T project teams.

$$\sum_{j=1}^{J} \sum_{k=1}^{K_A} x_{ijk}^{(a)} \approx N_T, \quad \forall i. \tag{38}$$

By adding the corresponding deviational variables, we obtain the set of constraints in Equation 39.

$$\sum_{j=1}^{J} \sum_{k=1}^{K_A} x_{ijk}^{(a)} + d_i^{1-} - d_i^{1+} = N_T, \quad \forall i \tag{39}$$

where both the positive and negative deviational variables are included in the objective (achievement) function. Note that the following set of auxiliary constraints restricts the number of project teams allocated to an adviser by ensuring that it is greater than or equal to $N_{T\text{-min}}$ and less than or equal to $N_{T\text{-max}}$.

$$N_{T\text{-min}} \leq \sum_{j=1}^{J} \sum_{k=1}^{K_A} x_{ijk}^{(a)} \leq N_{T\text{-max}}, \quad \forall i. \tag{40}$$

As in the student allocations process, we also want to satisfy the preferences of advisers as much as possible.

$$\sum_{j=1}^{J} \sum_{k \in K_{(i,j)}} k x_{ijk}^{(a)} \approx 1, \quad \forall i \tag{41}$$

Similarly, by adding the corresponding deviational variables, we obtain the constraint set in Equation 42.

$$\sum_{j=1}^{J} \sum_{k \in K_{(i,j)}} k x_{ijk}^{(a)} + d_i^{2-} - d_i^{2+} = 1, \quad \forall i, \tag{42}$$

where only the positive deviational variables are included in the objective (achievement) function. The number of advisers allocated to a project team must be N_A as expressed by the following set of constraints in Equation 43.

$$\sum_{i=1}^{I_A} \sum_{k=1}^{K_A} x_{ijk}^{(a)} = N_A, \quad \forall j. \tag{43}$$

In addition, note that the advisers in each project team must be different.

$$\sum_{k=1}^{K_A} x_{ijk}^{(a)} \leq 1, \quad \forall i, j. \tag{44}$$

We finally have

$$d_i^{\pm 1}, d_i^{2\pm} \geq 0, \quad \forall i \tag{45}$$

The terms of the objective (achievement) function are given by

$$\min z_1^{(a)} = \sum_{i=1}^{I_A} d_i^{1-} - d_i^{1+} \tag{46}$$

$$\min z_2^{(a)} = \sum_{i=1}^{I_A} d_i^{2+} \tag{47}$$

The objective (achievement) function of the weighted goal program is then given by

$$\min z_w = w_1^{(w)} \left(\sum_{i=1}^{I_A} d_i^{1-} - d_i^{1+} \right) + w_2^{(w)} \left(\sum_{i=1}^{I_A} d_i^{2+} \right), \tag{48}$$

where $w_1^{(w)}, w_2^{(w)}$ are the weights representing the relative importance of the corresponding goals, whereas for a particular lexicographic importance of goals, the objective (achievement) function of the lexicographic (pre-emptive) goal program is then written as

$$\text{lex min } z_p = \left\{ \begin{array}{c} \left(\sum_{i=1}^{I_A} d_i^{1-} - d_i^{1+} \right) \\ \left(\sum_{i=1}^{I_A} d_i^{2+} \right) \end{array} \right\} \tag{49}$$

or equivalently as

$$\min z_p = w_1^{(p)} \left(\sum_{i=1}^{I_A} d_i^{1-} - d_i^{1+} \right) + w_2^{(p)} \left(\sum_{i=1}^{I_A} d_i^{2+} \right), \tag{50}$$

where $w_1^{(p)} \gg w_2^{(p)}$ are the weights representing the lexicographic importance (priorities) of the corresponding goals.

5. Implementation and results

In the following two subsections, we summarise the implementation details and present the results for the first and second phases using three different real-life data-sets. We implement our formulations on a personal computer (with AMD Turion II P520 Dual-Core Processor (2.30 GHz) and 4.00 GB RAM) using Mathematical Programming Language (MPL) and Gurobi Optimiser. In each phase, our results are compared with the real-life allocations in terms of solution quality with respect to the problem criteria (i.e., the goals presented in the previous section). Note that, to quantify the first goal in each problem, the ones about satisfying the preferences of the students and advisers, we use a performance measure term computed as the averages of $\sum_j \sum_k k x_{ijk}^{(s)}$ for all students and $\sum_j \sum_k k x_{ijk}^{(a)}$ for all advisers; a smaller number implying a better preference satisfaction for both cases.

In both phases, in addition to the cases where all goals are of equal importance, we also use different settings to see the effects of changing weights on the optimal solution. To maintain the current length of our paper, we only consider a few different settings instead of performing a comprehensive sensitivity analysis. We additionally pre-empt each problem criterion for both validating the weight settings and observing the effects of pre-empting a particular problem criterion in both cases. We thus construct the corresponding *pre-emptive* settings where we solve a mathematical program with a single objective to minimise the dominating criterion. We finally note the different structures of the satisfaction scores for the preferences criteria in both phases as explained in the problem context, as a result of which it is more difficult for the decision maker to determine the importance of student or adviser preferences relative to the other problem criteria. We thus perform our final computational experiments considering the problematic issue about the preferences criteria in both phases. To address the issue, we combine both the weighted (non-pre-emptive) and lexicographic (pre-emptive) goal programming variants together to assign extremely more and extremely less importance to the preferences criteria. We thus construct the corresponding *pre-emptive* settings where we sequentially solve two programs each with a single objective to minimise the dominating criterion and the other dominated criteria, respectively. While solving the program for the second (pre-empted) objective (for the dominated criterion (criteria)), the optimal value of the first (pre-empting) objective (for the dominating criterion (criteria)) is included as a new constraint. Note that a small relaxation for the new constraint might be

necessary in some cases to prevent potential rounding errors. We use such relaxations only in cases where the optimal value of the first (pre-empting) objective (for the dominating criterion (criteria)) is non-integer (fractional), and thus, yields a non-integer (fractional) right-hand-side in the new constraint.

