Ann Oper Res (2019) 275:39–78
https://doi.org/10.1007/s10479-017-2710-1

S.I.: PATAT 2016

Handling preferences in student-project allocation

Marco Chiarandini[1] · Rolf Fagerberg[1] · Stefano Gualandi[2]

Published online: 18 November 2017

Abstract We consider the problem of allocating students to project topics satisfying side constraints and taking into account students' preferences. Students rank projects according to their preferences for the topic and side constraints limit the possibilities to team up students in the project topics. The goal is to find assignments that are fair and that maximize the collective satisfaction. Moreover, we consider issues of stability and envy from the students' viewpoint. This problem arises as a crucial activity in the organization of a first year course at the Faculty of Science of the University of Southern Denmark. We formalize the student-project allocation problem as a mixed integer linear programming problem and focus on different ways to model fairness and utilitarian principles. On the basis of real-world data, we compare empirically the quality of the allocations found by the different models and the computational effort to find solutions by means of a state-of-the-art commercial solver. We provide empirical evidence about the effects of these models on the distribution of the student assignments, which could be valuable input for policy makers in similar settings. Building on these results we propose novel combinations of the models that, for our case, attain feasible, stable, fair and collectively satisfactory solutions within a minute of computation. Since 2010, these solutions are used in practice at our institution.

Keywords Bipartite matching with one-sided preferences · Student-project allocation problem · Mixed integer linear programming · Fair assignment · Lexicographic optimization · Ordered weighted averaging · Profile-based optimization · Envy-free division

✉ Marco Chiarandini
marco@imada.sdu.dk

Rolf Fagerberg
rolf@imada.sdu.dk

Stefano Gualandi
stefano.gualandi@unipv.it

1 Department of Mathematics and Computer Science, University of Southern Denmark, Campusvej 55, 5230 Odense M, Denmark

2 Department of Mathematics, University of Pavia, via Ferrata, 1, 27100 Pavia, Italy

1 Introduction

At the Faculty of the Natural Sciences of the University of Southern Denmark (SDU) students in the first year of their bachelor degree education must undertake a course consisting of a group project. The course aims to give to the students a first experience with respect to methods and processes of scientific research and collaborative work under the supervision and facilitation of an expert researcher. The topic of the project has to fall within one of the disciplines of the natural sciences present at SDU and students are free to choose a project from almost any of these disciplines, independently of the specific branch of science in which they enrolled and will later specialize.

In a first phase, prospective supervisors, chosen among the scientific staff of the faculty, publish a one-page description of the topic of the project in which they are willing to supervise. For each topic they also state how many teams of students they can support in this specific project topic and give bounds on the sizes of these teams. Typically, teams can receive from a minimum of 3 students to a maximum of 6 students. In the next phase, students indicate their preferences by ranking a subset of project topics in which they would like to be engaged or that they find at least acceptable. They can register in groups together with a maximum of two other peers, if they want to be assigned to the same team.

The committee administrating the course collects by a given deadline all the pieces of information and computes, before the course starts, an allocation of students to projects such that: (i) each student is assigned to one team within the subset of project topics that he/she finds acceptable; (ii) students that registered in groups are assigned to the same team; (iii) the size of the teams satisfies the bounds for the project topic in question; (iv) the number of teams within each topic does not exceed the one offered; and (v) the assignment is optimal according to some satisfaction criterion. Students' preferences must be taken into account in the definition of the satisfaction, which is desirable to foster engagement and enthusiasm. In 2016, the number of students reached 416 and the number of project teams that could be created was 160. Even without the quality requirement (v) the allocation task with these numbers is not easy.

The problem is a special case of *resource allocation*. In this context, the selection among allocations that respect the limitations on resources is done on the basis of social welfare, also known as global satisfaction. Only allocations that are Pareto efficient are of interest. These are all allocations in which nobody can improve his/her situation without someone else worsening it. There can be many such allocations. To decide on one of them, the typical trade-off to be resolved is between fairness and total utility but there is no canonical definition for these criteria. For example, fairness can be determined by a just allocation process (procedural justice) or by a just outcome of the allocation (distributive justice) (Rawls 1971).

Initially, we focused on fairness as a just process guaranteeing equal chance to everybody to be allocated in their preferences. We then designed an allocation procedure that was a form of lottery, which however was not very good at dealing with some of the constraints of the allocation, namely constraints (i) and (iii). Moreover, being based on chances only, the allocation often ended up particularly unlucky for a few students, that is, the amount of inequality in the outcome was large. This procedure was in use in the first three years of the course from 2008 to 2010 but, as the size of the problem increased, the violation of constraints (i) and (iii) made the procedure impractical.

In this work, we set out to contribute to the modeling and to the solution of the student allocation problem with preferences on the side of students (also denoted one-sided preference matching). We put forward mixed integer linear programming (MILP) models to find feasible allocations that: (i) are Pareto efficient; (ii) are fair, in the sense that the most disadvantaged students are allocated in the best possible preference, thus promoting a certain degree of egalitarianism in the outcome, and (iii) maximize a function that aggregates the individual preferences met. We compare alternative ways to combine these criteria. Our final goal is to identify models that are satisfactory for all persons involved and are practical from the computational point of view. In our case, we have at most one week of time from when all data become available to when a solution has to be communicated to the students and we want to hedge against last minute changes in the data, which may arise after inspection of a solution by the administering committee. Hence, we aim at computation times to solve our instances of the problem in the order of a few minutes.

The article is organized as follows. In Sect. 2, we formulate our specific instance of student project allocation problem as a constraint optimization problem using ideas from social choice theory to define the optimization criteria. In particular, we capture the concept of *fairness*, or egalitarian individual welfare, by minimax optimization and the concept of *collective (or social) welfare (satisfaction)* by weighted and lexicographic optimization. Further, an important property for the acceptance of the allocation in practice is that the agents involved do not see easy improvements by considering only their own situation. This led us to take into account also an *envy-freeness* criterion and a *stability* criterion, this latter inspired by the concept of stable assignments in matching literature. We review the literature related to our problem and to these concepts of optimality in Sect. 3. In Sect. 4 we describe the lottery algorithm in use until 2010 and illustrate the inadequate results that it would have produced in the years from 2011 to 2016.

Our main contribution is in Sect. 5 where we model the student-project assignment problem within the framework of MILP. Beside the minimax and weighted sum, we formulate the lexicographic criteria using the so-called *distribution approach* (or profile-based optimization) and the *ordered weighted averaging* scheme by Yager (1996). This latter approach yields models that are computationally easier to solve than under the profile-based optimization approach. Further, we present a formulation, originally proposed in another context by Hooker and Williams (2012), that aims at handling the trade-off between global utility and equity. Next, we include the two other criteria that were not yet considered in the literature on this problem: envy-freeness and stability. The first minimizes a certain definition of envy between students, given in Sect. 5.5. The latter promotes allocations where students cannot see easy ways to improve their situation by, for example, finding a free place in a team of a topic they preferred to the topic, to which they have been assigned. As we will show in the experimental section, if not controlled, these situations may arise due to the lower bounds on the team sizes and may undermine the trust in the quality of the allocation mechanism. This criterion is expressed formally in Sect. 5.6.

We compare empirically these formulations both with respect to computational effort and to the quality of the allocations produced on the real life test instances from our application. The test set is composed of nine instances with a number of students from 200 to 420. Results are presented and discussed in Sect. 6. They show that our instances are solved by a state-of-the-art MILP solver to optimality in almost all formulated criteria within computation times of the order of tens of seconds. We conclude that three formulations are to be preferred: two profile-based optimization approaches and a weighted sum approach. All three formulations take into account also stability and, indirectly, envy-freeness. We leave the decision about which one of the three allocations to use to the final user (in our case the administering committee). In all our instances, the allocations outperform those produced by the lottery procedure under all the considered quality assessment criteria. Our solutions have been accepted by the administrating committee and are in use since 2011.

2 Problem definition

In the student-project allocation problem that arises at our institution, a set of *project topics* $\mathscr{P} = \{P_1, \ldots, P_n\}$ are offered to a set of *students* $S = \{s_1, \ldots, s_M\}$. For each topic P_i a limited number t_i of *teams* with the same minimum and maximum capacity can be created by allocating students to them. It will simplify our presentation to represent directly the set of *project teams* $P = \{p_1, \ldots, p_N\}$ generated from $\mathscr{P}$ by creating for each topic P_i, t_i replicates $p_j, \ldots, p_{j+t_i-1}$ in P. Each project topic is thus, in fact, a set of project teams, i.e., $P_i = \{p_j, \ldots, p_{j+t_i-1}\}$ for $i = 1, \ldots, n$ and the set P is partitioned by the set $\mathscr{P}$ and $N = \sum_{P_i \in \mathscr{P}} t_i$. Each team p_j from P has then an associated minimum and maximum capacity, l_j and u_j, respectively, which is the same for all teams that belong to the same topic.

Students can register in *groups* of at most $\ell = 3$ persons, if they want to be assigned to the same team. That is, the set of students S can be partitioned into groups, $\mathscr{G} = \{G_1, \ldots, G_m\}$ with $m \leq M$. We denote by a_k the cardinality of each group $G_k \in \mathscr{G}$. Students express their *preferences* by providing a subset of project topics ranked in *strict order*. The ranking for a student s_q is denoted by an ordered set $R(s_q) = (P^{(1)}, \ldots, P^{(n_q)})$, $P^{(r)} \in \mathscr{P}$, $1 \leq r \leq n_q \leq n$. Each student must rank a minimum number n_q of project topics (e.g., in our application $n_q = 7$). This requirement is introduced to increase the chances that an allocation exists, in which all students are assigned to a topic from their preference list. Students belonging to a group express their preferences jointly. Hence, for each group $G_k \in \mathscr{G}$ we require $R(G_k) = R(s_1) = R(s_2)$ for $s_1, s_2 \in G_k$. Hence, we could restrict our formal description to only regard groups and discard the view on students. However, for the sake of making notation easier later we maintain also the student view and use it interchangeably with the group view as most convenient. Further, we indicate by $L_k \subseteq P$ for the group G_k the set of project teams constructed by removing from P the project teams belonging to topics in $\mathscr{P}$ not present in the preference set $R(G_k)$.

Let the partial function $\sigma : \mathscr{G} \to P$ be a *student-project assignment* (or matching) with $\sigma(G_k) = p_j$, if the students of G_k are assigned to project team p_j, and $\sigma(G_k)$ undefined, if the students of G_k are unassigned.[1] The task is to find an assignment σ such that: (i) students are assigned to exactly one project from their preference set, i.e., for all $G_k \in \mathscr{G}$, $\sigma(G_k)$ is defined and belongs to L_k; (iii) team bounds are satisfied, i.e., for all $p_j \in P$, either the team has no student $\sum_{G_k \in \mathscr{G}:\sigma(G_k)=p_j} a_k = 0$ or $l_j \leq \sum_{G_k \in \mathscr{G}:\sigma(G_k)=p_j} a_k \leq u_j$; and (v) the quality, defined by some measure, is optimized. Constraints (ii) and (iv), mentioned in the introduction, allocating students of a group in the same team and restricting the number of teams per topic to be at most a given number are always satisfied by the definition of the set P and of the partial function σ. With respect to an assignment σ, we regard a project team as *open* if the number of students assigned to it is greater than 0, and *closed* otherwise.

