EDUCATION FOR CHEMICAL ENGINEERS 20 (2017) 11–21

Contents lists available at ScienceDirect

Education for Chemical Engineers

journal homepage: www.elsevier.com/locate/ece

Mathematical programming approach for optimally allocating students' projects to academics in large cohorts

Raúl Calvo-Serrano [a,b], *Gonzalo Guillén-Gosálbez* [a,c,*], *Simon Kohn* [a], *Andrew Masters* [a]

[a] *Centre for Process Systems Engineering (CPSE), Imperial College London, SW7 2AZ, United Kingdom*
[b] *School of Chemical Engineering and Analytical Science, The University of Manchester, Oxford Road, Manchester M13 9PL, United Kingdom*
[c] *Departament d'Enginyeria Química, Universitat Rovira i Virgili, Av.Països Catalans, 26, 43007 Tarragona, Spain*

ARTICLE INFO

Article history:
Received 7 March 2017
Received in revised form 9 May 2017
Accepted 6 June 2017
Available online 13 June 2017

Keywords:
Allocation
Mathematical programming
Large cohorts

ABSTRACT

Many university degree programs (including chemical engineering ones) require final year students and Masters' students to do an extended research project under the supervision of an academic staff member. However, obtaining a satisfying allocation for both students and supervisors is often a challenging task, especially when the amount of available supervisors is particularly tight and their popularities are highly diverse.

In this article we propose a novel method based on a ranked list of supervisors and categories provided by each student, where a category corresponds to a general research area, incorporating this information into the allocation process. A student's satisfaction may therefore correspond to getting a project either with a highly ranked supervisor and/or in a highly ranked category. With this information, we propose here a systematic approach that relies on a novel mixed-integer linear programming (MILP) model based on a flexible definition of students' satisfaction. Our MILP overcomes the limitations of manual allocation approaches, which when applied to large cohorts are highly time consuming and may produce suboptimal solutions leading to poor satisfaction levels. This MILP has been applied successfully in the School of Chemical Engineering and Analytical Science of The University of Manchester with increased levels of student satisfaction.

1. Introduction

In many universities, final year undergraduates and Masters students do an extended research project. This generally makes a big contribution to the final degree classification or the final mark. Not surprisingly, students are keen to do well in their project work and thus it is important to them that they be given projects that they enjoy and which match their skills and their interests.

While in many disciplines, like humanities, the student is responsible of devising the project, in physical sciences (e.g. chemical engineering) it is much more common for an academic supervisor only to run projects closely allied to his or her own research area. In this case, it is required that academic supervisors provide a list of possible projects, which the students can then select.

In the case that the number of students requiring projects is not too large compared to the number of available supervisors,

* *Corresponding author at*: Centre for Process Systems Engineering (CPSE), Imperial College London, SW7 2AZ, United Kingdom.
E-mail address: g.guillen05@ic.ac.uk (G. Guillén-Gosálbez).

http://dx.doi.org/10.1016/j.ece.2017.06.002

it may be possible for all students to choose their favourite supervisors. If an academic staff member happens to be popular and ends up with a larger work-load than other staff, then fairness can be restored by reducing this person's teaching duties elsewhere. This process, however, runs into problems for larger cohorts of students. As will be seen later, commonly some supervisors are more popular than others, and the former would end up supervising a disproportionate amount of students. One thus needs an allocation scheme that matches students to supervisors in such a way that overall student satisfaction is maintained but no staff member supervises more than a set maximum number of students.

A very small survey of various UK Universities and their Schools/Departments revealed a number of strategies in operation for managing project allocation. In all cases there was a cap on the number of students given to an individual supervisor. One method, used to allocate final year undergraduate students, was to make use of their academic performances of the previous year. As well, all students would submit a ranked list of supervisors. The student with the top marks would get his/her first choice supervisor. One then would simply work down the list, allocating according to the first choices, until one came to a student who had chosen a supervisor whose quota had been filled. This student would then receive his/her second choice. This procedure would carry down the entire list, though, by the time you get near the bottom, there is a big risk that a student will be allocated to a supervisor either very far down the ranking or, possibly, not even on the student's list at all. This of course, is a very easy algorithm to use and it does ensure that students with high marks get a very satisfactory allocation. The method cannot be used, however, for a one year MSc programme, where there would be no ranked mark list from the previous year. Also this runs counter to the more prevalently held belief, within the small survey, that this is an unfair process, because of the potential high degree of dissatisfaction experienced by students with less good examination results. Another approach that came up in the survey was the "first come, first served" method. Here students would need to meet prospective supervisors in person and, if the student wished to work with that supervisor and the supervisor was agreeable, then that student would be allocated to him/her. This would carry on until the supervisor has filled his/her quota. Once this had happened, a student would have to look elsewhere for a project. Again, any student who failed to fix up a supervisor in good time, ran the risk of ending up with a supervisor not in his/her ranked list.

This procedure is easy to administer and it does have the virtue of requiring students to discuss projects with supervisors and thus be in a more informed situation. Drawbacks are firstly that the supervisor does his/her own selection, so there is a danger that this can be based, maybe unconsciously, on overly subjective factors. Secondly the time element can create stress for students. Indeed, the motivation is to sign up one's supervisor as quickly as possible, rather than spending the time to consider a range of possible projects. And finally, this scheme is not designed to provide an overall optimisation of student satisfaction. Those who cannot find a supervisor quickly can end up with highly unsuitable projects.

The most common approach in order to solve these deficiencies is to try to match up students and supervisors so as to optimise student satisfaction while still retaining a cap on the number of students allocated to any given supervisor. In order to do so, usually students are required to provide a ranked list of supervisors. Frequently this is done by hand, but again, this procedure does not guarantee an optimal allocation in terms of student satisfaction. Also, unexpected changes during the allocation process (like agreed allocation of one student to one supervisor) may force to re-allocate all the students again. In addition to these disadvantages, the differences on supervisors' popularity imply that some students may have to be allocated with supervisors who do not appear on their ranked lists (also being allocated to unappropriated projects).