5.1. Phase I: Student allocations

The results are given in Table 2 for student allocations where we simply assume all goals are of equal importance. A simple normalisation procedure is used for scaling the deviations to represent all of them approximately between 0 and 1. Note that the normalisation procedure might be more difficult in some cases, such as the normalisation of a scale-free deviation. In our case, it seems to require a different consideration especially for the student preferences criterion. It is more difficult to put appropriate lower-upper bounds for the corresponding deviations since some random variations on these are observed for different data-sets and even for different people. We nevertheless consider a normalisation procedure for the corresponding deviations in the objective function by dividing them by the maximum of such deviations by evaluating the data-sets used in our study. We moreover utilise another goal program that combines the weighted and lexicographic variants to address the problematic issue about the student preferences.

Except the first two goals, the satisfaction scores of the goals are given as percentages, a higher number corresponds to a better solution. A smaller number implies a better preference satisfaction for the first goal. For the second goal, the deviations in the number of students in the teams are represented with three numbers as in Table 2. The number in parentheses shows the number of teams with desired number of students. The number to left (right) of the number in parentheses is the number of teams with one less (more) than the desired number of students. We summarise the performance of the proposed approach in terms of the goals of the first phase (student allocations) as follows:

(1) Preferences of the students should be satisfied as much as possible: It is noted that the proposed approach produces better results than the real-life allocations in all cases.
(2) Each team should have N_S students, approximate-average number of students for each team: It is noted that the proposed approach produces again better results than the real-life allocations in all cases.
(3) Average GPA of each team should be G_T, approximate-average GPA of the class: It is noted that the proposed approach produces again better results than the real-life allocations in all cases.
(4) Average language grade in each team should be at least L_T above-average grade: Both solutions satisfy the goal perfectly for all cases.
(5) Each team should have at least one student with computer programming skills: It is noted that the proposed approach produces again better results than the real-life allocations in all cases.
(6) Each team should have at least one student with general software skills: The proposed approach satisfies the goal perfectly for all cases. Note that real-life allocations also satisfy the goal perfectly for the first and third data-sets.
(7) Each team should have at least one student with database skills: It is noted that the proposed approach produces again better results than the real-life allocations in all cases.

Considering the superior performance of the proposed approach, we also note a few infeasible solutions in the real-life allocations. In particular, it is noted that the real-life allocations do not satisfy the hard constraints about the GPA in any case. In other words, although the real-life allocations produce seemingly satisfactory results, these are achieved by constructing some teams whose average GPA are below (above) the allowed lower (upper) bounds.

In addition to the case where all goals are of equal importance, we also use different settings to see the effects of changing weights on the optimal solution. To maintain a reasonable length for our paper, we only consider three different settings in addition to the solution given in Table 2 instead of performing a comprehensive sensitivity analysis. The first setting corresponds to the solution given in Table 2 (i.e., equal importance of each goal) and is presented for the convenience of the reader, whereas each of the remaining settings represents the dominance of a particular problem criterion. In particular, these settings represent the (i) equal importance of each goal (as in Table 2), and the dominance of the problem criterion for (ii) student preferences, (iii) number of students per team and (iv) student qualifications (GPA, language skills etc.). The results are summarised in Table 3. To dominate some particular problem criterion, we significantly increase the weight of the corresponding goal. Note that the importance of student preferences and number of students per team criteria are represented with the weights of the first two goals w_1 and w_2, respectively, whereas the importance of the remaining criterion (i.e., student qualifications) are determined with the respective weights of the remaining five goals from w_3 to w_7. We can thus define these settings as follows:

(1) In this setting, each criterion has equal importance, that is, $w_1 = w_2 = w_q$, where $q = 3, \ldots, 7$. represents the index of student qualification goals.

Table 2. Summary of results (student allocations).

Goal	Data-set-1		Data-set-2		Data-set-3	
	RLA-1	PS-1	RLA-2	PS-2	RLA-3	PS-3
1	2.58	1.85	2.18	1.52	2.44	2.03
2	2-(10)-11	0-(14)-9	5-(12)-1	4-(14)-0	6-(18)-1	5-(20)-0
3	95%	97%	95%	96%	94%	98%
4	100%	100%	100%	100%	100%	100%
5	87%	100%	61%	89%	68%	100%
6	100%	100%	89%	100%	100%	100%
7	57%	74%	28%	33%	36%	52%
ST (s)	-	1.92	-	1.31	-	20.24

Abbreviations: RLA: Real Life Allocations, PS: Proposed Solution, ST: Solution Time (seconds).

Table 3. Solutions for different settings (weighted GP).