Computational complexity Without group registrations and lower bounds on the number of students per team the problem of determining whether an assignment exists that matches all students and satisfies constraints (i)–(iv) can be formulated as a maximum flow problem and it is therefore polynomially solvable. The problem remains solvable by maximum flow also in the case of lower bounds but not allowing teams to remain closed (that is, all teams defined must receive a number of students between their lower and upper bounds, the case of zero students is not allowed, unless the lower bound itself is set to zero). The variant that asks for the minimum weight assignment under these conditions and the introduction of a

[1] In the student view, the partial function σ will be from S to P and it will indicate the project to which a single student $s_q \in S$ is assigned or it will be undefined if s_q is not assigned to any project.

weight for each pair of student and project team is still polynomially solvable by minimum cost flow algorithms. With group registrations and restrictions on acceptable projects but without lower bounds the existence problem is NP-complete. Indeed, this problem can be formalized as a matching problem: given a bipartite graph $B = (U, V, E \subset \{U \times V\})$, where each vertex of U has a size and each vertex of V has an integer capacity, determine the existence of a matching that matches all vertices of U while respecting the capacity restrictions on the vertices of V. Biró and McDermid (2014) show that, when all vertices in V have the same capacity, this matching problem corresponds to the decision version of the multiprocessor scheduling problem with eligibility constraints. Consequently, they conclude that the matching problem is NP-complete, already when all vertices in U have size 1 or 2 and each vertex in V has capacity 2. Arulselvan et al. (2016) have recently shown that the existence problem remains NP-complete even without group registrations but with lower bounds on the number of students per team and with closed teams allowed. More precisely, they have shown that the optimization problem asking to maximize the total weight of the assignment can be solved efficiently if the number of applicant students for every team is at most 2, while it becomes NP-hard as soon there are teams with at least 3 applicant students and students with preference list of at least 2 project teams or, alternatively, as soon as there are projects with upper bound of at least 3.

Handling preferences Preference sets can be transformed into a *value matrix* $V \in \mathbb{N}_0^{|S|\times|P|}$, where each element v_{qj} represents how student $s_q \in S$ ranks the topic of project team $p_j \in P$. The value 1 is set for the topic that is ranked best and r for the topic that is ranked in the rth position, for $1 \leq r \leq n_q$. If a project team p_j belongs to a topic that is not in the preference set of the student s_q, v_{qj} is set to be zero. The matrix V can be defined in the same way also for groups: i.e. $V \in \mathbb{N}_0^{|\mathscr{G}|\times|P|}$. In this case, we will denote its elements as v_{kj} for $G_k \in \mathscr{G}$, $p_j \in P$.

The quality of an assignment σ that satisfies all constraints is determined by a vector $\mathbf{v} = (v_{1,\sigma(1)}, \ldots, v_{M,\sigma(M)})$, $v_{q,\sigma(q)} > 0, \forall s_q \in S$, and by the distribution of students over ranks $\boldsymbol{\delta} = (\delta_1, \ldots, \delta_\Delta)$, where $\Delta = \max\{n_q \mid s_q \in S\}$ and $\delta_h = |\{s_q \in S \mid v_{q,\sigma(q)} = h\}|$ for $h = 1 \ldots \Delta$. In the decision-making process of an administering committee that has to solve the allocation problem under limited resources, the focus will be on the *collective welfare* (Moulin 2003). From this viewpoint, the interest is clearly only on assignments that are Pareto *efficient* with respect to their profile vectors $\mathbf{v}$. Unfortunately, this definition gives only a partial ordering of the feasible assignments. The two most widespread ways to aggregate a profile of preference relations into a collective preference relation with the weak ordering property are the *classical utilitarian ordering* and the *egalitarian ordering* (Moulin 2003). For two feasible assignments σ_1 and σ_2, the former assigns a weight to each value, $\omega : [1 \ldots \Delta] \to \mathbb{R}^+$, and compares $\sum_{s_q} \omega(v_{q,\sigma_1(q)})$ with $\sum_{s_q} \omega(v_{q,\sigma_2(q)})$, and the latter uses the *leximin order*, which consists in reordering the two vectors $\mathbf{v}^1$ and $\mathbf{v}^2$ by increasing coordinates and comparing them lexicographically (here to be understood in the less standard way from right to left defined formally in Sect. 5.3). Both relations can be redefined equivalently in terms of the corresponding distributions $\boldsymbol{\delta}(\sigma_1)$ and $\boldsymbol{\delta}(\sigma_2)$. In this case the weighted value order considers $\sum_{h=1\ldots\Delta} \omega(h)\delta_h(\sigma_1)$ and $\sum_{h=1\ldots\Delta} \omega(h)\delta_h(\sigma_2)$ and the leximin order considers the lexicographic ordering of $\boldsymbol{\delta}(\sigma_1)$ and $\boldsymbol{\delta}(\sigma_2)$.

The classical utilitarianism may lead to situations where the overall satisfaction is high, but at the cost of putting few students at a disadvantage. For example if only two assignments are feasible with the value vectors $\mathbf{v}_1 = (1, 2, 2, 2)$ and $\mathbf{v}_2 = (1, 1, 1, 3)$, the weighted value order with weights $\omega(i) = i$ for all $i = 1, 2, 3$ would prefer the second assignment. However, in the second assignment the fourth student is particularly disadvantaged to the benefit of the second and third student. The *individual welfare* criterion aims at ensuring that the student with the worst preference receives the best possible preference. In other terms, we search for the assignment that minimizes the worst value, that is, $\min \max\{v_{q,\sigma(q)} \mid s_q \in S\}$. This is also known as the *minimax* criterion (Yager 1996) (note that in the literature it is often referred to as maximin criterion in that satisfaction improves as values get larger, contrary to our definition of **v**, where instead low values are best). The minimax criterion promotes outcomes with a certain degree of equality and hence it can be used as a possible definition of *fairness*.

On the other hand, the minimax criterion alone makes no use of additional information to decide among assignments with the same guarantees on the maximum values in the vector **v**. For example, the two vectors $(1, 2, 2, 3)$ and $(1, 3, 3, 3)$ are not distinguishable under the individual welfare criterion, while the first one is clearly preferable to the second. The leximin order introduced above subsumes the minimax criterion and overcomes its drawback with respect to collective welfare. It constitutes therefore an interesting approach to our problem. An alternative approach that tries to conciliate both fairness and collective welfare is to first optimize according to the minimax criterion and then, restricted to only considering minimax optimal solutions, optimize collective welfare using the weighted value order.

3 Related work

Student-project allocation is a fair division and resource allocation problem. Different aspects for this class of problems have been studied by researchers in different fields. In welfare economics and social choice theory (Moulin 2003), evidence and analysis suggest that the allocation process should be automated by means of a centralized matching scheme (Roth 1984). The focus has then been on the definition and study of mechanisms that satisfy certain properties, such as, for example, strategy-proofness, that is, the guarantee that participants may not benefit by declaring false preferences. Criteria such as collective welfare and egalitarianism are most often formalized by weighted sum [aka Bentham utilitarian function (Young 1995)] and maximin [aka Rawlsian function (Rawls 1971)], respectively. Other metrics for fairness, beside minimum received utility, have been studied in the literature. They include: the difference between the maximum and the minimum received utility, the standard deviation or the coefficient of variation of received utilities, the Jain index, the Theil index, the mean log deviation of utilities and the Gini coefficient [see (Tempkin 1993) for these and more]. A more general approach to aggregate preference values in a context of Pareto optimality is given by ordered weighted averaging (OWA) functions (Yager 1996, 1997). Weighted value and leximin arise as particular cases of OWA functions. In this context, leximin optimization is known as *conditional means procedure*.

Contributions to the definition of criteria are to be found also in artificial intelligence. A criterion studied in this field for the allocation of indivisible objects in multi-agent systems is *envy-freeness* (Bouveret and Lang 2008). Recently, the field of operations research has looked closer at the trade-off between equity and utilitarianism (Hooker and Williams 2012; Bertsimas et al. 2012). Bertsimas et al. (2012) propose a family of social welfare functions parametrized by a single parameter, $\alpha \geq 0$, that measures the aversion of the system designer towards inequalities, thus achieving α-fairness (Atkinson 1970). Beside reducing the problem of designing an appropriate objective to the selection of a single parameter, Bertsimas et al. (2012) give quantitative information on the percentage loss in global utility (price of equity) and in minimum utility (price of utilitarianism) as a function of the parameter α and the number of participants in the division of resources. This information can be used by the decision maker to decide a value for α thus resolving a priori the trade-off between equity and utilitarianism. We note however that for a fixed α the objective function is nonlinear, which complicates the solution of the resource allocation problem from the combinatorial viewpoint.

Hooker and Williams (2012) attempt to combine equity and utilitarianism in a way that has intuitive meaning and therefore, supposedly, easy to control by the decision maker. They propose a scheme in which each participant contributes to the welfare function in an utilitarian way, if its utility under the resource allocation is more than a threshold τ from the one of the worst off participant, and with the worst utility among all participants, otherwise. Although the scheme in its initial form is non-convex and not representable in mixed integer linear programming, Hooker and Williams (2012) show that it can be linearized and used successfully in MILP formulations.

Several cases of student-project allocation problems have been studied in the literature. We found articles discussing how the problem has been tackled in practice at the universities of Southampton, Singapore, York and Hohenheim. To find allocations these articles use different combinatorial optimization techniques from the fields of operations research and artificial intelligence, namely, integer programming (Proll 1972; Anwar and Bahaj 2003), evolutionary algorithms (Harper et al. 2005; Srinivasan and Rachmawati 2008), constraint programming (Dye 2001) and local search (Geiger and Wenger 2010).

Proll (1972) considers a one-to-one assignment, hence without group registrations (or with equal-size groups formed a priori) and without lower bounds on team size. He tackles the problem by a two phase weighted assignment MILP formulation where first a minimax objective function is optimized and then, subject to the constraint on the worst off student imposed by the minimax solution, a weighted sum of preferences is optimized. The instances have up to 50 students and about the same number of projects. Empirical results showed that with preference lists constrained to be of length greater than 5, students could always be assigned to a project they ranked and about 50% were assigned to their first choice.

Anwar and Bahaj (2003) study two MILP models. The first is a one-to-one assignment for which they also propose a two phase approach by first minimizing the largest number of projects supervised by a single lecturer (a minimax criterion to balance the work load among lectures) and then finding among the solutions of the first phase those that optimize a weighted sum of preferences. The second model is a project assignment with lower and upper bounds on the team size, but without group registrations, that they solve by optimizing the weighted sum criterion. The first model is tested on the data made available by Teo and Ho (1998), comprising 372 students, 483 projects and preference list of length 10. In Teo and Ho (1998), the best solution returned after 24 h by a repeated randomized greedy algorithm, which is very similar to our lottery described in Sect. 4, left 45 students unassigned. Anwar and Bahaj (2003) solve instead their MILP model in a few minutes and obtain an optimal solution without unassigned students. However, group registrations, that may make the problem harder to solve, are not considered. Moreover, there is no comparison with alternative goals, such as, for example, stability, envy-freeness and leximin.

Harper et al. (2005) address a simpler student-project allocation problem than ours (without project topics, group registrations, and minimal capacity constraints on teams), but with the same objective of achieving fair assignments. They use an evolutionary algorithm to solve the formulation of the problem with weighted sum with the motivation that the multiple solutions returned by the algorithm can be used to address the issue of fairness, that is, the decision maker has ultimately to choose among the returned solutions the most fair.