Another classical approach for student allocation is to consider the preferences of students over specific projects. If this was applied together with the approaches presented above, it would be necessary to rank the preference of supervisors to specific projects (assuming that projects and supervisors are completely unrelated). In any case, this would only worsen even more the overall satisfaction as the allocation becomes more constrained and complex. Therefore, the use of a less restrictive satisfaction metric like the preference towards project categories (instead of specific projects) should improve the overall satisfaction. Here, project categories are general research areas (such as process modelling or bioprocesses in chemical engineering), represented by several supervisors. Generally, once an academic has a set of students to supervise, discussion can take place, taking into account the students' specific preferences, to arrive at satisfactory project allocations. Additionally, in most cases all the projects proposed by each supervisor share category (research area) and nature (approach followed, for example experimental or computational). If the projects proposed by a supervisor are very different in nature, it could be possible to add these project characteristics as categories, since both are project descriptors.

Again, as for the supervisor preference, it would be required that the students provide information about their preferred project categories (ideally together with information on preferred supervisors), so at least supervisor or category satisfaction can be achieved for each student. While this facilitates the generation of good allocations, it does not ensure that the allocation is optimal, particularly if students are to be allocated via manual procedures. Furthermore, the methodology is still very sensitive to unexpected changes in supervisors and/or students availability.

Automating the allocation process can greatly reduce the time needed to obtain a satisfactory allocation, allowing to recalculate it in reasonable time even if new unexpected changes occur. In addition, automated methods may be the only reasonable and sensible procedure in today's academic environment, which has experienced a pronounced increase in university students (for example from 2009 to 2014, the number of chemical engineering students in the UK increased by 97%) ("UCAS (Universities and Colleges Admissions Service) Database" 2016). This automation, however, requires of specifically designed algorithms (depending on the particular criteria followed when allocating), usually based on mathematical principles so as to find an optimal allocation.

Several allocation methodologies can be found in the literature, some using the principles mentioned above. The allocation (or matching) problem has been widely considered in the operations engineering (Arora and Puri, 1998; Wu and Sun, 2006) and chemical engineering (Ceccon et al., 2016; Kang and Liu, 2014) research literature, especially for scheduling of processes (Kondili et al., 1993; Méndez et al., 2006), where certain tasks have to be allocated in time periods in order to optimise the overall production time. In these studies, highly sophisticated algorithms and methodologies have been presented, efficiently organising tasks and resources so as to optimise in each case the operation cost and duration. The student-project allocation problems (SPA) in the academic context are clearly different to the engineering allocations. The academic allocations generally have fewer constraints, presenting smaller problem sizes, and the satisfaction metrics are directly related to the allocation variables, instead of, for example, the complex relationship between process parameters and operational cost. Furthermore, the project allocation problem contains only assignment decisions but no sequencing constraints, so it resembles scheduling problems with the difference being that the duration of projects and their temporal independencies, if any, play no role in the optimisation.

Due to these differences, the approaches developed to solve the SPA problems tend to be less sophisticated than the existing methods for engineering purposes, but equally efficient for the academic purposes. In this paper it is not intended to present a multi-purpose allocation method, but one that allows straightforward and satisfactory assignments of students to supervisors and therefore to projects.

In this context, the student-project allocation problem faced at present by many universities has been already considered in the research literature. The main objective in the SPA problem is to allocate students into projects, while maximising the satisfaction of both, students and supervisors. The SPA problem has been formulated as an assignment problem, for which several optimisation algorithms have been proposed. This problem was at first considered as an special case of the marriage assignment problem (Dye, 2001; McVitie and Wilson, 1971), generating the basis for future algorithmic methodologies. Along these lines, Teo and Ho (1998) presented an algorithm that considers only the preference of the students over the projects and allocates them using a recursive loop that generates multiple solutions through recursive search, selecting the best of them as the final allocation. In order to improve the allocation, Abraham et al. (2007) introduced two algorithms (developed further in Iwama et al., 2012; Manlove and O'Malley, 2008) that maximise the satisfaction of both students and supervisors. Variations of these algorithms have been created, depending on their specific application, as in the work presented by Moussa and El-Atta (2011).

Other authors presented methods based on mixed-integer linear programming (MILP) models, in which binary variables model allocation decisions (i.e. they are one when a student is allocated to a project and zero otherwise). Harper et al. (2005) defined a simple model that optimises the weighted sum of the students' satisfaction levels, where these weights are defined by the supervisors.

Anwar and Bahaj (2003) defined a formulation comprising two models (therefore two objective functions, minimisation of the number of projects given to the supervisors and maximisation of the students' satisfaction, respectively), designed to be sequentially optimised by keeping the optimal objective function values computed in previous runs as constraints for the next ones. Expanding this idea, Pan et al. (2009) defined a goal programming formulation with three objective functions (maximisation of students allocated, maximisation of students' satisfaction and maximisation of supervisors' satisfaction, respectively). As in the previous case, these objective functions were solved sequentially considering previous solutions as constraints.

Most of the previously mentioned works did consider, as in the method presented in this paper, some sort of preference metric of students towards supervisors and/or projects. However, other authors, such as Harper et al. (2005) and Pan et al. (2009) also consider not only the academic performance of the students but also the preference of academics towards students, adding another dimension to the supervisors' satisfaction.

In this paper we propose our own mixed-integer linear programming (MILP) model for the allocation problem that was tailored to the allocation needs of The School of Chemical Engineering and Analytical Science of The University of Manchester. The key feature of this MILP and what makes it unique compared to the aforementioned models is that it is based on a flexible definition of the allocation problem that allocates students to project categories and project supervisors, rather than to projects only. Hence, the students' satisfaction is maximised by fulfilling two criteria instead of one: by allocating them to their preferred supervisor and/or to their preferred project category (i.e., project topic), which provides the model with more flexibility and therefore ultimately leads to better solutions. The MILP based on this approach is capable of dealing with highly constrained problems in which it would be impossible to satisfy all the students simultaneously according to a single criterion (i.e. if all had to be allocated to their preferred supervisor only rather than to either their preferred supervisor or preferred category).

The article is organised as follows. The problem under study is formally stated first, the methodology followed is introduced afterwards. Some numerical results are then presented and discussed and the conclusions of the work are finally drawn.