Goal	Setting-1	Setting-2	Setting-3	Setting-4
Data-set-1				
1	1.69	1.60	1.78	1.82
2	2-(10)-11	6-(2)-15	0-(14)-9	2-(10)-11
3	96%	95%	97%	98%
4	100%	100%	100%	100%
5	91%	83%	91%	100%
6	100%	100%	100%	100%
7	70%	61%	74%	74%
Data-set-2				
1	1.32	1.24	1.32	1.56
2	4-(14)-0	7-(8)-3	4-(14)-0	4-(14)-0
3	96%	96%	96%	98%
4	100%	100%	100%	100%
5	67%	61%	67%	89%
6	100%	83%	100%	100%
7	33%	33%	33%	33%
Data-set-3				
1	1.74	1.64	1.87	1 91
2	9-(12)-4	12-(6)-7	5-(20)-0	8-(14)-3
3	97%	96%	97%	98%
4	100%	100%	100%	100%
5	84%	72%	80%	100%
6	100%	100%	100%	100%
7	52%	48%	48%	52%

(2) In this setting, student preference criterion is significantly more important than the other criteria and each other criterion has equal importance, that is, $w_1 > w_2 = w_q$, where $q = 3, \ldots, 7$. represents the index of student qualification goals.
(3) In this setting, number of students per team criterion is significantly more important than the other criteria and each other criterion has equal importance, that is, $w_2 > w_1 = w_q$, where $q = 3, \ldots, 7$. represents the index of student qualification goals.
(4) In this setting, student qualifications criterion is significantly more important than the other criteria and each other criterion has equal importance, that is, $w_q > w_1 = w_2$, where $q = 3, \ldots, 7$. represents the index of student qualification goals.

The results presented in Table 3 show the satisfaction scores of the goals for each data-set as in Table 2 for each setting. As noted before, the first setting (Setting-1) represents the equal importance of each goal as given in Table 2, and is included for the convenience of the reader. We note that the setting where the student preference criterion (Setting-2) dominates the other criteria produces the best satisfaction scores for the corresponding criterion for all data-sets as expected. Other satisfaction scores of the setting, on the other hand, show that improving student preferences comes at a price. It seems that other satisfaction scores usually are at their worsts in this setting which is not the case for the other settings where a particular criterion dominates the others. In Setting-3, for instance, although the satisfaction scores for the number of students per team criterion are at their bests, it seems that we can still get better results than those of Setting-2 for the student qualifications criterion. Setting-4 produces the best results for the student qualifications criterion as expected. It seems, on the other hand, that constructing the best-qualified student teams yields the worst allocations in terms of satisfying the student preferences.

We also pre-empt each problem criterion for both validating the weight settings and observing the effects of pre-empting a particular problem criterion. We thus construct the corresponding *pre-emptive* settings where we solve a mathematical program with a single objective to minimise the dominating criterion. It turns out from the results that we cannot improve the student preference and number of students criteria more than the corresponding satisfaction scores given in Table 3. In terms of student qualifications criterion, it is observed that we can only improve the satisfaction score of a single goal (GPA) in the first and second data-sets by 1% and 2%, respectively.

We finally note the different structures of the satisfaction scores for the student preferences criterion as explained in the problem context, as a result of which it is more difficult for the decision maker to determine the importance of student preferences relative to the other problem criteria. We thus perform our final computational experiments considering the problematic issue about the preferences criteria in both phases. To address the issue, we utilise the lexicographic (pre-emptive) goal programming variant to assign extremely more and extremely less importance to the preferences criterion. We thus construct the corresponding *pre-emptive* settings where we sequentially solve two programs each with a single objective to minimise the satisfaction scores of the dominating criterion and the

other dominated criteria, respectively. We thus define the following six settings as:

(1) In this setting, student preference criterion is extremely more important than the other criteria and each other criterion has equal importance, that is, $w_1 \gg w_2 = w_q$, where $q = 3, \ldots, 7$. represents the index of student qualification goals.
(2) In this setting, student preference criterion is extremely more important than the other criteria and number of students criterion is more important than the student qualifications criterion, that is, $w_1 \gg w_2 > w_q$, where $q = 3, \ldots, 7$. represents the index of student qualification goals.
(3) In this setting, student preference criterion is extremely more important than the other criteria and number of students criterion is less important than the student qualifications criterion, that is, $w_1 \gg w_q > w_2$, where $q = 3, \ldots, 7$. represents the index of student qualification goals.
(4) In this setting, student preference criterion is extremely less important than the other criteria and each other criterion has equal importance, that is, $w_2 = w_q \gg w_1$, where $q = 3, \ldots, 7$. represents the index of student qualification goals.
(5) In this setting, student preference criterion is extremely less important than the other criteria and number of students criterion is more important than the student qualifications criterion, that is, $w_2 > w_q \gg w_1$, where $q = 3, \ldots, 7$. represents the index of student qualification goals.
(6) In this setting, student preference criterion is extremely less important than the other criteria and number of students criterion is less important than the student qualifications criterion, that is, $w_q > w_2 \gg w_1$, where $q = 3, \ldots, 7$. represents the index of student qualification goals.