Srinivasan and Rachmawati (2008) also use evolutionary algorithms to solve a similar problem. They emphasize the multiobjective nature of the problem where objectives of collective welfare, and load balancing among lecturers, departments and student career are in trade-off. Then they put forward a way to capture decision makers' preferences in an aggregated objective by means of fuzzy logic. The evolutionary algorithm finds assignments that optimize the aggregated objective. They conduct tests on instances with about 600 students and 600 projects.

In the case addressed by Geiger and Wenger (2010), a number of students is assigned to given scientific topics subject to lower and upper bound capacity constraints on the topics and a weighted sum of students' preferences is then optimized. In this work, students are given a budget of "points", and express their preferences by assigning these points on the topics. Ties in the preferences of the students are therefore allowed. Although this method captures the size of relative preferences between the projects and provides easy access to an utility function for every agent, this form of cardinal measure of utility is criticized in social choice theory literature. There seems to be experimental evidence that the representation of preferences over uncertain outcomes by utility functions is inadequate due to the limited rationality of the agents (Kagel and Roth 1997) and to the different perception of the value of points among participants (Moulin 2003). The authors develop a local search algorithm that uses several neighborhood definitions with the aim of identifying a set of good (but possibly suboptimal) solutions and use them in a multi-objective setting similarly to what proposed by Harper et al. (2005).

In computer science, the focus has been on complexity and approximability analysis as well as algorithm design. Abraham et al. (2007) and Manlove and O'Malley (2008) look at the problem as a two-sided matching problem in which beside students, also teachers express preferences about which students they would like to receive. The problem can thus be seen as a generalization of the admission of students to colleges, the assignment of residents to hospitals and the assignment of students to dormitories, which are generalization of the stable matching problem (Gusfield and Irving 1989; Iwama and Miyazaki 2008; Perach et al. 2008; Manlove 2013, Sect. 5.5). In this context, a relevant property of assignments is their *stability*, that is, the guarantee that no two participants who are not matched together in the solution would rather be matched to one another than remain with their assignment in the solution. Such a pair would form private agreements and undermine the integrity of the central solution. Group registrations and lower bound capacity constraints are not present in the problems from Abraham et al. (2007), Manlove and O'Malley (2008). Thus, it is shown that linear time algorithms are possible for finding stable allocations that are student-optimal and lecturer-optimal, that is, each student (lecturer) is assigned to the best project (student) that the student (lecturer) would obtain in any stable assignment. These authors also show that with the definition of stability that they consider, all stable matchings have the same number of students unassigned. On the other hand, it is in general NP-complete to determine whether a stable matching exists, given an instance of the problem with lower and upper bounds for the teams (Biró et al. 2010).

Dye (2001) in 2001 uses constraint programming to solve the student-project assignment formulated as a stable marriage problem with load balancing. His results show that at that time the approach did not scale well and it was not practicable for problems with more than 20 students and 20 projects. In an another version of the problem with two-sided preference lists, lecturers may supply preferences over the projects that they offer rather than over students (Manlove and O'Malley 2008). The new definition of stability in this problem looses the property that all stable matchings have the same number of unassigned students and minimizing this number can be shown to be NP-hard. Thus, the focus from an algorithmic perspective has shifted to approximation results (Iwama et al. 2012). Finally, El-Atta and Moussa (2009), consider the case where lecturers have preferences over (student, project) pairs under the motivation that the model is more flexible and this type of preferences may help to minimize the number of unassigned students. In our specific application at SDU, preferences of lecturers for students are perceived as undesirable. Indeed, lecturers may prefer students on the basis of their previous merits, which is deemed unfair, or choose only students that are enrolled in the curriculum of study of their field thus being an obstacle to interdisciplinarity, which is one of the declared goals of the course. Similarly, preferences of lecturers for project topics are unnecessary since each lecturer is a priori assigned to the topics offering only projects on topics he/she decided.

A similar problem that arises in the context of conference management systems is the paper-reviewer assignment problem. It does not have group registrations but it includes minimum and maximum capacity constraints on both sides of the matching. On the side of the reviewer, bounds are needed to ensure that the load is balanced and on the side of the paper, to guarantee that each paper is reviewed a sufficient number of times. The fact that a reviewer is assigned a number of papers, differently from the students that get only one project, complicates the assessment of assignments, because each element of the vector value $\mathbf{v}$ becomes now itself a vector. Garg et al. (2010), discuss quality issues and algorithmic aspects of this problem. They define the minimax and leximin criterion based on either lexicographic order or weighted value order of the vector values that belong to each reviewer. They show that for both cases, when referees can express preferences by means of only two different values (say, $\{1, 2\}$), then there exists a polynomial time algorithm, while, when the number of different values is larger than two, the problem is NP-hard. For this latter case they give an approximation algorithm for the general case based on rounding a fractional assignment produced by linear programming via max-flow computations. They report numerical results on one instance with 14 reviewers, 202 papers and 3 different values to express preferences. They focus their final comments on the comparison of minimax against leximin under weighted approaches. There is no definitive conclusion in their comparison and it is not possible to assess the distance from optimal solution for each of the criteria since exact MILP approaches are not investigated.

In conclusion, we found only the article by Anwar and Bahaj (2003) reporting computational results on a problem close to ours. They show that a MILP approach is viable to obtain optimal solutions under a minimax and weighted utility criterion. Our problem is complicated by group registrations and hence the viability of MILP has to be reassessed. Moreover, we found a wide variety of alternative criteria that seem suitable in our context but whose applicability both in terms of computational cost and practical results has not been assessed before. We set out to determine which of these alternative criteria, or which combinations thereof, lead to models that are computationally satisfactory in our context.

4 A lottery solution

In the 2008 manual of the course[2] the registration and assignment procedure is presented to the students as follows.

[2] Birgitte H. Kallipolitis, Marianne Holmer, Paul C. Stein, Per Lyngs Hansen, Rolf Fagerberg and Søren Sten Hansen. Naturvidenskabeligt Projekt. Målsætning & Krav. 2008. Course document at the Faculty of Natural Science, University of Southern Denmark.

Data: P set of project teams; $\mathscr{P}$ set of project topics; $\mathscr{G}$ set of groups; $R(G_k)$, $G_k \in \mathscr{G}$ priority lists;
Result: A student-project assignment σ described as a collection of sets of groups $X(p_j)$, $p_j \in P$;
1 $h := 1$;
2 **while** $\mathscr{G}$ *not empty* **do**
3   **for** P_i *in* $\mathscr{P}$ **do**
4   Let $C(P_i) = \{G_k \in \mathscr{G} \mid P^{(h)} \in R(G_k),\ P^{(h)} = P_i\}$, i.e., the set of student groups with P_i in hth position of their priority list
5   **for** P_i *in* $\mathscr{P}$ *such that* $C(P_i)$ *is not empty* **do**
6   shuffle the groups in $C(P_i)$ at random
7   **for** G_k *in* $C(P_i)$ **do**
8   **for** p_j *in* P_i **do**
9   **if** $\sum_{G_l \in X(p_j)} a_l + a_k \le u_j$ **then**
10                     $X(p_j) = X(p_j) \cup \{G_k\}$;
11                     $\mathscr{G} = \mathscr{G} \setminus G_k$;
12   break;
13   **if** $G_k \in \mathscr{G}$ *and* $h = |R(G_k)|$ **then**
14                 $\mathscr{G} = \mathscr{G} \setminus G_k$;
15     $h := h + 1$;

Algorithm 1: Pseudocode for the lottery procedure

> [...] The registration takes place declaring which project topics you would like to be engaged in, sorted in a priority list. You are asked to prioritize at least 7 project topics and you may well express more. [...] If you choose to register together with other students, you have to agree on a common priority list of project topics. [...]
> After the end of the registration period the project topics will be distributed according to your priority list. The assignment occurs by selecting students at random from all those with the same first priority. If your first priority is filled up before you are chosen, you will be put in the lottery for the second priority until there are not anymore places either in this topic. In this way, everybody will have the same chance to have fulfilled a certain priority.
> If you register together with some colleague, you will stay together in the lottery and you will be assigned to the best priority where there are free places left for all members of the group. Therefore, if you register together with others you have to be aware that this can have an influence on which project topic you will be assigned to.

Algorithm 1 formalizes the allocation procedure in pseudo-code. The algorithm takes as input data the project teams and the preference lists of the groups and outputs a student project assignment σ defined by a collection of sets $X(p)$, $p \in P$, representing the groups of students assigned to each project team p. For a priority h, starting from the first priority, $h = 1$, a set $C(P_i)$ is created for each project topic P_i, $i = 1, \ldots, n$, made by remaining unassigned groups that had P_i in the hth position of their priority queue (lines 3–4). The sets $C(P_i)$ are then shuffled at random (line 6) and group assignments to project teams are proposed in the obtained order (lines 8–12). A group is assigned to a project team p_j of topic P_i if there is still room for the group, given the maximum capacity of the team and the number of students currently assigned to it. A group is removed from the set $\mathscr{G}$ if it is assigned to some $X(p_j)$ (lines 10–11) or if all its priorities have been considered and the group is still unassigned (lines 13–14).

The while loop on line 2 will run at most Δ times, with $\Delta = \max\{|R(G)| \text{ s.t. } G \in \mathscr{G}\}$, and the loops at lines 5 and 7 produce $m = |\mathscr{G}|$ iterations each time, in the worst case that $\mathscr{G}$ does not decrease in size. Hence the running time of the algorithm is $O(\Delta m T)$, where $T = \max\{t_j \mid p_j \in P\}$. From the point of view of the assignment, the algorithm can terminate with:

– teams with less students assigned than their lower bounds, i.e., there might be $p \in P$ for which $\sum_{G_k \in X(p)} a_k < l_j$;
– groups unassigned, i.e., there might be $G \in \mathscr{G}$ discarded on lines 13–14 after missing the chance of being assigned in all $R(G)$ lotteries in which G took part.

The execution of the algorithm on the instances of our interest is very fast, therefore, similarly to what done by Teo and Ho (1998), we use a *repeated lottery* consisting in running the algorithm above for a number of times and selecting among the assignment produced by the runs the one that minimizes the number of unassigned students breaking ties with the worst preference used.

Although this procedure is easy to present to the students and has the lottery mechanism as guarantee for fairness, it makes a bad use of resources. Table 1 reports the performance of the lottery algorithm on our nine real-life instances. There is always a number of students left unassigned. In addition there are teams that have less students than their lower bounds and that therefore cannot be effectively created, thus, implying yet more unassigned students. Moreover, although the number of students assigned to their first priority is high, this is achieved at the disadvantage of a few students who are assigned very low in their priority list: there are students assigned to projects far beyond the seventh position. This algorithm and its assignments were in use until 2010. The students left unassigned were called by the administering committee for rediscussing their priority list and finding together a placement among the undersized teams. This process was very demanding and complicated. In the year 2011, the committee was unable to find a solution for the 20 students left and eagerly welcomed our MILP solutions.

5 Mixed integer linear programming formulations

In this section, we first formulate the set of feasible student-project assignments using a combination of linear constraints. Then, we present alternative linear objective functions to promote collective welfare.