2. Problem statement

The problem we aim to solve can be formally stated as follows (see Fig. 1, which also outlines the general solution procedure discussed later during the article). We are given a set of students and supervisors. Supervisors offer research projects in a set of categories (i.e. topics related to chemical engineering, such as catalysis, molecular modelling, process systems engineering, etc.). Prior to the allocation of projects, a questionnaire is filled in by the students in order to elicit their preferences. These preferences are expressed in the form of a ranking of supervisors and project categories. Expressing preferences in terms of the two criteria (i.e., categories and supervisors) provides the model with more flexibility to satisfy students, as it becomes easier to find solutions that fulfil either of the two preferences (as opposed to having to fulfil one of them compulsorily). As discussed in the case study section, this specific feature of the model leads eventually to larger satisfaction levels of both students and academics. To the best of our knowledge, this is the first time the allocation problem is defined in this way, which makes our MILP approach unique.

The goal of the problem we aim to solve here is to allocate the students to the projects available (which implies allocating them to a supervisor and project category as well) so as to maximise the overall level of satisfaction. Note that the satisfaction of a student is maximised by giving her/him the project category and project supervisor she/he prefers (the ones appearing in the top of her/his ranking). Obviously, this might not be possible for all of the students, as their preferences tend to overlap (e.g. the most popular academics are prioritized

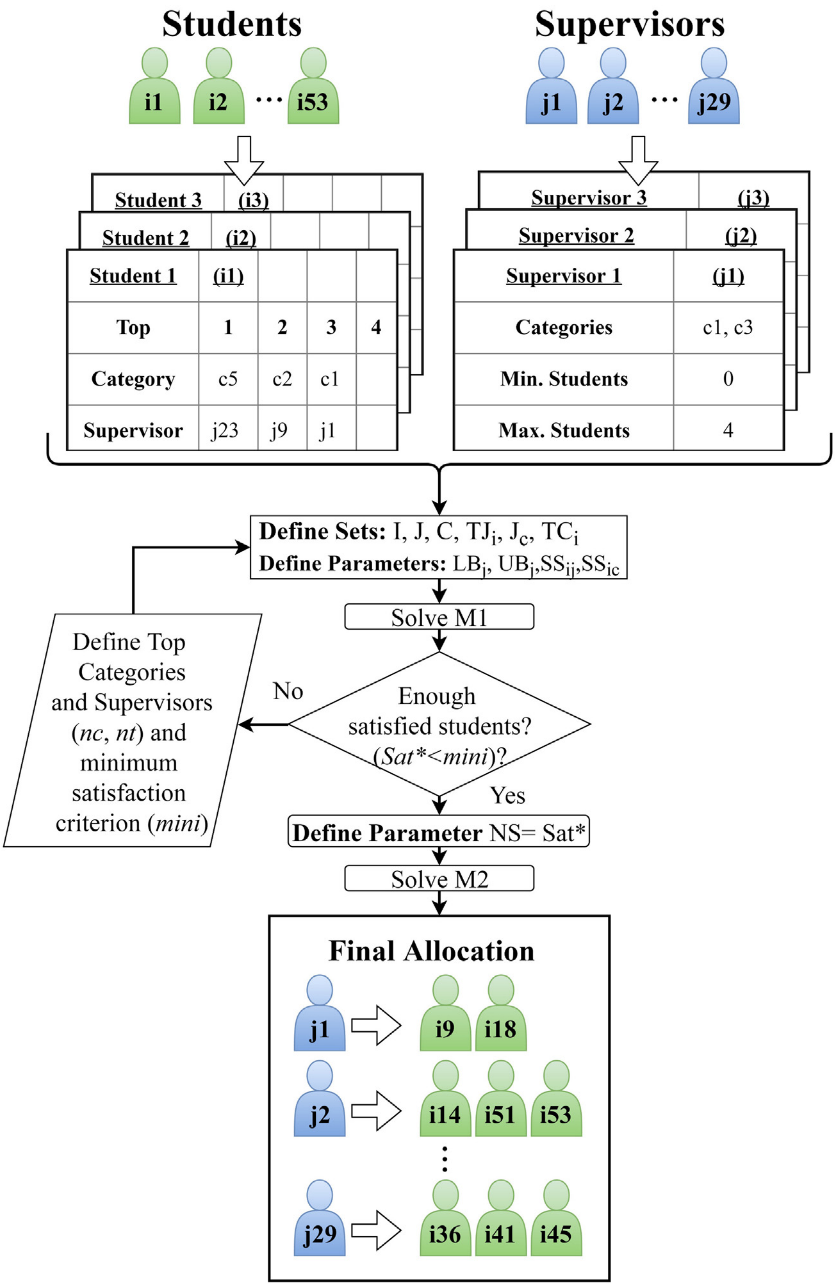

Fig. 1 – Problem representation and outline of the approach. The students' preferences expressed as a list of top supervisors and top categories, the minimum and maximum number of students that each academic can supervise, and the categories within which the supervisor's projects fall are given as inputs to the algorithm. Run M1, if all the students are satisfied, then run M2 to further refine the solution, otherwise redefine the top supervisors and top categories sets and run M1 again.

by many students, but each academic can only supervise a reduced number of projects). On the other hand, academics prefer to have a low number of students, as otherwise their teaching load (considering other duties, like lectures, PhD students' supervision, etc.) would increase. The reason why they have this perception is because the research projects, which last for three months, barely lead (unless very bright students are involved) to publishable results (or in case they do they still require much extra time from the academics involved in order to produce a manuscript of enough quality). For this reason, supervising students is considered as an extra teaching load, rather than as a research opportunity.

Additionally, in order to provide more flexibility during the allocation and ultimately improve the level of satisfaction of students, the supervisors offer several projects, sometimes even in different categories. For each supervisor, a lower and upper bound on the number of students are defined so as to make sure that the total teaching load is kept within the desired limits.

As seen, there is a clear trade-off between the satisfaction of the students (that want to be allocated to the most popular supervisors) and that associated with academics (that want to have as few students as possible). In such situations (common in multi-objective optimisation), the solution of the problem will not consist of a single allocation alternative, but rather be given by a set of optimal allocations (i.e., Pareto points), each achieving a unique combination of satisfaction levels for students and academics. As will be explained in the next section, for simplicity, we present herein a single-objective formulation where the level of satisfaction of the students is indeed maximised, while the level of satisfaction of the academics is handled through constraints that impose upper bounds on the number of students allocated to them. These bounds can be proposed by the academics, but ultimately, the department should be able to modify them to answer academics' demands or to allow for the allocation of more students. This was the case in the case study presented below.