The results presented in Table 4 show the satisfaction scores of the goals for each data-set as in Table 2 and 3 for each setting. We first note that the best possible satisfaction scores for student preferences yield worse satisfaction scores for the other problem criteria as expected. It is noted that the results of the first three settings are exactly the same in the first data-set which are also produced by the weighted goal program where the student preferences criterion is more important than the other criteria as presented in Table 4 (i.e., Setting-2, that is, $w_1 > w_2 = w_q$, where $q = 3, \ldots, 7$. represents the index of student qualification goals). In the other data-sets (Data-set-2 and Data-set-3), on the other hand, changing the relative importance of other criteria produces different results in favour of the number of students per team and student qualifications criterion in the corresponding setting (i.e., Setting-2 and Setting-3), respectively.

Table 4. Solutions for different settings (GP with lexicographic preferences).

Goal	Setting-1	Setting-2	Setting-3	Setting-4	Setting-5	Setting-6
Data-set-1						
1	1.60	1.60	1.60	1.88	1.88	1.97
2	6-(2)-15	6-(2)-15	6-(2)-15	0-(14)-9	0-(14)-9	0-(14)-9
3	95%	95%	95%	98%	98%	99%
4	100%	100%	100%	100%	100%	100%
5	83%	83%	83%	100%	100%	100%
6	100%	100%	100%	100%	100%	100%
7	61%	61%	61%	74%	74%	74%
Data-set-2						
1	1.24	1.24	1.24	1.54	1.54	1.90
2	7-(8)-3	7-(8)-3	9-(4)-5	4-(14)-0	4-(14)-0	4-(14)-0
3	96%	96%	96%	97%	97%	99%
4	100%	100%	100%	100%	100%	100%
5	61%	61%	67%	89%	89%	89%
6	83%	83%	94%	100%	100%	100%
7	33%	33%	33%	33%	33%	33%
Data-set-3						
1	1.64	1.64	1.64	2.06	2.06	2.11
2	12-(6)-7	12-(6)-7	13-(4)-8	5-(20)-0	5-(20)-0	5-(20)-0
3	96%	96%	96%	98%	98%	98%
4	100%	100%	100%	100%	100%	100%
5	72%	76%	76%	100%	100%	100%
6	100%	100%	100%	100%	100%	100%
7	48%	44%	44%	52%	52%	52%

The satisfaction scores of the student preferences criterion are usually worse than those produced by the setting where the student preferences criterion has the lowest importance (i.e., Setting-4) of the weighted goal program. Setting-4 and Setting-5 in Data-set-2 represent the exceptional cases where the satisfaction scores of the student preferences criterion (i.e., 1.54 in both settings) are slightly better than those of Setting-4 of the weighted goal program (i.e., 1.56).

The worst satisfaction scores overall, on the other hand, are always produced by the last setting (Setting-6) of the goal program with lexicographic variant as noted in Table 4. We moreover note that changing the relative importance of other criteria (number of students per team and student qualifications) does not affect the satisfaction score of the number of students per team criterion at all in the last three settings. It however slightly improves the corresponding scores for the student qualifications in the last setting where the relative importance of the student qualifications criterion is higher than that of the number of students per team criterion.

When compared to the results of the weighted goal program, it is noted that the satisfaction scores for the number of students per team criterion in the last three settings of the goal program with lexicographic variant are exactly the same as the best satisfaction scores for the corresponding criterion in the weighted goal program whereas a few slight improvements over the weighted variant are observed in the final setting (Setting-6) in terms of student qualifications.

Table 5. Summary of results (adviser allocations).

	Data-set-1		Data-set-2		Data-set-3	
Goal	RLA-1	PS-1	RLA-2	PS-2	RLA-3	PS-3
1	3.78	2.65	1.33	1.14	3.74	2.94
2	0-(9)-2	0-(9)-2	0-(8)-3	0-(8)-3	1-(8)-3	0-(10)-2
ST (s)	-	0.19	-	0.06	-	0.09

Abbreviations: RLA: Real Life Allocations, PS: Proposed Solution, ST: Solution Time (seconds).

5.2. Phase II: Adviser allocations

The results are given in Table 5 for adviser allocations where we assume all goals are of equal importance as in the first phase. As in the student allocations, a simple normalisation procedure is used for scaling the deviations to represent all of them approximately between 0 and 1. Again note that, as for the student preferences criterion, the normalisation procedure seems to require a different consideration for the adviser preferences criterion. We consider a similar normalisation procedure as in the student allocations. Corresponding deviations in the objective function are divided by the maximum of such deviations by evaluating the data-sets used in our study for normalisation. We moreover utilise another goal program that combines the weighted and lexicographic variants to address the problematic issue about the adviser preferences as in the student allocations.

For the first goal, just as in the student allocations, a smaller number implies a better result. For the second goal, on the other hand, deviations from the desired number of teams an adviser administrates (i.e., deviations from N_T) should be minimised. We also summarise the performance of the proposed approach in terms of the goals of the second phase (adviser allocations) as follows:

(1) Preferences of the advisers should be satisfied as much as possible: As in the student preferences, the proposed approach again produces better results than the real-life allocations for all cases.
(2) Each adviser should have N_T project teams, that is the average number of project teams per adviser: We note that, the proposed approach produces a better result for the third case whereas for the first and second cases, both solutions satisfy the goal equally.