Let x_{kj} be a 0–1 variable equal to 1 if group $G_k \in \mathscr{G}$ is assigned to project team $p_j \in L_k$. Recall that $L_k \subseteq P$ is the set of project teams belonging to topics present in the preference set of group G_k, a_k is the number of students in group G_k, v_{kj} is the preference value given by group k to project j. Let y_j be a 0–1 variable equal to 1 if project team $p_j \in P$ has any group assigned to it. We denote by $\mathscr{X}$ the set of feasible assignments that we define as follows:

$$\mathscr{X} = \Bigg\{ \sum_{p_j \in L_k} x_{kj} = 1, \quad \forall G_k \in \mathscr{G} \tag{1}$$

$$\sum_{G_k \in \mathscr{G}\,:\,p_j \in L_k} a_k x_{kj} \leq u_j y_j, \quad \forall p_j \in P \tag{2}$$

$$\sum_{G_k \in \mathscr{G}\,:\,p_j \in L_k} a_k x_{kj} \geq l_j y_j, \quad \forall p_j \in P \tag{3}$$

Table 1 The distribution of students over priorities for the best assignments in 200 runs of the lottery algorithm

Instance					Students per priority														Unass. stds	Unders. teams	Seconds
Year	$\|S\|$	$\|\mathscr{G}\|$	$\|P\|$	$\|\mathscr{P}\|$	14	13	12	11	10	9	8	7	6	5	4	3	2	1			
2008	200	173	70	52	0	0	0	0	0	0	1	4	10	7	9	7	25	135	2	3	4.37
2009	129	107	48	45	0	0	0	0	0	0	0	0	0	3	6	1	11	108	0	0	2.18
2010	193	158	62	52	0	0	2	1	0	0	0	0	3	4	9	1	21	150	2	0	4.12
2011	259	219	83	69	2	0	5	0	1	2	2	7	5	6	11	5	31	170	12	1	6.45
2012	300	247	102	81	0	0	0	2	0	3	5	6	9	11	10	16	34	186	18	3	7.96
2013	355	293	114	95	0	0	0	0	3	1	3	3	8	8	11	18	46	226	28	6	10.68
2014	371	290	131	98	0	0	0	1	0	0	5	9	6	12	11	19	37	241	30	5	11.17
2015	422	422	150	96	0	0	0	0	0	1	3	6	6	13	10	23	55	298	7	7	15.96
2016	416	416	160	100	0	0	0	0	0	1	1	7	2	10	6	22	29	332	6	10	14.82

$$\left.\begin{aligned} x_{kj} &\in \{0,1\}, \quad \forall G_k \in \mathscr{G}, \forall p_j \in L_k \quad (4)\\ y_j &\in \{0,1\}, \quad \forall p_j \in P \end{aligned}\right\}. \quad (5)$$

Constraints (1) ensure that each group is assigned to exactly one project under the topics in its preference list L_k. Constraints (2) and (3) impose that the number of students assigned to project p_j is within the given lower and upper bound whenever $y_j = 1$. If the project team is not open ($y_j = 0$), then no group can be assigned to project p_j.

As mentioned in Sect. 2, finding assignments in $\mathscr{X}$ with the best individual values amounts to solve a multi-criteria optimization problem in the Pareto sense, which leads to an incomplete ordering and hence multiple incomparable solutions. As in previous works, we induce a complete weak ordering by introducing an aggregation operator that provides a unique numeric evaluation for each assignment $\sigma \in \mathscr{X}$, that is:

$$F(\sigma) = H(v_{1,\sigma(1)}, \ldots, v_{M,\sigma(M)}).$$

5.1 Classical utilitarian ordering

In the classical utilitarian ordering, the function $F(\sigma)$ is a weighted sum of the preferences of students (or groups) over their assignments to projects. For $\omega : \mathbb{N}_0 \to \mathbb{R}$, it is

$$F(\sigma) = \sum_{q=1}^{M} \omega(v_{q,\sigma(q)}).$$

Let R_h be the set of group-project pairs that yields the preference value h, that is, $R_h = \{(k, j) \in \mathscr{G} \times L_k \mid v_{kj} = h\}$, and $w_h = \omega(h)$. We can then write the first MILP model for the student-project allocation problem:

$$\min \sum_{h\in\{1,\ldots,\Delta\}} w_h \sum_{(k,j)\in R_h} a_k x_{kj} \quad (6)$$

$$\text{s.t.} \quad x_{kj} \in \mathscr{X}. \quad (7)$$

The inner summation in (6) gives the total number of students that received a project with preference value h.

The simplest weight scheme consists in setting each preference with a weight equal to its value (*identity scheme*):

$$w_h = h, \quad \forall h \in \{1, \ldots, \Delta\}. \quad (8)$$

Minimizing the objective function (6) with these weights is equivalent to minimizing the overall sum of preference values received by the students.

When all groups are of size one, our model (6)–(8) is equivalent to the model studied by Anwar and Bahaj (2003). Although in that work constraints (2)–(3) are formulated with the big-M method (Williams 2013), which, depending on the data, could make the formulation less tight than ours, in practice, we do not expect to experience substantial differences with state-of-the-art solvers.

A different weight scheme is the following (*exponential scheme*):

$$w_h = -2^{\max\{8-h,0\}}, \quad \forall h \in \{1, \ldots, \Delta\}. \quad (9)$$

The rationale behind the scheme (9) is that the perceived benefit of a student is higher for equal differences at the high levels of preferences rather than at the low levels. In other terms, if in an assignment a student s_1 has a better preference value than a student s_2, a worsening in the value of s_1 should yield a higher welfare decrease compared to a worsening in the value of student s_2. Note that with the scheme (8) the welfare decrease for the two students would be the same. For example, if a student s_1 worsens its value from 1 to 5 and a student s_2 worsens from 2 to 6 the decrease in welfare function is 1 for both students under the scheme (8) and $-2^6+2^7=64$ for s_1 and $-2^2+2^3=4$ for s_2 under scheme (9). The second scheme is therefore more aggressive towards high preferences although this might imply that under most measures of fairness the resulting allocation will be less fair. The value 8 in (9) is tuned for our case that assumes $\Delta \geq 7$.

5.2 Minimax

A simple idea of fairness is to guarantee that the student that receives the worst preference is assigned to a project with preference as high as possible. That is, we minimize the maximum preference value attained by any student. Formally, this leads to an alternative way of associating assignments with a single numerical value:

$$F(\sigma) = \max\{v_{q,\sigma(q)} \mid s_q \in S\}.$$

The optimal assignment σ^* is then the one with $F(\sigma^*) = \min\{F(\sigma) \mid \sigma \in \mathscr{X}\}$.

In MILP, this idea is formulated using an additional variable z and a number of new linear constraints:

$$\min \quad z \tag{10}$$

$$\text{s.t.} \quad z \geq \sum_{p_j \in L_k} v_{kj} x_{kj} \quad \forall\ G_k \in \mathscr{G} \tag{11}$$

$$x_{kj} \in \mathscr{X}. \tag{12}$$

The objective function (10) together with constraints (11) force the variable z to take the highest preference value of the group-project assignments. As mentioned, the drawback of this minimax criterion is that it does not consider preferences apart from the worst.

5.3 Profile-based optimization

Although a complete weak ordering, the max function leaves many ties among alternative assignments, because it only orders assignments with respect to their worst values. The leximin order provides a refinement of the max function ordering by accounting for both equity and utilitarianism. In this case, $F(\sigma)$ is not defined as a unique number but as an M-tuple, $\mathbf{F}(\sigma) = (f_1, f_2, \ldots, f_M)$, where f_i is the ith smallest of the elements of $\mathbf{v}$. Thus, $f_1 \leq f_2 \leq \cdots \leq f_M$ where $f_i = v_{q,\sigma(q)}$ for some $s_q \in S$. Using this definition of $\mathbf{F}$ we can define a *leximin ordering* $\preceq_{\text{leximin}}$ as follows:

$$\begin{aligned} \sigma_1 <_{\text{leximin}} \sigma_2 \quad & \text{iff } \exists\, \ell \geq 1 \text{ such that} \quad && \text{for all } i > \ell,\ f_i(\sigma_1) = f_i(\sigma_2), \\ & && \text{and } f_\ell(\sigma_1) < f_\ell(\sigma_2) \\ \sigma_1 =_{\text{leximin}} \sigma_2 \quad & \text{iff for all } i = 1, \ldots, m,\ f_i(\sigma_1) = f_i(\sigma_2) && \end{aligned} \tag{13}$$

In words, this ordering compares two assignments with respect to their worst value f_M, if they are not equal, then it indicates the assignment with the lowest value as being preferred. If they are equal, then it looks at the second worst value f_{M-1} and repeats the process until one of the assignment is preferred to the other or all students have the same value in which case the assignments are tied. An optimal assignment σ^* will then be one for which the relation $\sigma^* \preceq_{\text{leximin}} \sigma'$ holds for all $\sigma' \in \mathscr{X}$.

The leximin ordering defined in (13) can be legitimately defined differently, if instead of starting the comparison from the worst value f_M, we start it from the best value f_1. These two alternative definitions lead to different Pareto efficient solutions and might therefore both be interesting. Below we describe how these two types of alternative optimal solutions can be found. We call the first *generous maximum matchings* (or maximum cardinality, rank maximal matching) and the second *greedy maximum matchings* as done in Manlove (2013, Sect. 1.5.6). First, we reformulate the leximin order with the distribution approach. Then, we consider the generous maximal matchings arising from Definition 13 and present two approaches to find them: one solving a series of optimization problems and the other using the ordered weighted averaging approach. Finally, we discuss briefly how to find greedy maximum matchings.

Distribution approach While the leximin method strengthens the ordering with respect to the max order, it makes finding the optimal assignment more complicated since we do not have anymore a single number associated to an assignment but at each comparison between assignments we need to sort the values of the vectors $\mathbf{F}$ and then proceed with M pairwise comparisons of their elements.

It can be shown that the optimal solution in the leximin ordering can also be found by the *distribution approach* (see, e.g., Ogryczak et al. 2005). Here, $F(\sigma)$ is a $\varDelta$-tuple, $\mathbf{F}'(\sigma) = (\delta_1, \delta_2, \ldots, \delta_\varDelta)$ where δ_h is the number of students that received the hth smallest of the values in $\mathbf{v}$. The leximin ordering $\leq_{\text{leximin}}$ is then defined as

$$\begin{aligned}
\sigma_1 <_{\text{leximin}} \sigma_2 \;\; &\text{iff} \;\; \exists \ell \geq 1, \;\text{such that} \quad \text{for all } h > \ell, \delta_h(\sigma_1) = \delta_h(\sigma_2), \\
& \qquad\qquad\qquad\qquad\qquad \text{and } \delta_\ell(\sigma_1) < \delta_\ell(\sigma_2) \\
\sigma_1 =_{\text{leximin}} \sigma_2 \;\; &\text{iff} \;\; \text{for all } h = 1 \ldots \varDelta, \; \delta_h(\sigma_1) = \delta_h(\sigma_2)
\end{aligned}$$

Note that with this definition the elements of $\mathbf{F}'(\sigma)$ do not need to be reordered and hence a nonlinear aspect of the aggregation is removed.

Generous maximum matchings Using the distribution approach, a procedure to find an optimal assignment in the leximin order is the following: first, minimize the number of students that have received their last preference; second, minimize the number of students that have received their second-last preference; then, minimize the number of students that have received the third-last preference, and so forth until the second preference.