Additional aspects could be included in the model, such as eligibility factors based on the seniority of the supervisors, available funding, etc. However, as in our case, such information often is not readily accessible, being necessary to use simpler metrics (like the maximum amount of students to be allocated to each supervisor) as proxy of the academics' satisfaction.

The resulting model is presented in detail in the ensuing sections.

3. Methodology

The approach we propose to solve the allocation problem is based on a mixed-integer linear programming (MILP) model. In essence, this MILP includes both binary and continuous variables, a set of linear constraints and a linear objective function. As already mentioned, the model should consider two contradicting objectives: the level of satisfaction of students and the level of satisfaction of academics. To keep the MILP simple, we present here a single-objective formulation that maximises the students' satisfaction. The satisfaction of the academics is maximised in this MILP by allocating to them no more than a maximum number of students. This upper bound on the number of students is defined beforehand taking into account other research, administrative and teaching tasks carried out by the supervisors. The MILP is described in detail next.

3.1. Allocation constraints and bounds

We define binary variable x_{ij} that takes a value of one if student i is allocated to supervisor j and zero otherwise. Every student must be allocated to one supervisor, a condition enforced via the following constraint:

$$\sum_{j \in J} x_{ij} = 1, \forall i \in I \tag{1}$$

In this equation, J is the set of all supervisors and I is the set of all the students.

We define lower bounds on the number of students allocated to each supervisor in order to ensure (to the maximum extent possible) a flat allocation of students to supervisors, which reflects a more equitable scenario, as follows:

$$LB_j \leq \sum_{i \in I} x_{ij} \leq UB_j, \forall j \in J \tag{2}$$

where LB_j and UB_j represent the minimum and maximum number of students allocated to academic j, respectively.

3.2. Objective function

The level of satisfaction of the students is quantified by the number of them for which the preferences expressed in the questionnaire are fully met. More precisely, we assume that a student is satisfied if she/he is given a project that either belongs to her/his top categories or is supervised by her/his top supervisors. The concept of "top" needs to be defined in more detail. In practice, the students are told that they will be assigned a project within the top n categories or top n supervisors (i.e., top 3, top 4, and so on). This condition can however be relaxed if by doing so more students are fully satisfied. This point is further clarified in the results section.

Hence, the model seeks to maximise the following objective function:

$$Sat = \sum_{i \in I} y_i \tag{3}$$

where Sat is the objective function value (overall students' satisfaction) and y_i is a binary variable that takes the value of one if the student is satisfied because either allocation criterion is met (i.e., the student was allocated to one of her/his top categories or to one of her/his top supervisors), and it is zero otherwise (i.e., the student is allocated to neither one of the top categories nor one of the top supervisors). The definition of binary variable y_i is enforced via the set of constraints described below.

Eq. (4) is used to calculate the value of the slack variable z_i^{sup}, which is one if the student is not allocated to her/his top supervisor (is dissatisfied) and it is zero otherwise.

$$\sum_{j \in TJ_i} x_{ij} + z_i^{sup} = 1, \forall i \in I \tag{4}$$

Note that z_i^{sup} can be defined as a continuous variable, as the model will force it to take a 0-1 value anyway. In this equation, recall that x_{ij} models the allocation of student i to supervisor j, while TJ_i is the set of top supervisors for student i (e.g., top 2, top 3, etc., depending on the case). Hence, if the student is not allocated to any of the top supervisors (in the set TJ_i), then the summation of binary variables x_{ij} will be zero, and variable z_i^{sup} will have to take a value of one in order to satisfy the equality constraint.

$$z_i^{cat} \leq 1 - \sum_{j \in J_c} x_{ij} \forall i \in I, \forall c \in TC_i \tag{5}$$

$$z_i^{cat} \geq 1 - \sum_{c \in TC_i} \sum_{j \in J_c} x_{ij}, \forall i \in I \tag{6}$$

Eqs. (5) and (6) are used to determine the value of the slack variable z_i^{cat}, which is one if student i is not allocated to a top category and zero otherwise. In these equations, TC_i is the set of top categories defined by student i. Hence, in Eq. (5), if the student is allocated to a project in the student's top category, then the summation of x_{ij} will be one. This will force the variable z_i^{cat} to be zero (as it is defined as strictly positive). Furthermore, in Eq. (6) the summation of x_{ij} will be one as well, and the inequality will be inactive. If the student is not allocated to any project in his/her top categories, then Eq. (5) will force the slack variable to be less than or equal to one, while Eq. (6) will make the slack variable greater than or equal to one. Therefore, the slack variable will have to take a value of one in the latter case. Note that Eqs. (5) and (6) are slightly different than Eq. (4) because they need to model the case in which a supervisor offers more than one category. Hence, if a student was allocated to a supervisor that has more than one of the student's top categories, an equivalent summation as the one in Eq. (4) would lead to values greater than one (because the binary variable allocating the student to the supervisor would be summed up several times, as the same supervisor would appear in several categories), making the slack variable negative.

Constraints (7)–(9) are added to calculate the value of binary variable y_i, which is one if any (or both) of the two criteria is satisfied, and it is zero otherwise. More precisely, Constraints (7) and (8) force y_i to take a value of one when either z_i^{cat} or z_i^{sup} are zero (i.e., the student is satisfied when any of the two criteria is met). Note that in this case constraint 9 is inactive. When neither criterion is satisfied, then Constraints (7) and (8) are inactive, while Constraint (9) forces y_i to take a value of 0.

$$y_i \geq 1 - z_i^{cat}, \forall i \in I \tag{7}$$

$$y_i \geq 1 - z_i^{sup}, \forall i \in I \tag{8}$$

$$y_i \leq 2 - z_i^{cat} - z_i^{sup}, \forall i \in I \tag{9}$$

Finally, the model (model M1) can be expressed in compact form as follows:

Model M1:

$$\max Sat = \sum_{i \in I} y_i$$

s.t. Eqs. (1)–(9)

$$y_i, x_{ij} \in \{0, 1\}$$

$$z_i^{cat}, z_i^{sup} \in \mathbb{R}^+$$

Note that the optimal solution identified when solving M1 can be further improved without affecting the total number of students satisfied. This is because the algorithm makes no distinction between the members of the sets TJ_i and TC_i, so swaps between students might lead to larger satisfaction levels. As an example, consider that two students, say A and B, have been allocated to her/his second supervisor, say A to C and B to D, but it turns out that the top supervisor (most preferred one) of A is D, while the top of B is C. Hence, we can easily generate then another solution where A is assigned to D and B to C, thereby increasing the satisfaction levels of A and B simultaneously. Let us clarify that in both solutions, students A and B are satisfied (understanding by satisfied the fact that they were allocated to any member within their top supervisors set), but clearly in the second case they will work with their very top choice (first option in the questionnaire) rather than with the second in their preference list. Hence, it is clear that their level of satisfaction will increase by a simple swap.