As in the student allocations, in addition to the case where all goals are of equal importance, we also use different settings to see the effects of changing weights on the optimal solution. Also as in the student allocations, to maintain a reasonable length for our paper, we only consider two different weight settings in addition to the solution given in Table 5 instead of performing a comprehensive sensitivity analysis. The first setting corresponds to the solution given in Table 5 (i.e., equal importance of each goal), and is presented for the convenience of the reader, whereas each of the remaining settings represents the dominance of a particular problem criterion. In particular, these settings represent the (i) equal importance of each goal (as in Table 2), and the dominance of the problem criterion for (ii) adviser preferences and (iii) number of teams per adviser. The results are summarised in Table 6. To dominate some particular problem criterion, we significantly increase the weight of the corresponding goal. Note that the importance of adviser preferences and number of teams per adviser criteria are represented with the weights of the two goals w_1 and w_2, respectively. We can thus define these settings as follows:

(1) In this setting, each criterion has equal importance, that is, $w_1 = w_2$.
(2) In this setting, the adviser preferences criterion is significantly more important than the number of teams per adviser criterion, that is, $w_1 > w_2$.
(3) In this setting, the number of teams per adviser criterion is significantly more important than the adviser preferences criterion, that is, $w_1 < w_2$.

The results presented in Table 6 show the satisfaction scores of the goals for each data-set as in Table 5 for each setting. As noted before, the first setting (Setting-1) represents the equal importance of each goal as given in Table 5, and is included for the convenience of the reader. Although some improvements are observed in the first two data-sets for the first (adviser preference) criterion, it is noted from the results that our simple sensitivity analysis also illustrates the highly constrained structure of the feasible space of the problem. It is especially observed for the second (number of student teams per adviser) criterion in all data-sets and for all criteria in the third data-set. In fact, the third data-set represents an extreme case where employing any of the defined settings does not affect the optimal solution. Note that the corresponding highly constrained structure of the feasible space is caused by the hard constraints on the number of student teams per adviser given by Equation 40. These constraints guarantee some level of fairness among the advisers by defining some lower and upper bounds for the number of student teams per adviser. Similar cases are partially observed for the first two data-sets. It is noted, in these data-sets, that the dominance of the second (number of student teams per adviser) criterion does not affect the optimal solution, whereas we can improve the first (adviser preference) one if it becomes the dominating criterion. We finally note that being unable to improve the second (number of student teams per adviser) criterion is an expected for the data-sets available since the first setting already produces the best satisfaction score in terms of the number of student teams per adviser criterion. In other words, it is noted that since the total number of advisers is not a divisor of the total number of allocations (the total number of projects multiplied by the number of

Table 6. Solutions for different settings (weighted GP).

Goal	Setting-1	Setting-2	Setting-3
Data-set-1			
1	2.65	2.54	2.65
2	0-(9)-2	2-(5)-4	0-(9)-2
Data-set-2			
1	1.14	1.08	1.14
2	0-(8)-3	1-(6)-4	0-(8)-3
Data-set-3			
Goal	Setting-1	Setting-2	Setting-3
1	2.94	2.94	2.94
2	0-(10)-2	0-(10)-2	0-(10)-2

Table 7. Solutions for different settings (GP with lexicographic preferences).

Goal	Setting-1	Setting-2
Data-set-1		
1	2.54	2.65
2	2-(5)-4	0-(9)-2
Data-set-2		
1	1.08	1.14
2	1-(6)-4	0-(8)-3
Data-set-3		
1	2.94	2.94
2	0-(10)-2	0-(10)-2

advisers per project), some advisers must administrate as many additional projects as the remainders of the division (i.e., 2, 3 and 2 additional projects in the three data-sets, respectively).

We also pre-empt each problem criterion for both validating the weight settings and observing the effects of pre-empting a particular problem criterion. We thus construct the corresponding *pre-emptive* settings where we solve a mathematical program with a single objective to minimise the dominating criterion. It turns out from the results that we cannot improve the satisfaction score of any criterion more than the corresponding satisfaction scores given in Table 6.

As in the student allocations, to note the difficulty for the decision maker to determine the importance of adviser preferences relative to the other problem criterion (i.e., number of teams per adviser), we perform our final computational experiments considering the problematic issue about the adviser preferences. To address the issue, we utilise the lexicographic (pre-emptive) goal programming variants to assign extremely more and extremely less importance to the preferences criterion. We thus construct the corresponding *pre-emptive* settings where we sequentially solve two programs each with a single objective to minimise the satisfaction scores of the dominating criterion and the other dominated criteria, respectively. While solving the program for the second (pre-empted) objective (for the dominated criterion (criteria)), the optimal value of the first (pre-empting) objective (for the dominating criterion (criteria)) is included as a new constraint. We thus define the following two settings as:

(1) In this setting, the adviser preferences criterion is extremely more important than the number of teams per adviser criterion, that is, $w_1 \gg w_2$.
(2) In this setting, the adviser preferences criterion is extremely less important than the number of teams per adviser criterion, that is, $w_1 \ll w_2$.

The results presented in Table 7 show the satisfaction scores of the goals for each data-set as in Table 5 and 6 for each setting. It is noted that the satisfaction scores produced by Setting-1 and Setting-2 of the goal program with lexicographic variant are exactly the same as the satisfaction scores of those produced by the Setting-2 and Setting-3 of the weighted goal program. In other words, pre-empting a particular goal does not improve the corresponding satisfaction scores more than those produced by the corresponding setting of the weighted goal program, implying the aforementioned highly constrained structure of the feasible space in the adviser allocations.