To represent the distributions of students over preferences we introduce the variable r_h, with $h = \{1, \ldots, \varDelta\}$, that represents the number of students that under the assignment received a project that they valued h. In order to find the minimal feasible values for r_h, for $h = \{1, \ldots, \varDelta\}$, we solve a sequence of MILP optimization problems for decreasing values of h starting from $\varDelta$. Each optimization problem exploits the fact that we have already solved the problem for all $d > h$, and therefore, we have already determined the corresponding values r_d. Hence, for h from $\varDelta$ to 2 the optimization problem to solve is the following:

$$r_h = \min \sum_{(k,j) \in R_h} a_k x_{kj} \tag{14}$$

$$\text{s.t.} \quad r_d = \sum_{(k,j) \in R_d} a_k x_{kj}, \quad \forall d \in \{(h+1), \ldots, \varDelta\} \tag{15}$$

$$x_{kj} \in \mathscr{X}. \tag{16}$$

The objective function (14) minimizes the number of groups that are assigned to projects that have rank h. When the problem is solved with $h = \varDelta$, constraints (15) vanish. When $h < \varDelta$, the term r_d in constraints (15) is fixed, since it was determined when solving the problem for higher values of h.

This lexicographic procedure requires the solution of $\Delta - 1$ problems (14)–(16) that differ from each other by a single constraint of type (15). We can also stop the procedure earlier, if $\sum_{d\in\{h,\ldots,\Delta\}} r_d = |S|$ but this is never the case in our instances (fortunately, because such assignments have low global utility). Note that if we solve problem (14)–(16) starting with $h = \Delta$ and we stop as soon as a solution with $r_h > 0$ exists, we have solved a *minimax* version of the problem: i.e., we have minimized the maximum preference attained by any group, as in Sect. 5.2.

Ordered weighted averaging If the number of preferences Δ is very large and each instance of problem (14)–(16) is computationally demanding, the lexicographic distribution procedure described above may be a non-viable solution approach. As an alternative, Yager (1997) formulated the leximin ordering as a particular ordered weighted averaging (OWA) function. Applied to our case, this formulation yields:

$$F(\sigma) = \sum_{i=1}^{M} w_i \bar{f}_i,$$

where $\bar{f}_i$ is the ith smallest value of $v_{q,\sigma(q)}$, $s_q \in S$, normalized to be in the interval [0, 1], and the weights w_i are defined as:

$$\begin{aligned} w_1 &= \tfrac{\beta^{M-1}}{(1+\beta)^{M-1}} \\ w_h &= \tfrac{\beta^{M-h}}{(1+\beta)^{M+1-h}} \quad \forall h \in \{2, \ldots, M\} \end{aligned} \tag{17}$$

where β is "infinitesimally smaller than the smallest distinction that can be perceived between the values being aggregated" (i.e. the $\bar{f}_i$ terms) (Yager 1997). We normalize $v_{q,\sigma(q)}$ by dividing it by Δ. Hence, the smallest possible distinction among the normalized values becomes $1/\Delta$. Consequently, we set $\beta = 1/\Delta - 0.001$.

As an example, for $M = 10$ and $\Delta = 8$ we set $\beta = 0.125 - 0.001$ and weights

$$\begin{aligned}(0.00002; 0.00020; 0.00177; 0.01604; 0.14538; 1.31782;\\ 11.94538; 108.27905; 981.49719; 8896.79715)\end{aligned}$$

after rescaling by 10,000. Then for $\mathbf{v}_1 = (1, 2, 2, 3, 4, 5, 5, 6, 7, 8)$, we have $F(\sigma_1) = 9845.2$ and for $\mathbf{v}_2 = (1, 2, 2, 3, 4, 4, 5, 6, 7, 8)$, $F(\sigma_2) = 9845.0$ and σ_2 is the preferred assignment.

In our instances $M \gg \Delta$ hence the proposed method would encounter numerical problems. Therefore, we changed the proposed OWA definition using the distribution approach as follows:

$$F(\sigma) = \sum_{h=1}^{\Delta} w_h \bar{f}_h \delta_h$$

with $\bar{f}_h = h/\Delta$, for $h = 1, 2, \ldots, \Delta$ and weights w_h derived in the same way as in scheme (17) but with the much smaller value of Δ in place of M.[3]

[3] In the implementation, to avoid numerical problems, we used the weighting scheme (17) until 8 even for instances with $\Delta > 8$. Then, for values of $h > 8$ we set $w_h = w_8 + 1$. It is not too hard to prove that using the distribution approach and Δ in the definition of weights still yields the leximin solution. It will suffice here to show that in our experimental results the assignments we found had indeed the same value vectors as those of the leximin solutions.

Hence, we can solve the leximin ordering approach by solving once problem (6)–(7) with the weighting scheme (17). In practice, we experienced that this optimization procedure was faster than the one solving a sequence of Δ optimization problems.

Note that the proposed weighting scheme, mimicking the leximin approach for generous maximal matchings, yields a rather different guidance criterion with respect to the exponential scheme (9). Indeed, here higher importance is set on differences at high preference values rather than at the low values.

Greedy maximum matchings These type of profile-based matchings brings to the extreme the idea underlying the exponential weighting scheme (9): that is, the importance of giving someone, for example, the first choice rather than the second is higher than the importance of giving someone the fourth choice rather than the fifth one. They are defined as the matchings that among all maximum (or complete) matchings, maximize the number of students that are assigned to their first priority and subject to this condition, maximize the number of students that are assigned to their second priority and so on. These assignments can be found with the same MILP-based procedure (14)–(16) described above, simply inverting the order in which h is considered and maximizing, instead of minimizing, the objective function (14). The drawback of this type of matchings with respect to the generous maximum matching approach is that they can end up assigning students to projects relatively low down in their preference list.

5.4 Combined utilitarian-minimax formulation

Hooker and Williams (2012) proposed an alternative policy to address the trade-off between equity and utilitarianism. In our context, this policy leads to a social welfare function that switches from the minimax criterion, that is problem (10)–(12), to an utilitarian criterion, that is problem (6)–(8), when pursuing fairness for the worst off student goes to the disadvantage of too many other students. More precisely, the contribution to the objective function of each student $s_q \in S$ is equal to the overall worst preference value, $\bar{v} = \max\{v_{q,\sigma(q)} \mid s_q \in S\}$, if $v_{q,\sigma(q)}$ differs from $\bar{v}$ less than a threshold τ, or his/her attained preference value, $v_{q,\sigma(q)}$, if the difference is greater than τ. Hooker and Williams motivate this policy with some indirect evidence, which they found in the literature, that people tend to regard as unrealistic to take the minimax principle to its extreme; that is, to continue with such a policy when it takes too many resources from others. Hence, the proposal to switch to an utilitarian objective in extreme circumstances.

Similarly to what done in Hooker and Williams (2012), the formulation of this policy for our problem yields:

$$\min \quad f \tag{18}$$

$$\text{s.t.} \quad f \geq (1-M)\tau + Mz + \sum_{G_k \in \mathscr{G}} a_k \min\{0, \tau - z + v_k\} \tag{19}$$

$$v_k = \sum_{p_j \in L_k} v_{kj} x_{kj} \quad \forall G_k \in \mathscr{G} \tag{20}$$

$$z \geq v_k \quad \forall G_k \in \mathscr{G} \tag{21}$$

$$x_{kj} \in \mathscr{X}. \tag{22}$$

Recall that M is the total number of students, and the variable z is used to express the quantity $z = \max\{v_{k\sigma(G_k)} \mid G_k \in \mathscr{G}\}$. Inequality (19) is constructed in such a way that each group of students G_k makes an utilitarian contribution of v_k, if v_k differs from z more than τ, and is otherwise represented by z. Indeed:

- If all groups have preference values that differ by less than τ from the worst off student, the summation in (19) vanishes and the objective becomes to minimize the quantity $(1-M)\tau + Mz$, which entails minimizing a (scaled) z. That is, it is like if all groups contributed by z.
- If there is one group G_h that is far worse than all the others in preference value and $v_h - v_k > \tau$ for all $G_k \in \mathscr{G}$, $h \neq k$, then (19) becomes (since we are minimizing z is set equal to v_h):

$$\begin{aligned} f &\geq (1-M)\tau + Mv_h + \sum_{G_k \in \mathscr{G} \setminus \{G_h\}} a_k(\tau - v_h + v_k) \\ &= (1-a_h)\tau + \sum_{G_k \in \mathscr{G}} a_k v_k, \end{aligned}$$

  and the minimization of f implies the minimization of the total utility with an offset (the τ term) that depends on which group is the worst off. This is because the students in that group do not receive utilitarian treatment.
- If only groups $G_k \in \mathscr{G}' \subset \mathscr{G}$ have preference values within τ of z the constraint (19) becomes:

$$f \geq (1-M)\tau + Mz + \sum_{G_k \in \mathscr{G} \setminus \mathscr{G}'} a_k(\tau - z + v_k)$$

$$f \geq \left(1 - \sum_{G_k \in \mathscr{G}'} a_k\right)\tau + \sum_{G_k \in \mathscr{G}'} a_k z + \sum_{G_k \in \mathscr{G} \setminus \mathscr{G}'} a_k v_k.$$

  Thus everyone within τ of the worst preference is identified with the least advantaged group weighted by the size of the group while the other groups contribute with an utilitarian component. The first term is again an offset that depends on the number of students in the worst off groups.

The formulation (18)–(22) is not linear because of the min operator in (19). Hooker and Williams (2012) show that we can linearize constraint (19) by introducing m real variables u_k and m binary variables ξ_k and treating the min function as a disjunction over the values of the new variables u_k. The variables ξ_k are then used to decide which of the two values the disjunction takes. Formally, (19) can be reformulated as follows:

$$f \geq (1-m)\tau + \sum_{G_k \in \mathscr{G}} u_k \tag{23}$$

$$\tau\xi_k + a_k v_k \leq u_k \leq \tau + a_k v_k \quad \forall G_k \in \mathscr{G} \tag{24}$$

$$(\tau - \Gamma)\xi_k + z \leq u_k \leq z \quad \forall G_k \in \mathscr{G} \tag{25}$$

$$u_k \geq 0 \quad \forall G_k \in \mathscr{G} \tag{26}$$

$$\xi_k \in \{0, 1\} \quad \forall G_k \in \mathscr{G} \tag{27}$$

The constant Γ is chosen such that $v_h - v_k \leq \Gamma$ for all $G_h, G_k \in \mathscr{G}$. Thus, for $\xi_k = 1$ group G_k would contribute with $u_k = \tau + v_k$ and for $\xi_k = 0$ with $u_k = z$. A formal proof of the correctness and sharpness of this formulation is given in Hooker and Williams (2012).

5.5 Envy-free assignments

Brought to our context, the concept of *envy-free* resource allocation from the social choice literature (Brams and Taylor 1996) may sound as follows: an envy-free allocation is one in which every student thinks he/she received the best project – based on his/her own valuation – and hence does not envy any other student for being in a project he/she would like more than the one received.