Hence, to further optimize the solution provided by the MILP, we can proceed as follows. We define the following objective function:

$$OF = \sum_{i \in I} \sum_{j \in TJ_i} x_{ij} SS_{ij} + \sum_{i \in I} \sum_{c \in TC_i} \sum_{j \in J_c} x_{ij} SC_{ic} \tag{10}$$

That is, the new objective function is the maximisation of a total score, which is calculated from the scores of each project category (SC_{ic}) and supervisor (SS_{ij}) and the assignment variable x_{ij}. There are different manners to define the score values, which will affect the outcome of the optimisation. In our case, these values are determined from the questionnaires. A possible way to proceed is the following: to give the top supervisor or category a given maximum amount of points p (where p is the cardinality of the set of top categories/supervisors), $p - 1$ points to the second option, and so on until the last member in the preference list receives only one point, while the rest are given zero points. The MILP for refining the solution provided by model M1 can then be expressed as follows:

Model M2:

$$\max.OF = \sum_{i \in I} \sum_{j \in TJ_i} x_{ij} SS_{ij} + \sum_{i \in I} \sum_{c \in TC_i} \sum_{j \in J_c} x_{ij} SC_{ic}$$

s.t. Eqs. (1)–(9)

$$\sum_{i \in I} y_i = NS \tag{11}$$

$$y_i, x_{ij} \in \{0, 1\}$$

$$z_i^{cat}, z_i^{sup} \in \mathbb{R}^+$$

In essence, it is the same model as M1, but with a new objective function and an additional constraint (Eq. (11)). This constraint forces the model to satisfy the same number of students as in the solution that we aim to improve. Note that omitting this constraint could lead to larger total scores at the expense of leaving some students unsatisfied, a situation that we want to avoid. Note also that M2 is guaranteed to be feasible, as it is run to improve a solution that is already feasible (i.e. we first need to identify an optimal solution for M1 that satisfies a given number of students, and then fix this number into model M2 and try to find a better solution in terms of total score, knowing beforehand that the original solution to M1 is in turn feasible in M2, and therefore M2 cannot render unfeasible).

Our approach to solve the allocation problem is therefore as follows:

1. Solve model M1 for a given set of nt top supervisors and nc top categories (redefine the sets TJ_i and TC_i, such that $|TJ_i| = nt$, $|TC_i| = nc$).
2. If $Sat^* < mini$, where Sat^* is the optimal objective value of M1, that is, if the total number of students satisfied is lower than a minimum value ($mini$), then increase either nt or nc (or both), and go to step 1; (nested condition) if nt and nc cannot be increased any further, decrease $mini$ and go to step 1. If the condition is satisfied, then go to step 3. By default, the minimum number of satisfied students ($mini$) is typically set equal to the total number of students (as is in the case study below), unless, due to external constraints, not all the students can be satisfied.
3. Define $NS = Sat^*$ and solve M2. The solution to M2 is the final allocation sought.

4. Case study and data

The capabilities of our MILP approach are illustrated through its application to the allocation of students in the School of Chemical Engineering and Analytical Science, The University of Manchester, UK, for the academic year 2016/2017 with a total of 53 students ($I = \{i_1, \ldots, i_{53}\}$) and 29 academics ($J = \{j_1, \ldots, j_{29}\}$), who offered 53 projects in five categories (i.e., Molecular and rheological modelling (c1), Process design and modelling (c2), Bioprocesses (c3), Catalysis and chemical reactivity (c4) and Sensing and Healthcare (c5)). To illustrate the benefits of using our systematic approach, the results obtained are compared with the original practice ("first come, first served").

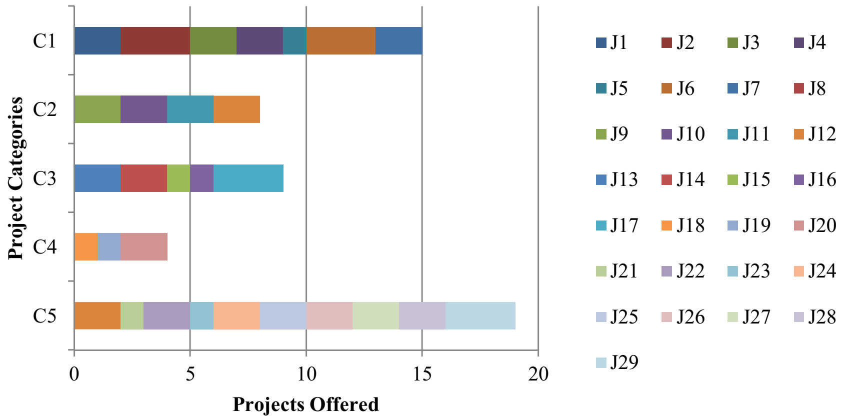

Fig. 2 – Projects offered by each supervisor $J = \{j_1, \ldots, j_{29}\}$ within each category (C1–C5).

Table 1 – Maximum number of students for each supervisor.

Supervisor	Max. Students	Supervisor	Max. Students
J1	2	J16	1
J2	3	J17	3
J3	2	J18	1
J4	2	J19	1
J5	1	J20	2
J6	3	J21	1
J7	2	J22	2
J8	0	J23	1
J9	2	J24	2
J10	2	J25	2
J11	2	J26	2
J12	2	J27	2
J13	2	J28	2
J14	2	J29	3
J15	1		

Fig. 2 displays the number of projects offered by each supervisor within each category. As observed, the supervisors have different backgrounds. Most of them offered several projects within the same category, while very few offered projects in two categories (mainly those combining experiments with modelling).

Additionally, each supervisor can accept a minimum and maximum number of students (the minimum for all cases is zero). In defining these bounds, we need to make sure that all the academics together can allocate the total amount of students considered. The maximum acceptable students for each supervisor are detailed in Table 1.