6. Solution efficiency

Since the main philosophy of goal programming is about the achievement of a set of target levels as closely as possible (i.e., satisfying), a solutions produced by a satisfying decision maker may be Pareto inefficient; that is, the value of one or more of the objectives can be improved without worsening any others. Such a situation also indicates the pessimistic viewpoint of the decision maker in setting the target levels. Although we do not present an analysis on the efficiency issue in the following pages to maintain the current length of our paper; for the sake of completeness, in this section, we present a brief discussion about the efficiency of solutions, detection and restoration of Pareto efficiency adapted from Jones and Tamiz (2010), intended as a guideline for future studies. The interested reader might refer to a more detailed discussion presented by Jones and Tamiz (2010).

We first note the necessity of different treatments for Pareto efficiency detection for different goal programs. Detection of Pareto efficiency, for instance, continuous-weighted and lexicographic goal programming variants is different from those of integer (binary)-goal programming models. In continuous-weighted and lexicographic goal programming variants, there are several tests presented by Jones and Tamiz (2010) for detection of Pareto efficiency in many cases.

The first test (Test A) states that two-sided goals can be classified as Pareto efficient. If we represent the penalised and not-penalised deviational variables with α and β, respectively; we can say that any objective with a basic non-zero α at optimal solution is Pareto efficient (Test B) whereas any objective with a non-basic β which can be entered into the basis without lowering the values of any basic β or increasing the value of any

basic α is Pareto inefficient (Test C). Similarly, another test (Test D) for detection indicates that for any objective with a non-basic β, if that β can be increased by entering of any variable to the basis without causing an increase of another objective's α or decrease in another objective's β, then, that objective is Pareto inefficient.

Although we can detect Pareto efficiency of many objectives using these tests without performing any simplex iterations, another test (Test E) detailed by Jones and Tamiz (2010) can be used for the other unclassified objectives as an alternative which might require the consideration of computational issues.

We note that although tests A and B can still be used for integer or binary goal programs (as in our case), tests C and D are not applicable anymore whereas a modified version of test E proposed by Tamiz, Mirrazavi and Jones (1999) can be applied instead. It classifies the objectives of any integer or binary goal program into Pareto-inefficient and Pareto-efficient subsets. It is summarised as follows for objective t.

Step (1): Impose the following bounds.

$$\alpha_q \leq \alpha_q^*, \quad q = 1, \ldots, Q \tag{51}$$

$$\beta_q \geq \beta_q^*, \quad q = 1, \ldots, Q, q \neq t \tag{52}$$

where α_q^* and β_q^* represent the values of the deviational variables of the objective q in the integer optimal solution.

Step (2): Solve the following integer program starting with the solution of the original integer program.

$$\max z = \sum_{q \in Q^u} \beta_q \tag{53}$$

subject to

$$g_q(x) + n_q - p_q = b_q, \quad q = 1, \ldots, Q \tag{54}$$

$$\alpha_q \leq \alpha_q^*, \quad q = 1, \ldots, Q \tag{55}$$

$$\beta_q \geq \beta_q^*, \quad q = 1, \ldots, Q \tag{56}$$

$$x \in F \tag{57}$$

$$n_q, p_q \geq 0, \quad q = 1, \ldots, Q \tag{58}$$

where Q^u is the set of not-yet classified objectives.

Step (3): Inspect the optimal solution of the integer program and classify any objectives with $\beta_q \geq \beta_q^*$ as Pareto inefficient. If no such objectives exist, then, classify all remaining unclassified objectives as Pareto efficient and terminate the algorithm. Otherwise update set Q^u and go to Step (2).

In our case, note that we have seven and three objectives in the proposed integer (binary)-goal programs of the first- and second-phase problems, respectively. Among aforementioned tests, we can thus use test A and B as well as the modified version of test E for Pareto efficiency detection of these objectives. We also note the applicability of test A for the two-sided objectives; the second and third objectives of the first-phase problem and the first objective of the second-phase problem. The other applicable tests (test B and modified version of test E) can be considered for the remaining objectives. In case of a Pareto inefficiency, some methods are proposed to restore Pareto efficiency, such as,

(1) Straight Restoration
(2) Preference-Based Restoration
(3) Interactive Restoration

We note that, among these, straight restoration method's assumptions, stating that each β is desired not only to be maximised but also with equal importance, can be considered as its drawbacks. Also note that although the usage of preference-based restoration might be a preferred way of restoring Pareto efficiency without interfering with the decision maker, it has a disadvantage of assuming the same below and above target level preferences for the decision maker. The interactive restoration method, however, requires the decision maker's involvement to make a choice from the set of inefficient objectives at each step of an iterative process. We do not present more details on these approaches to maintain the current length of our paper by referring the interested reader to Tamiz and Jones (1996), Tamiz et al. (1998) and Jones and Tamiz (2010) for more detailed discussions on these restoration approaches.

As a final remark, we note different considerations are required for detection and restoration of Pareto efficiency for Chebyshev and non-linear goal programs. In a Chebysev goal program, balancing, as another underlying philosophy of goal programming, must be taken into account whereas some other special approaches are proposed for non-linear goal programs such as the ones presented by Caballero et al. (1998) and Caballero and Hernandez (2006).