For example, consider the case of three groups G_1, G_2, G_3 and three project teams p_1, p_2, p_3 that can accommodate a single group each. The students have ranked the three projects yielding the following preference-value matrix (v_{kj} for $k = 1, 2, 3$ and $j = 1, 2, 3$):

$$
\begin{array}{lll}
v_{11} = 0 & v_{12} = 1 & v_{13} = 4 \\
v_{21} = 1 & v_{22} = 0 & v_{23} = 7 \\
v_{31} = 2 & v_{32} = 3 & v_{33} = 6
\end{array}
$$

Consider the assignment $\langle \sigma(G_1) = p_3, \sigma(G_2) = p_1, \sigma(G_3) = p_2 \rangle$. Under this assignment students in group G_1 envy students in group G_3, because group G_1 prefers project p_2 to project p_3, i.e. $v_{12} < v_{13}$. Similarly, students in group G_3 envy students in group G_2, because $v_{31} < v_{32}$.

More formally, a group G_k assigned to a project team p_j is envious of another group G_h assigned to project p_i, if p_i is preferred to p_j by G_k, that is, if $v_{ki} < v_{kj}$. A project assignment is *envy-free* if no group is envious of any other group. Note that the envy of a group depends only on the allocations and on its own preference set. The preference sets of the other groups are irrelevant for the envy of a given group. The computational complexity of determining whether an efficient envy-free allocation exists in fair division problems is analyzed in Bouveret and Lang (2008). In general, to decide if an envy-free assignment exists is an NP-complete problem.

For assignments that are not envy-free, we can minimize the number of students that are envious. We can measure the amount of envy of G_k for G_h as the difference in the preference value of G_k for the two projects (i.e., $v_{k\sigma(G_k)} - v_{k\sigma(G_h)}$). The total amount of envy of a group $G_k \in \mathscr{G}$ can then be defined as the maximum envy with respect to all other groups:

$$e_k = \max \{v_{k\sigma(G_k)} - v_{k\sigma(G_h)} \mid G_h \in \mathscr{G}, h \neq k, \sigma(G_h) \in L_k\}$$

Note that a group G_k might envy another group G_h only for the projects to which both groups have expressed a preference. If group G_h is assigned to a project p_i that is not in L_k (the set of project that could be assigned to group G_k), then G_k does not envy G_h. The equation for e_k can be expressed using the following linear constraints:

$$e_k \geq \sum_{p_j \in L_k} v_{kj} x_{kj} - \sum_{p_i \in L_h \cap L_k} v_{ki} x_{hi} - \sum_{p_i \in L_h \setminus L_k} \Delta x_{hi} \quad \forall G_h \in \mathscr{G}, h \neq k. \tag{28}$$

Since we impose $e_k \geq 0$, the r.h.s. is ignored when it becomes negative; hence, the previous equation correctly accounts for the envy of group G_k.

The objective that minimizes the total envy of students is:

$$\min \sum_{G_k \in \mathscr{G}} a_k e_k. \tag{29}$$

Note that, envy-free assignments can fail to be Pareto efficient. For example, consider an assignment that is not efficient because everybody could obtain a better project according to their preferences under another assignment. If the price for this better assignment for all students is to end up with a student who envies another one that has got an even better assignment than he/she did, then it might not be worthwhile in terms of envy to move to the new assignment. In fact there are cases where envy-free and Pareto efficient assignments are incompatible. For example, the situation where we have a single project and two groups who both want it but there is not space for both in it: in that case, any allocation is either efficient but not envy-free (when the project is given to one of the two groups), or envy-free but not efficient (if the project is not allocated to anyone).

5.6 Stability

The following undesired situation may arise in the assignments obtained by the models so far formulated: a group G_k is assigned in σ to project team $p_{\sigma(k)}$, while there is another project p_j preferred by G_k (i.e, $v_{kj} < v_{k,\sigma(k)}$) that has still space left to accommodate G_k. This situation may happen when it is profitable under the previously seen criteria to have some students in $p_{\sigma(k)}$ and the cheapest way to satisfy the lower bound on the team size (constraints 3) is to have G_k in $p_{\sigma(k)}$. We observed that this situation does arise in practice in the assignments found with the previous criteria. Since this situation is easy to spot and undesirable from the local standpoint of the groups, we set out to minimize its occurrences. We call *stable* an assignment that does not contain any undesired situation of the type described in the lines above.

The criterion of striving towards stability can be formalized in MILP with additional variables and linear constraints, as follows. First, we introduce a slack variable b_j, $p_j \in P$ for the upper bound constraints (2):

$$\sum_{G_k \in \mathscr{G}\,:\,p_j \in L_k} a_k x_{kj} + b_j = u_j y_j, \quad \forall p_j \in P. \tag{30}$$

Second, we define for each project $p_j \in P$ a 0–1 variable z_{kj}, $G_k \in \mathscr{G}$, $p_j \in L_k$ that equals 1 if there is space for G_k in project p_j either when it has a positive slack (i.e., $b_j > 0$) or when it is not assigned to any group (i.e., $y_j = 0$). These two conditions are formalized with the following linear constraints:

$$b_j + 1 - a_k \le u_j z_{kj}, \quad \forall G_k \in \mathscr{G}, p_j \in L_k, \tag{31}$$

$$a_k + 1 - (1 - y_j)l_j \le u_j z_{kj} + (u_j + 1)y_j, \quad \forall G_k \in \mathscr{G}, p_j \in L_k \tag{32}$$

If project team p_j is open ($y_j = 1$) and there is space left ($b_j > 0$), constraint (31) is binding and sets $z_{kj} = 1$ if the space is enough for G_k. If project team p_j is closed ($y_j = 0, b_j = 0$) and G_k in p_j satisfies the lower bound on the team size, constraint (32) is binding and sets $z_{kj} = 1$. In the other case left ($y_j = 1, b_j = 0$), constraints (31) and (32) are not binding for z_{kj} (we assume $a_k \le u_j$ for all $G_k \in \mathscr{G}$ and $p_j \in L_k$) and its value will be set to zero because of the following. We introduce a further variable $d_{ki} \ge 0$ for all $G_k \in \mathscr{G}$ and $p_i \in L_k$ that indicates the deviation from stability for group G_k with respect to project p_i:

$$d_{ki} = \begin{cases} \max\{0, \max\{v_{ki} - v_{kj} \mid p_j \in L_k \wedge z_{kj} = 1\}\} & \text{if } x_{ki} = 1 \\ 0 & \text{otherwise} \end{cases}$$

For all $G_k \in \mathscr{G}$ and $p_i \in L_k$ this expression can be linearized as follows:

$$d_{ki} \ge (v_{ki} - v_{kj})(x_{ki} + z_{kj} - 1) \quad \forall p_j \in L_k, p_j \ne p_i, v_{kj} < v_{ki}. \tag{33}$$

Then, if we wish to enforce stability in the definition of a feasible assignment, we will add the following constraint:

$$\sum_{G_k \in \mathscr{G}} \sum_{p_j \in L_k} d_{kj} = 0. \tag{34}$$

Alternatively, if we wish to minimize the deviation from stability (or amount of instability) we will instead minimize the summation in the previous constraint, that is:

$$\min \sum_{G_k \in \mathscr{G}} \sum_{p_j \in L_k} a_k d_{kj}. \tag{35}$$

6 Experimental results

We consider the instances from to the last nine editions of the course. A few statistics on these instances are reported in the first five columns of Table 1. In the instances of 2015 and 2016 students were not allowed to register in groups (see Sect. 6.5 for a discussion on this). All instances are publicly available with anonymized names at http://www.imada.sdu.dk/~marco/SPA/.

All experiments were conducted on a desktop machine with CPU Intel(R) Core(TM) (8 cores) i7-2600 at 3.40GHz, 16 GB RAM and operating system Linux Ubuntu 64-bit. The MILP solver used was Gurobi 7.0.1, which in all cases was run with a single thread.

In what follows we compare the criteria introduced in the previous section on the basis, primarily, of the distribution of students over priorities. We also report three other quality measures: the total instability, the total utility with weighting scheme (8) and the total envy, as defined in Sect. 5. Then, for each measure x of total utility or total envy we compute the relative deviation $(x - best)/best$, where $best$ is the best possible value for that measure given by the model that minimizes only that measure. For the total utility, the best possible value is given by the *weighted_identity* model, while for the envy it is given by the *envyfree* model. In none of the instances these values are zero.

6.1 Quality criteria: separated analysis

Numerical results for the models arising by the different individual criteria introduced in Sect. 5 are shown in Table 2. We denote by *leximin* and *weighted_owa* the two different procedure for finding generous maximal matchings. Beside the measures mentioned above, the table shows also the computation times to solve the models. A computation time of 3600 seconds indicates that the solver was halted before termination and the best incurred solution was returned without proof of optimality. For reference we rewrite in the table also the results from Table 1 of the repeated lottery procedure. For reasons of space we omit the number of unassigned students and under-size teams, but these numbers are the main focus of Table 1 and are a main drawback of the lottery method.

The preference distributions of Table 2 are visualized by a stacked barchart in Fig. 1. The bars encode proportions and add up to one within each model. Comparison is done through lengths of bars for corresponding priorities, as indicated in the legend. The presence of more bars in the stack indicates the use of a wider range of priorities. In addition, it is possible to compare the cumulative distribution of students. For two distributions, the one with cumulative fractions of students more shifted towards left is better. For example, we

Table 2 For each year and model the table reports the distribution of students over the priorities, the deviation from stability (instability), and the relative deviation of total utility and total envy