Figs. 3 and 4 summarise the outcome of the questionnaires passed to the students, namely the ranking of categories according to the students' preferences. The two figures are essentially very similar, but differ in the units in which the preferences are expressed (i.e. in Fig. 3 we show the scores, that is, the total number of points received by each category—4 points to the top category, 3 points to the second one, and so on—while in Fig. 4 we display the number of students that ranked the category as either top 4, 3, 2 or 1). As an example, as shown in Fig. 3, 16 students voted for c4 as first option, 11 as second, 11 as third, and 3 as fourth. This means that C4 received 64 points from the students that chose it in first place (16 multiplied with 4), 33 from the students that chose it in second place (11 multiplied with 3), 22 from the students that chose it in third place (11 multiplied with 2), and 3 from the students that chose it in fourth place (3 multiplied with 1). Note that a category could receive many votes of "low quality" if it is chosen by many students who give it low scores. As an example, category C1 received 31 votes, but most of them were "top 4", so consequently in Fig. 4 the bar associated with C1 is smaller (in relative terms with respect to the other categories) than that in Fig. 3.

Figs. 5 and 6 are equivalent to Figs. 3 and 4, respectively, but apply to the preferences on supervisors rather than preferences on categories.

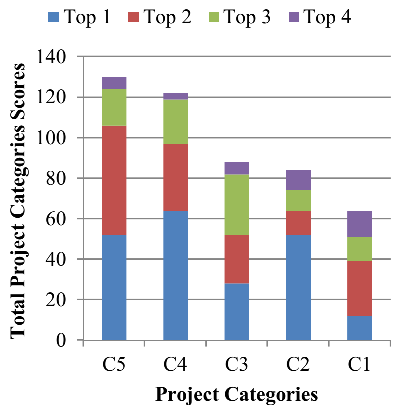

Fig. 3 – Students' scores for the project categories. The vertical axis represents the total scores given by the students to their top categories, while the horizontal axis represents the 5 project categories available.

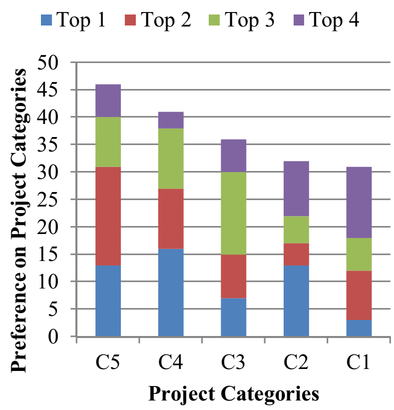

Fig. 4 – Students' classification of project categories. The vertical axis represents the amount of students that ranked the category in a given way (top 4, 3, 2 or 1), while the horizontal axis represents the five project categories available.

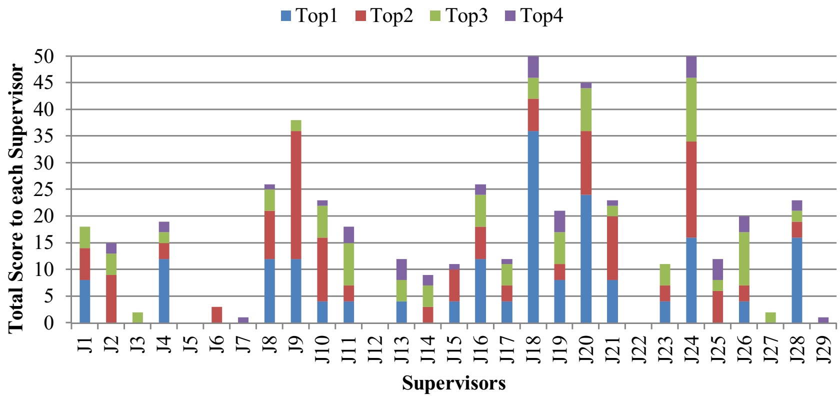

Fig. 5 – Students' scores for supervisors. The vertical axis represents the total score given by the students to their top supervisors, while the horizontal axis represents the supervisors.

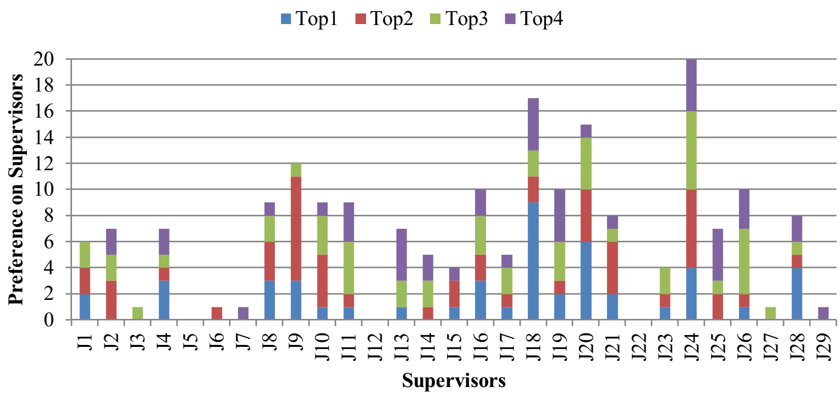

Fig. 6 – Students' classification of supervisors. The vertical axis represents the preferences given by the students to their top supervisors, while the horizontal axis represents the supervisors.

As seen in these figures, the preferred category is C5, followed by C4 and C3. In terms of supervisors, the preferred ones are J24 (chosen by more students as compared to the other supervisors) and J18 (most preferred as top 1).

Note that given these data, it is virtually impossible to allocate all the students to their top category. As an example, 16 students selected C4 as very top category (first in the preference list), but it turns out that there are only four projects offered in this category (and in addition to this, a full allocation of students to this category might violate constraints on the minimum and maximum number of students allocated to each supervisor). Similarly, it is not possible to allocate all the students to their top supervisor. A total of nine students selected academic J18 as top supervisor, but he offered only one project.

After performing the preliminary analysis of the data discussed above, we constructed the MILP M1, which featured 485 constraints, 160 continuous variables and 1,537 binary variables. The model was implemented in GAMS 24.7 and solved with CPLEX 12.6 on an Intel Core i5-4570 3.20 GHz computer. The CPU time was 0.14 s.

We started by defining as top supervisor only the first in the preference list and the same for the top project category. After running the MILP M1, we obtained a solution with 37 students satisfied (out of 53). Clearly, the MILP M1 is unable to find a solution that satisfies all the students simultaneously when only the very top supervisor and category of each student are considered in the allocation.