7. Conclusions

In this study, we present a two-phase binary-goal programming-based approach for solving a novel system design project-team formation problem which involves several restrictions and requirements as well as the preferences of the potential team members. The problem instance considered in this study basically includes two types of allocations, as the allocations of students and academic advisers, which are performed in the first and second phase of the proposed solution approach, respectively. Although it represents a

particular case as represented in this study, it can be easily generalised to be used for solving similar project-team formation problems. We implement our model on a real-life project-team formation problem in an academic institution and compare our solutions to the real-life allocations performed manually. It is noted that, in terms of satisfying the goals of the problem, our approach significantly outperforms the real-life allocation procedure. In addition, computational results show our model's ability to solve similar-sized real-life problems considered in this study in reasonable time periods (i.e., less than a minute) on an average personal computer, implying its potential for significant savings in terms of the human resources available. In terms of its scientific contributions, our study not only defines and conceptualises the project-team formation problem (which combines some aspects of different problems in the literature, i.e., team formation problems, skilled-workforce planning and scheduling problems etc.) by considering all real-life problem parameters, but also presents an approach to optimally solve the similar-size problem instances in reasonable time periods.

Several extensions to our approach are planned in future studies. First of all, the process for the identification of student qualifications can be improved since, currently, a few simple quantities are used for this purpose. Considering a student's educational background in more detail (i.e., student's grades in particular classes) and identifying the student's qualifications based on this can produce more accurate results. Consideration of the areas of expertise of advisers is another future study direction to allocate advisers accordingly. Since, in addition to academic advisers, project teams might also include advisers from the organisations where the projects are implemented, we think, it is important to consider the restrictions, requirements and preferences of these organisations as well as the advisers from the organisations. We also want to take social relationships among students and advisers into account in our future studies, probably one of the most important factors to form successful project teams.

In terms of computational issues, although the proposed solution approach's performance is quite successful for the real-life problem considered in this study, we can randomly generate larger-size problem instances to analyse its computational performance. Developing alternative solution approaches might be considered as another future study direction.

One of the shortcomings of our study is the lack of the feedback from the students and advisers on the allocations. Although currently it is not possible to obtain such feedback about the proposed project-team formation approach since it is not implemented in practice, we can obtain some feedback from the students and advisers about the real-life allocations. The results can also be used to determine the relative importance of the problem criteria since we currently use the equal importance of problem criteria assumption of the decision maker in this study. Although it is a reasonable assumption, we think that a more comprehensive weight determination approach in our goal program might be very useful. We plan to use some group decision-making technique to determine the corresponding weights to be able to consider the perspective of a larger group of people in the decision making process.

Disclosure statement

No potential conflict of interest was reported by the authors.

ORCID

Fatih Cavdur http://orcid.org/0000-0001-8054-5606
Asli Sebatli http://orcid.org/0000-0002-9445-6740
Merve Kose-Kucuk http://orcid.org/0000-0001-6877-2937

Supplementary material

The underlying research materials for this article can be accessed at http://fatihcavdur.home.uludag.edu.tr/.

References

Abraham, D. J., Irwing, R. W., & Manlove, D. F. (2007). Two algorithms for the student-project allocation problem. *Journal of Discrete Algorithms, 5*(1), 73–90.

Agustin-Blas, L. E., Salcedo-Sans, S., Ortiz-Garcia, E. G., Portilla-Figueras, A., Perez-Bellido, A. M., & Jimenez-Fernandez, S. (2011). Team formation based on group technology: A hybrid grouping genetic algorithm approach. *Computers and Operations Research, 38*, 484–495.

Anwar, A. A., & Bahaj, A. S. (2003). Student project allocation using integer programming. *IEEE Transactions on Education, 46*(3), 359–367.

Aouni, B., & Kettani, O. (2001). Goal programming model: A glorious history and a promising future. *European Journal of Operational Research, 133*(2), 225–231.

Azaiez, M. N., & Al Sharif, S. S. (2005). A 0–1 goal programming model for nurse scheduling. *Computers and Operations Research, 32*(3), 491–507.

Bacon, D. R., Stewart, K. A., & Anderson, E. S. (2001). Methods of assigning players to teams: A review and novel approach. *Simulation and Gaming, 32*(1), 6–17.

Badri, M. A., Davis, D., & Davis, D. (2001). A comprehensive 0–1 goal programming model for project selection. *International Journal of Project Management, 19*(4), 243–252.

Baykasoglu, A., Dereli, T., & Das, S. (2007). Project team selection using fuzzy optimization approach. *Cybernetics and Systems: An International Journal, 38*(2), 155–185.

Bengtsson, L., Galia, R., Gustafsson, T., Hjorring, C. & Kohl, N. (2007). Railway crew pairing optimization. *Algorithmic Methods for Railway Optimization*, 126–144.

Blowers, P. (2003). Using student skill self-assessments to get balanced groups for group projects. *College Teaching, 51*(3), 106–110.

Borrego, M., Karlin, J., McNair, L. D., & Beddoes, K. (2013). Team effectiveness theory from industrial and organizational psychology applied to engineering student project teams: A research review. *Journal of Engineering Education, 102*(4), 472–512.

Caballero, R., & Hernandez, M. (2006). Restoration of efficiency in a goal programming problem with linear fractional criteria. *European Journal of Operational Research, 172*(2006), 31–39.