Inst.	Criterion	Students per priority														Instab.	Utility	Envy	
year		14	13	12	11	10	9	8	7	6	5	4	3	2	1				sec.
2008	repeated_lottery	0	0	0	0	0	0	1	4	10	7	9	7	25	135	15	2.02	40.32	4.37
2008	minimax	0	0	0	0	0	0	0	0	0	0	19	75	61	45	53	0.41	1.19	0.50
2008	leximin	0	0	0	0	0	0	0	0	0	0	2	38	69	91	7	0.06	0.24	2.59
2008	weighted_owa	0	0	0	0	0	0	0	0	0	0	2	38	69	91	7	0.06	0.24	0.67
2008	weighted_identity	0	0	0	0	0	0	0	0	0	3	8	22	51	116	1	0.00	0.07	1.12
2008	weighted_exp	0	0	0	0	2	0	0	0	4	9	6	15	33	131	0	0.07	0.26	0.73
2008	greedy_maximum	0	0	0	0	2	1	0	2	9	8	6	14	23	135	8	0.16	0.52	5.36
2008	stability	0	0	0	1	3	3	2	19	24	14	33	24	22	55	0	1.22	3.34	2.04
2008	envyfree	0	0	0	0	0	0	0	0	0	6	10	27	56	101	0	0.10	0.00	3602.88
2009	repeated_lottery	0	0	0	0	0	0	0	0	0	3	6	1	11	108	21	4.92	160.29	2.18
2009	minimax	0	0	0	0	0	0	0	0	0	0	0	37	34	58	29	0.40	2.03	0.37
2009	leximin	0	0	0	0	0	0	0	0	0	0	0	5	31	93	2	0.01	0.06	1.19
2009	weighted_owa	0	0	0	0	0	0	0	0	0	0	0	5	31	93	3	0.01	0.06	0.28
2009	weighted_identity	0	0	0	0	0	0	0	0	0	0	0	7	26	96	2	0.00	0.10	0.30
2009	weighted_exp	0	0	0	0	0	1	1	0	3	1	2	1	14	106	6	0.09	0.68	0.30
2009	greedy_maximum	0	0	0	0	0	0	1	4	4	1	2	0	9	108	26	0.18	1.19	1.79
2009	stability	0	0	1	1	2	1	7	11	8	11	23	18	20	24	0	2.14	9.65	1.20
2009	envyfree	0	0	0	0	0	0	0	0	0	1	2	6	28	92	1	0.06	0.00	3602.21
2010	repeated_lottery	0	0	2	1	0	0	0	0	3	4	9	1	21	150	21	2.65	64.79	4.12
2010	minimax	0	0	0	0	0	0	0	0	0	0	0	78	65	50	29	0.51	1.76	0.69
2010	leximin	0	0	0	0	0	0	0	0	0	0	0	20	56	117	3	0.05	0.18	2.36
2010	weighted_owa	0	0	0	0	0	0	0	0	0	0	0	20	56	117	3	0.05	0.18	0.49
2010	weighted_identity	0	0	0	0	0	0	0	0	0	2	6	7	41	137	4	0.00	0.04	0.40
2010	weighted_exp	0	0	0	0	0	1	0	2	2	3	8	2	27	148	10	0.06	0.25	0.43
2010	greedy_maximum	0	0	0	0	0	0	1	2	2	4	6	3	24	149	18	0.16	0.63	7.86
2010	stability	2	0	0	4	5	6	8	7	11	15	21	18	43	51	0	1.69	5.97	2.44
2010	envyfree	0	0	0	0	0	0	0	0	0	3	9	14	38	129	0	0.09	0.00	3602.94
2011	repeated_lottery	2	0	5	0	1	2	2	7	5	6	11	5	31	170	87	0.99	20.10	6.45
2011	minimax	0	0	0	0	0	0	0	0	21	29	43	52	54	60	38	0.52	1.07	1.15
2011	leximin	0	0	0	0	0	0	0	0	5	17	23	22	77	115	1	0.08	0.17	6.02
2011	weighted_owa	0	0	0	0	0	0	0	0	5	17	23	22	77	115	2	0.08	0.17	0.83
2011	weighted_identity	0	0	0	0	0	0	2	4	6	9	19	12	59	148	1	0.00	0.01	0.76
2011	weighted_exp	0	0	2	0	3	2	6	9	4	6	10	4	45	167	14	0.15	0.34	0.83
2011	greedy_maximum	2	0	3	1	4	2	8	4	6	8	9	7	33	171	11	0.25	0.55	28.97
2011	stability	5	1	5	3	8	12	8	29	27	30	24	22	31	53	0	1.48	3.12	3.36
2011	envyfree	0	0	0	0	0	1	2	3	7	8	22	20	61	135	1	0.06	0.00	3604.24
2012	repeated_lottery	0	0	0	2	0	3	5	6	9	11	10	16	34	186	1	0.69	16.79	7.96
2012	minimax	0	0	0	0	0	0	0	0	0	67	75	60	48	50	250	0.63	1.32	1.74
2012	leximin	0	0	0	0	0	0	0	0	0	18	28	55	72	127	9	0.08	0.19	8.00
2012	weighted_owa	0	0	0	0	0	0	0	0	0	18	28	55	72	127	9	0.08	0.19	1.34
2012	weighted_identity	0	0	0	0	0	0	0	2	7	11	23	35	60	162	2	0.00	0.02	1.17
2012	weighted_exp	0	1	0	2	2	2	3	13	13	7	12	21	38	186	9	0.14	0.33	1.20
2012	greedy_maximum	0	0	0	3	2	2	3	13	13	7	12	21	38	186	7	0.14	0.32	12.78
2012	stability	1	1	4	0	2	3	13	27	40	32	36	39	53	49	0	1.09	2.27	4.69
2012	envyfree	0	0	0	0	0	0	0	4	8	15	20	46	56	151	2	0.07	0.00	3604.04
2013	repeated_lottery	0	0	0	0	3	1	3	3	8	8	11	18	46	226	45	0.42	13.49	10.68
2013	minimax	0	0	0	0	0	0	0	18	20	34	57	77	68	81	80	0.55	1.09	1.78
2013	leximin	0	0	0	0	0	0	0	1	8	18	18	50	102	158	12	0.04	0.07	12.74
2013	weighted_owa	0	0	0	0	0	0	0	1	8	18	18	50	102	158	12	0.04	0.07	1.56
2013	weighted_identity	0	0	0	0	0	0	2	3	8	13	20	42	81	186	7	0.00	-0.01	1.40
2013	weighted_exp	0	0	2	1	7	10	6	6	4	12	13	20	60	214	8	0.16	0.32	1.86
2013	greedy_maximum	0	0	2	2	6	3	7	13	13	16	14	19	37	222	20	0.23	0.47	37.30
2013	stability	1	1	3	6	13	15	19	34	27	34	49	47	53	51	0	1.34	2.68	5.53
2013	envyfree	0	0	0	0	0	1	1	4	6	12	28	48	82	173	4	0.04	0.00	3605.47
2014	repeated_lottery	0	0	0	1	0	0	5	9	6	12	11	19	37	241	90	0.39	13.84	11.17
2014	minimax	0	0	0	0	0	0	0	0	14	77	79	77	57	67	202	0.67	1.36	1.96
2014	leximin	0	0	0	0	0	0	0	0	1	20	32	45	106	167	7	0.04	0.08	12.19
2014	weighted_owa	0	0	0	0	0	0	0	0	1	20	32	45	106	167	7	0.04	0.08	1.82
2014	weighted_identity	0	0	0	0	0	0	1	1	6	21	25	30	85	202	1	0.00	0.00	1.82
2014	weighted_exp	0	0	0	1	4	3	5	8	11	13	20	22	54	230	10	0.10	0.22	1.77
2014	greedy_maximum	0	0	0	0	4	3	10	13	13	21	20	14	36	237	29	0.19	0.42	21.54
2014	stability	0	0	1	1	7	5	8	41	40	42	45	53	52	74	0	1.07	2.26	8.07
2014	envyfree	0	0	0	0	0	0	1	1	8	20	26	47	92	176	4	0.07	0.00	3604.81
2015	repeated_lottery	0	0	0	0	0	1	3	6	6	13	10	23	55	298	0	0.56	19.58	15.96
2015	minimax	0	0	0	0	0	0	0	0	0	0	0	155	139	128	199	0.36	0.78	2.61
2015	leximin	0	0	0	0	0	0	0	0	0	0	0	47	141	234	23	0.03	-0.07	11.11
2015	weighted_owa	0	0	0	0	0	0	0	0	0	0	0	47	141	234	21	0.03	-0.08	3.25
2015	weighted_identity	0	0	0	0	0	0	0	0	0	0	10	38	112	262	21	0.00	-0.15	3.02
2015	weighted_exp	0	0	0	1	1	2	1	2	6	9	7	26	78	289	24	0.08	0.06	3.42
2015	greedy_maximum	0	0	0	1	1	2	3	9	6	10	8	19	70	293	55	0.15	0.24	38.08
2015	stability	0	0	2	4	6	4	18	34	46	26	32	49	94	106	0	1.40	3.39	13.92
2015	envyfree	0	0	0	0	0	0	0	0	2	13	20	50	93	244	3	0.15	0.00	3609.32
2016	repeated_lottery	0	0	0	0	0	1	1	7	2	10	6	22	29	332	0	0.69	24.38	14.82
2016	minimax	0	0	0	0	0	0	0	0	0	99	82	97	73	65	505	1.25	3.41	2.68
2016	leximin	0	0	0	0	0	0	0	0	0	1	3	33	116	263	4	0.04	-0.08	16.43
2016	weighted_owa	0	0	0	0	0	0	0	0	0	1	3	33	116	263	8	0.04	-0.08	3.98
2016	weighted_identity	0	0	0	0	0	0	0	1	0	1	12	26	76	300	6	0.00	-0.17	2.91
2016	weighted_exp	0	0	0	0	1	1	3	5	4	7	10	13	48	324	21	0.08	0.09	3.80
2016	greedy_maximum	0	0	0	0	1	1	3	6	5	7	11	13	43	326	34	0.09	0.14	31.16
2016	stability	0	0	0	0	5	6	7	48	44	40	43	34	71	118	0	1.53	4.18	13.64
2016	envyfree	0	0	0	0	0	0	1	2	0	18	23	52	104	216	6	0.33	0.00	3607.33

Note that since the *envyfree* model is truncated before the solution has been proven optimal other models can achieve a better value, thus yielding a negative ratio. The last column reports the total solution time in seconds of the MILP solver

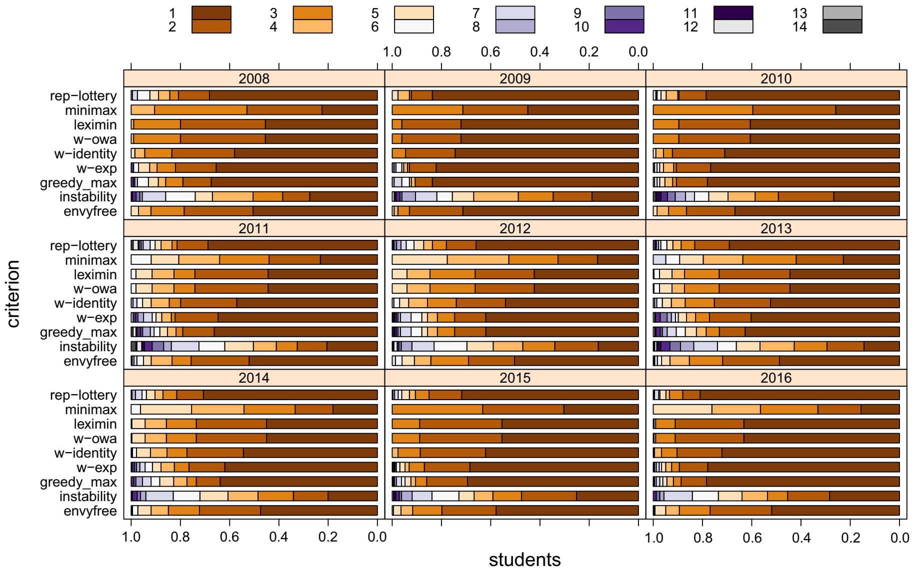

Fig. 1 A stacked barchart showing the proportion of students on each priority using the data from Table 2. Note that the plots are better interpreted from right to left (in compliance with the tables), i.e., the fraction of students is accumulated from right to left, and the first interval from the right of the bar represents the first priority, the second interval the second priority and so on. Hence, a long interval on the right indicates a large number of students in the first priority, which is clearly desired

see that in 2009 the *leximin* and *weighted_owa* models assign more students in the first two priorities than the other models do.

We make the following observations.

- Since all MILP models returned an optimal or at least feasible solution, as defined by constraints (1)–(5), no student is left unassigned and no project undersized. This is a clear breakthrough made available by the MILP based approaches with respect to the repeated lottery algorithm of Table 1. Thanks to these models the cumbersome post-processing task of calling back students and asking them to change their priorities can be avoided.
- With the only exception of the *envyfree* model, all models are computationally easy to solve. The *envyfree* model was not solved to optimality but a feasible solution was found within the given time limit.
- The *minimax* approach improves considerably the (distributive) fairness criterion over the repeated lottery and *weighted_identity*. On year 2011, 2 students would have been assigned to their 14th priority while with *minimax* no student is assigned to a priority worse than the 6th. On the other hand, the *minimax* model is unable to discriminate among the possibly many distributions that guarantee the best possible worst priority.
- The *leximin* model improves considerably, as expected, the results of *minimax* while enforcing the minimax criterion as well. For example, in year 2012, *leximin* assigns 18 students to their 5th priority against 67 assigned in the first solution returned by *minimax*. Clearly, the *leximin* model has to be preferred to *minimax*.
- The *weighted_owa* model obtains exactly the same results as *leximin*, as expected from theory. Small discrepancies could be due to numerical rounding but they did not appear

in our experiments. The advantage of the *weighted_owa* model is that it implies faster computation times as it solves one single model instead than a sequence of models.