We then solved several MILPs M1 for different potential definitions of top category and top supervisor. The results of this analysis are displayed in Table 2, where the rows are the number of top supervisors that belong to the set TJ_i (parameter nt), while the columns are the number of categories that belong to TC_i (parameter nc). The number inside the cell represents the number of students for which either of these two criteria is met and therefore are satisfied with the allocation (parameter $mini$). As it can be seen, for the top two categories and/or top four supervisors, the MILP M1 can identify a solution where all the students are satisfied, while the same is impossible to achieve when we force the model to allocate the student in the very top category.

Table 2 – Students satisfied for each combination of top categories and top supervisors obtained by solving M1 (maximisation of satisfied students). Solutions that cannot satisfy all the 53 students are labelled with *.

Satisfied students		Categories			
		Top 1	Top 2	Top 3	Top 4
Supervisors	Top 1	37*	52*	53	53
	Top 2	46*	52*	53	53
	Top 3	49*	53	53	53
	Top 4	50*	53	53	53

To illustrate the benefits of our approach compared to the standard method that maximises the total score regardless of the number of students satisfied, we ran next model M2 without any limit on the total number of satisfied students. More precisely, M2 was executed maximising the scores given by the allocation to academics (without any constraint on the minimum number of students satisfied). The results are shown in Table 3. As seen, the model execution leads to worse solutions in terms of total number of students satisfied than those

Table 3 – Students satisfied for each combination of top categories and top supervisors solving M2 maximising the total academic allocation score. Solutions that cannot satisfy all the 53 students are labelled with *.

Satisfied students		Categories			
		Top 1	Top 2	Top 3	Top 4
Supervisors	Top 1	31*	40*	48*	51*
	Top 2	38*	43*	47*	51*
	Top 3	40*	45*	47*	51*
	Top 4	42*	47*	48*	50*

produce by our approach. This happens because when we maximise scores only, the model seeks to allocate students to their very top category or supervisor (the first one in their list), as the objective function greatly improved by doing so. This has the negative side effect that to accomplish this the model may decide to leave other students unsatisfied. This caveat of models based solely on maximising scores highlights the need to apply a more flexible definition of satisfaction like the one adopted in this work.

Model M2 can be solved for any combination of top categories and supervisors in M1. However, using weaker constraints on the top categories and supervisors lead to larger satisfaction scores in M2. This happens because the feasible region of M2 becomes larger. Furthermore, a larger feasible region facilitates last minute allocation modifications. On the other hand, when using weaker constraints, most of the students are assigned to less preferred academics and categories, potentially raising students' complaints. Therefore, it is recommended to use intermediate values for the feasible top categories and supervisors.

Following this idea, we finally selected the solution for top 2 categories and top 4 supervisors (Table 4). This solution was later refined using M2 and following the improvement strategy described before while forcing the model to satisfy all of the students (to allocate them either to the top 2 categories or top 4 supervisors).

The second MILP M2 identified a solution with a total score of 324 (vs 185 in the original solution) (Table 5). In this solution, the allocation of students greatly differed from the previous one (only two out of 53 students were allocated to the same supervisors as in the previous solution produced by M1). This final solution was the one sent to academics and students, who benefited from the use of a systematic approach for allocating projects vs the traditional method that took more time and led to worse solutions.

The benefits of the presented methodology were further investigated through its comparison with the "first come, first served" original practice. This methodology can be expressed as follows. Each student is assigned to any of the top available supervisors (supervisors with higher preference in the student's list are allocated first). If no preferred supervisors are available, the student is assigned to a supervisor offering a project on any of his/her top project categories (supervisors with categories ranked higher in the preference first). If none of the available supervisors have projects in the top categories, assign the student to a supervisor with more unassigned projects. Repeat this procedure for all the students in order of request arrival. Table 6 presents the allocation obtained when applying this "first come, first served" practice on the considered case study.

As can be seen in Table 6, the original practice was not able to ensure a minimal satisfaction for all the students (6

Table 4 – Student-Supervisor allocation from model M1, considering top 2 categories and top 4 supervisors.

Supervisors	Students		
J1	i2	i25	
J2	i4	i8	i18
J3	i35	i52	
J4	i22	i23	
J5	i46		
J6	i21	i49	i51
J7	i31	i33	
J8			
J9	i26	i45	
J10	i41	i48	
J11	i13	i42	
J12	i12	i37	
J13	i34	i39	
J14	i5	i30	
J15	i10		
J16	i29		
J17	i36	i44	i47
J18	i19		
J19	i50		
J20	i14	i43	
J21	i53		
J22	i32	i40	
J23	i28		
J24	i6	i38	
J25	i11	i16	
J26	i15	i24	
J27	i7	i20	
J28	i17	i27	
J29	i1	i3	i9

Table 5 – Student-Supervisor allocation from model M2, considering top 2 categories and top 4 supervisors.

Supervisors	Students		
J1	i9	i18	
J2	i14	i51	i53
J3	i10	i49	
J4	i20	i52	
J5	i8		
J6	i7	i13	i22
J7	i4	i25	
J8			
J9	i26	i40	
J10	i37	i43	
J11	i23	i28	
J12	i27	i31	
J13	i2	i39	
J14	i6	i48	
J15	i29		
J16	i19		
J17	i21	i34	i47
J18	i24		
J19	i3		
J20	i17	i32	
J21	i42		
J22	i12	i38	
J23	i1		
J24	i30	i33	
J25	i15	i44	
J26	i11	i16	
J27	i5	i50	
J28	i35	i46	
J29	i36	i41	i45

Table 6 – Student-Supervisor allocation following the heuristic "first come, first served", considering top 2 categories and top 4 supervisors as satisfaction criteria. Unsatisfied students are labelled with *.