Caballero, R., Ray, L., & Ruiz, F. (1998). Lexicographic improvement of the target values in convex goal programming. *European Journal of Operational Research, 107*(1998), 644–655.

Chang, C. T. (2007a). Binary fuzzy goal programming. *European Journal of Operational Research, 180*(1), 29–37.

Chang, C. T. (2007b). Multi-choice goal programming. *Omega, 35*(4), 389–396.

Chang, C. T. (2008). Revised multi-choice goal programming. *Applied Mathematical Modelling, 32*(12), 2587–2595.

Chang, C. T. (2011). Multi-choice goal programming with utility functions. *European Journal of Operational Research, 215*(2), 439–445.

Charnes, A., & Cooper, W. W. (1957). Management models and industrial applications of linear programming. *Management Science, 4*(1), 38–91.

Charnes, A., Cooper, W. W., & Ferguson, R. (1955). Optimal estimation of executive compensation by linear programming. *Management Science, 1*, 138–151.

Cheang, B., Li, C. H., Lim, A., & Rodrigues, B. (2003). Nurse rostering problems – A bibliographic survey. *European Journal of Operational Research, 151*(3), 447–460.

Ernst, A. T., Jiang, H., Krishnamoorthy, M., & Sier, D. (2004). Staff scheduling and rostering: A review of applications, methods and models. *European Journal of Operational Research, 153*(1), 3–27.

Gale, D., & Shapley, L. S. (1962). College admissions and the stability of marriage. *American Mathematical Monthly, 69*, 9–15.

Gusfield, D., & Irwing, R. W. (1989). *The stable marriage problem: Structure and algorithms*. Boston, MA: MIT Press.

Hubscher, R. (2010). Assigning students to groups using general and context-specific criteria. *IEEE Transactions on Learning Technologies, 3*(3), 178–189.

Huxham, M., & Land, R. (2000). Assigning students in group work projects. Can we do better than random? *Innovations in Education and Training International, 37*(1), 17–22.

Ignizio, J. P. (1978). A review of goal programming: A tool for multiobjective analysis. *Journal of the Operational Research Society, 29*(11), 1109–1119.

Iwama, K., Miyazaki, S., & Yanagisawa, H. (2012). Improved approximation bounds for the student-project allocation problem with preferences over projects. *Journal of Discrete Algorithms, 13*, 59–66.

Jones, D., & Tamiz, M. (2010). *Practical goal programming*. New York, NY: Springer.

Liao, C. N. (2009). Formulating the multi-segment goal programming. *Computers and Industrial Engineering, 56*(1), 138–141.

Lin, H. W., Nagalingam, S. V., & Lin, G. C. I. (2009). An interactive meta-goal programming-based decision analysis methodology to support collaborative manufacturing. *Robotics and Computer-Integrated Manufacturing, 25*(1), 135–154.

Manlove, D. F., & O'Malley, G. (2008). Student-project allocation with preferences over projects. *Journal of Discrete Algorithms, 6*, 553–560.

Meskens, N., Duvivier, D., & Hanset, A. (2013). Multi-objective operating room scheduling considering desiderata of the surgical team. *Decision Support Systems, 55*(2), 650–659.

Oakley, B., Felder, R. M., Brent, R., & Elhajj, I. (2004). Turning student groups into effective teams. *Journal of Student Centered Learning, 2*(1), 9–34.

Proll, L. G. (1972). A simple method of assigning projects to students. *Operational Research Quarterly, 23*(2), 195–201.

Romero, C. (2001). Extended lexicographic goal programming: A unifying approach. *Omega, 29*(1), 63–71.

Tamiz, M., & Jones, D. F. (1996). Goal programming and Pareto efficiency. *Journal of Information and Optimization Sciences, 17*(2), 291–307.

Tamiz, M., Jones, D. F., & Romero, C. (1998). Goal programming for decision making: An overview of the current state-of-the-art. *European Journal of Operational Research, 111*(3), 569–581.

Tamiz, M., Mirrazavi, S. K., & Jones, D. F. (1999). Extensions of Pareto efficiency analysis to integer goal programming. *Omega, 27*(2), 178–188.

Tseng, T. L. B., Huang, C. C., Chu, H. W., & Gung, R. R. (2004). Novel approach to multi-functional project team formation. *International Journal of Project Management, 22*(2), 147–159.

Uria, M. V. R., Caballero, R., Ruiz, F., & Romero, C. (2002). Meta-goal programming. *European Journal of Operational Research, 136*(2), 422–429.

Van den Bergh, J., Belien, J., De Bruecker, P., Demeulemeester, E., & De Boeck, L. (2013). Personnel scheduling: A literature review. *European Journal of Operational Research, 26*(3), 367–385.

Wi, H., Oh, S., Mun, J., & Jung, M. (2009a). A team formation model based on knowledge and collaboration. *Expert Systems with Applications, 36*(5), 9121–9134.

Wi, H., Mun, J., Oh, S., & Jung, M. (2009b). Modeling and analysis of project team formation factors in a project-oriented virtual organization (ProVO). *Expert Systems with Applications, 36*(3), 5775–5783.

Yan, S., & Chang, J. C. (2002). Airline cockpit crew scheduling. *European Journal of Operational Research, 136*(3), 501–511.