Supervisors	Students		
J1	i9	i18	
J2	i14	i37*	i51
J3	i46*	i52	
J4	i13	i20	
J5	i47*		
J6	i22	i50*	i53
J7	i48*	i49	
J8			
J9	i4	i8	
J10	i26	i27	
J11	i23	i28	
J12	i31	i43	
J13	i2	i29	
J14	i6	i45*	
J15	i19		
J16	i10		
J17	i21	i34	i39
J18	i5		
J19	i3		
J20	i12	i15	
J21	i30		
J22	i38	i40	
J23	i1		
J24	i7	i17	
J25	i24	i32	
J26	i16	i33	
J27	i35	i36	
J28	i11	i25	
J29	i41	i42	i44

students remain unsatisfied). As well, the overall satisfaction score obtained (291) is significantly lower than the score obtained when solving M2 (324). The satisfaction improvement obtained is also manifested in the feedback from the students, ranking 3/5 the allocation obtained with the original practice and 4.5/5 the allocation with the new system. From these results we conclude that the presented methodology outperforms the original allocation practice in all the aspects evaluated, ultimately helping to improve the students' satisfaction during the project allocation.

5. Conclusions

In order to successfully allocate projects to a large cohort of students, we argue that it is convenient to consider not only the preferences for supervisors but also the preferences for subject categories. This provides the allocation task with much more flexibility to find a satisfactory solution for all the parties involved. In order to achieve an optimal allocation following these principles, we have developed a flexible MILP-based approach tailored to the needs of our studies. This algorithm has been used to allocate the MSc in Chemical Engineering students in the School of Chemical Engineering and Analytical Science, The University of Manchester, with the result that the vast majority of students obtained projects either in one of their two most highly ranked supervisors or in their top two categories.

The unique feature of this MILP is that, based on the observation above, it considers two satisfaction criteria simultaneously when performing the allocation: preferences on supervisors and preferences on project categories. This ultimately leads to better satisfaction levels, avoiding allocations in which some students are given supervisors and/or categories (topics) far from their interests (which has the potential risk of creating frustration and leading to worse academic performance).

The MILP identifies in a systematic manner (and very quickly) solutions that improve the level of satisfaction of students and academics. Compared to manual approaches, our method saves significant time (and therefore money), producing solutions with much better satisfaction levels than those attained via heuristics and which would be very difficult (if not possible) to identify via trial and error methods and/or rules of thumb.

Our approach, which can be easily implemented in off-the-shelf optimisation packages and which we can make available upon request, is intended to assist academics during the planning of their teaching duties in order to make a better use of the resources available.

No allocation system can guarantee every student his/her first choice when the number of students is significantly greater than the number of supervisors, but the evidence is that our hybrid system outperforms previous allocation methodologies, greatly increasing general satisfaction and versatility.

Acknowledgements

AJM acknowledges a very helpful conversation with Mark Wilson (Chemistry, Durham) where Mark explained his ideas about the use of categories in project allocation.

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

Abraham, D.J., Irving, R.W., Manlove, D.F., 2007. Two algorithms for the Student-Project Allocation problem. J. Discret. Algorithms 5, 73–90, http://dx.doi.org/10.1016/j.jda.2006.03.006.

Anwar, A.A., Bahaj, A.S., 2003. Student project allocation using integer programming. IEEE Trans. Educ. 46, 359–367, http://dx.doi.org/10.1109/TE.2003.811038.

Arora, S., Puri, M.C., 1998. A variant of time minimizing assignment problem. Eur. J. Oper. Res. 110, 314–325, http://dx.doi.org/10.1016/S0377-2217(97)00266-X.

Ceccon, F., Kouyialis, G., Misener, R., 2016. Using functional programming to recognize named structure in an optimization problem: application to pooling. AIChE J. 44, 3085–3095, http://dx.doi.org/10.1002/aic.15308.

Dye, J., 2001. A Constraint Logic Programming Approach to the Stable Marriage Problem and Its Application to Student-project Allocation. BSc Honours Proj. report, Dep. Comput. Sci. Univ. York.

Harper, P.R., de Senna, V., Vieira, I.T., Shahani, A.K., 2005. A genetic algorithm for the project assignment problem. Comput. Oper. Res. 32, 1255–1265, http://dx.doi.org/10.1016/j.cor.2003.11.003.

Iwama, K., Miyazaki, S., Yanagisawa, H., 2012. Improved approximation bounds for the Student-Project Allocation problem with preferences over projects. J. Discret. Algorithms 13, 59–66, http://dx.doi.org/10.1016/j.jda.2012.02.001.

Kang, L., Liu, Y., 2014. Target-oriented methodology on matching heat transfer areas for a multiperiod heat exchanger network retrofit. Ind. Eng. Chem. Res. 53, 17753–17769, http://dx.doi.org/10.1021/ie502691u.

Kondili, E., Pantelides, C.C., Sargent, R.W.H., 1993. A general algorithm for short-term scheduling of batch operations—I.

MILP formulation. Comput. Chem. Eng. 17, 211–227, http://dx.doi.org/10.1016/0098-1354(93)80015-F.

Manlove, D.F., O'Malley, G., 2008. Student-Project Allocation with preferences over Projects. J. Discret. Algorithms 6, 553–560, http://dx.doi.org/10.1016/j.jda.2008.07.003.

McVitie, D.G., Wilson, L.B., 1971. The stable marriage problem. Commun. ACM 14, 486–490, http://dx.doi.org/10.1145/362619.362632.

Méndez, C.A., Cerdá, J., Grossmann, I.E., Harjunkoski, I., Fahl, M., 2006. State-of-the-art review of optimization methods for short-term scheduling of batch processes. Comput. Chem. Eng. 30, 913–946, http://dx.doi.org/10.1016/j.compchemeng.2006.02.008.

Moussa, M.I., El-Atta, A.H.A., 2011. A visual implementation of student project allocation. Int. J. Comput. Theory Eng. 3, 178–184.

Pan, L., Chu, S.C.K., Han, G., Huang, J.Z., 2009. Multi-criteria Student Project Allocation: A Case Study of Goal Programming Formulation with DSS Implementation. Eighth Int. Symp. Oper. Res. Its Appl. Zhangjiajie, China, 75–82.

Teo, Y., Ho, D.J., 1998. A systematic approach to the implementation of final year project in an electrical engineering undergraduate course. IEEE Trans. Educ. 41, 25–30, http://dx.doi.org/10.1109/13.660783.

UCAS (Universities and Colleges Admissions Service), 2016. Database [WWW Document], https://www.ucas.com/corporate/data-and-analysis (Accessed 9 January 2016).

Wu, M.-C., Sun, S.-H., 2006. A project scheduling and staff assignment model considering learning effect. Int. J. Adv. Manuf. Technol. 28, 1190–1195, http://dx.doi.org/10.1007/s00170-004-2465-0.