– Moving in the other direction of preference than *leximin* and *weighted_owa*, that is, towards assignments in which more importance is given to differences among the highest priorities rather than among the lowest, we encounter the weighted schemes and the *greedy_maximum* approach. The effect of moving in this direction of the spectrum of possible choices is clearly visible in Fig. 1: the number of students in the first priority increases moving towards *greedy_maximum* while the tails in the worst priorities becomes longer.
  The *weighted_exp* model promotes more than *weighted_identity* the best priorities. In this weighting scheme, the weight of rank values above 7 is uniform and equal to 1, thus yielding the long tails with the bad priorities that we observe in Fig. 1. The *greedy_maximum* approach assigns the highest number of students to their best priority among all models. It has however the mentioned drawback of using priorities below the *minimax* solution. Moreover, the running times, although still affordable, are the worst after those of the *envyfree* model.
– In the *stability* model we treat stability as the only objective and observe that stable assignments exist for all nine instances (the column "Instability" reports the total free places that students would prefer to occupy from an individual point of view). When several stable assignments exist the one described in Table 2 is the first one found, or equivalently, a random one returned by the solver. Its quality from the point of view of student distribution is not good.
  Separately, we assessed the price of stability in terms of fairness by including stability in the feasibility conditions (1)–(5). Compared to the *minimax* model from Table 2 the new model, that minimizes the worst preference value subject to stability and the other feasibility constraints, attains worse results only on instance 2012, where the worst preference became 6 instead of 5.
– The *envyfree* model is not solvable to optimality within the 1 h time limit that we used. The upper bound results after the time limit are interesting and comparable with the other models in terms of utility. However, the *weighted_identity* and *leximin* criteria seem to do better in terms of distributions and to do well, indirectly, also with respect to envy.

We conclude that the models *weighted_identity*, *weighted_owa*, *weighted_exp* and *greedy_maximum* are the most appealing with respect to the distribution of students over preferences. However, a closer look at their solutions revealed that the lack of stability is easily detectable and this was a serious issue for their acceptability in practice. Moreover, the two weighting schemes, identity and exponential, and the greedy maximum matching do not take into account fairness and may end up with undesired long tail distributions.

6.2 Combinations of criteria

In the light of the previous observations, a reasonable attempt is to focus on feasible assignments that are (i) "stable" and that (ii) respect the minimax criterion and to use weighted utility approaches to further direct the search. We achieve this goal by including in the feasibility conditions (1)–(5) a constraint that imposes stability and a constraint on the worst preference value as determined by preprocessing the instance with the *minimax* model. For the instance 2012 we had to decide between enforcing a minimax solution of 6 instead of 5 or allowing 2 students to see an unstable solution. We chose the former.

We finally tested the following combinations (terms in parentheses are feasibility constraints added to those of Sect. 5):

a. *(stable)-weighted_owa*
b. *(stable+minimax)-weighted_identity*
c. *(stable+minimax)-weighted_exp*
d. *(stable+minimax)-greedy_max*

Recall that the minimax criterion is subsumed by the *weighted_owa* (i.e., *leximin*) and hence we do not consider here the model *(stable+minimax)-weighted_owa*. In order to assess empirically the loss of global utility due to equity (price of fairness) and stability (price of stability) we include also the following models:

b'. *(stable)-weighted_identity*
c'. *(stable)-weighted_exp*
d'. *(stable)-greedy_max*
b''. *(minimax)-weighted_identity*
c''. *(minimax)-weighted_exp*
d''. *(minimax)-greedy_max*

The results of these combinations are reported in Table 3 and Fig. 2. We can draw the following conclusions:

- The first important observation is that all combined models remain computationally practicable with respect to our aim of solution times in the order of a few minutes. The only exception is the combination *(stable)-greedy_max* that exhibits highly varying running times, reaching 5157 s for the instance of 2015.[4]
- Comparing the combinations *(minimax)*-$*$ and *(stable+minimax)*-$*$ we observe that some stability is lost if only the minimax constraint is included. On the other hand, looking at the column "Utility", which gives the deviation from the best utility, we see instead that the loss in global utility is very small if also stability is included among the constraints. Hence, the changes to make the solution stable seem not to affect the global utility too much.
- The exclusion of the minimax constraint in the models b', c' and d' leads to assignments that still exhibit long tails, which we deem undesirable. On the other hand, fairness in the sense of minimax criterion has a price in terms of global utility. Comparing the models *(stable)*-$*$ versus *($*$+minimax)*-$*$ we see that the price to pay for fairness is fewer students receiving their first and second priorities.
- The solutions of the models *(stable+minimax)-greedy_max* and *(stable+minimax)-weighted_exp* are very similar in terms of distributions. However, the model *(stable+minimax)-greedy_max* might require slightly more computation time than *(stable+minimax)-weighted_exp*, since it has to solve a series of MILP problems. The introduction of the stability constraint seems to worsen considerably the computation times of *(stable+minimax)-greedy_max* while the minimax constraint helps to speed up the solution. However, we can regard the *(stable+minimax)-weighted_exp* as a surrogate of the policy implemented by the *(stable+minimax)-greedy_max* model.

[4] We set a time limit of 3600 s per MILP problem but on instance 2015 *(stable)-greedy_max* had to solve 13 MILP problems.

Table 3 Combined criteria models: for each year and model the table reports the distribution of students over the priorities, the deviation from stability (instability), and the relative deviation of total utility and total envy

Inst. year	Criterion	Students per priority ...	11	10	9	8	7	6	5	4	3	2	1	Instab.	Utility	Envy	sec.
2008	(stable)-weighted_owa	0	0	0	0	0	0	0	0	3	35	68	94	0	0.05	0.20	2.91
2008	(stable)-weighted_identity	0	0	0	0	0	0	0	3	8	20	55	114	0	0.00	0.07	2.62
2008	(minimax)-weighted_identity	0	0	0	0	0	0	0	0	11	23	54	112	4	0.01	0.08	0.55
2008	(stable+minimax)-weighted_identity	0	0	0	0	0	0	0	0	10	23	57	110	0	0.01	0.08	2.00
2008	(stable)-weighted_exp	0	0	2	0	0	0	4	9	6	15	33	131	0	0.07	0.26	2.24
2008	(minimax)-weighted_exp	0	0	0	0	0	0	0	0	15	24	43	118	5	0.02	0.11	0.54
2008	(stable+minimax)-weighted_exp	0	0	0	0	0	0	0	0	13	26	44	117	0	0.01	0.10	2.13
2008	(stable)-greedy_max	0	0	2	0	0	2	9	9	6	12	26	134	0	0.15	0.47	25.84
2008	(minimax)-greedy_max	0	0	0	0	0	0	0	0	18	22	41	119	5	0.02	0.14	2.10
2008	(stable+minimax)-greedy_max	0	0	0	0	0	0	0	0	20	21	40	119	0	0.03	0.17	6.68
2009	(stable)-weighted_owa	0	0	0	0	0	0	0	0	0	6	33	90	0	0.03	0.13	1.05
2009	(stable)-weighted_identity	0	0	0	0	0	0	0	1	2	5	21	100	0	0.01	0.13	1.06
2009	(minimax)-weighted_identity	0	0	0	0	0	0	0	0	0	7	26	96	2	0.00	0.10	0.20
2009	(stable+minimax)-weighted_identity	0	0	0	0	0	0	0	0	0	10	24	95	0	0.02	0.23	0.92
2009	(stable)-weighted_exp	0	0	0	0	0	1	0	1	3	5	16	103	0	0.03	0.26	3.39
2009	(minimax)-weighted_exp	0	0	0	0	0	0	0	0	0	8	24	97	2	0.00	0.10	0.20
2009	(stable+minimax)-weighted_exp	0	0	0	0	0	0	0	0	0	11	22	96	0	0.02	0.23	0.90
2009	(stable)-greedy_max	0	0	0	0	0	2	1	2	3	3	14	104	0	0.08	0.52	41.00
2009	(minimax)-greedy_max	0	0	0	0	0	0	0	0	0	11	20	98	5	0.01	0.19	0.78
2009	(stable+minimax)-greedy_max	0	0	0	0	0	0	0	0	0	11	22	96	0	0.02	0.23	2.06
2010	(stable)-weighted_owa	0	0	0	0	0	0	0	0	0	20	56	117	0	0.05	0.17	1.98
2010	(stable)-weighted_identity	0	0	0	0	0	0	0	1	7	8	41	136	0	0.00	0.01	2.06
2010	(minimax)-weighted_identity	0	0	0	0	0	0	0	0	0	23	48	122	2	0.05	0.18	0.41
2010	(stable+minimax)-weighted_identity	0	0	0	0	0	0	0	0	0	23	50	120	0	0.05	0.18	1.59
2010	(stable)-weighted_exp	0	0	0	1	0	1	1	2	7	6	27	146	0	0.11	0.41	2.36
2010	(minimax)-weighted_exp	0	0	0	0	0	0	0	0	0	26	43	124	3	0.05	0.21	0.37
2010	(stable+minimax)-weighted_exp	0	0	0	0	0	0	0	0	0	25	46	122	0	0.05	0.18	1.65
2010	(stable)-greedy_max	2	0	0	1	2	1	1	3	9	2	24	148	0	0.15	0.58	25.11
2010	(minimax)-greedy_max	0	0	0	0	0	0	0	0	0	26	43	124	6	0.05	0.21	1.47
2010	(stable+minimax)-greedy_max	0	0	0	0	0	0	0	0	0	25	46	122	0	0.05	0.18	4.11
2011	(stable)-weighted_owa	0	0	0	0	0	0	5	17	23	22	78	114	0	0.08	0.17	3.35
2011	(stable)-weighted_identity	0	0	0	0	2	4	7	8	19	11	61	147	0	0.00	0.01	3.81
2011	(minimax)-weighted_identity	0	0	0	0	0	0	10	12	20	14	66	137	0	0.02	0.04	0.72
2011	(stable+minimax)-weighted_identity	0	0	0	0	0	0	10	13	19	15	63	139	0	0.02	0.04	3.19
2011	(stable)-weighted_exp	5	0	2	2	6	4	6	7	10	7	39	168	0	0.18	0.41	3.84
2011	(minimax)-weighted_exp	0	0	0	0	0	0	17	15	16	10	51	150	7	0.04	0.09	0.84
2011	(stable+minimax)-weighted_exp	0	0	0	0	0	0	16	13	18	10	54	148	0	0.03	0.07	3.32
2011	(stable)-greedy_max	3	0	2	2	8	6	7	9	10	7	30	171	0	0.22	0.50	53.69
2011	(minimax)-greedy_max	0	0	0	0	0	0	24	22	20	8	28	157	22	0.14	0.30	5.46
2011	(stable+minimax)-greedy_max	0	0	0	0	0	0	20	20	19	11	36	153	0	0.10	0.24	33.27
2012	(stable)-weighted_owa	0	0	0	0	0	0	1	20	28	53	69	129	0	0.09	0.20	5.35
2012	(stable)-weighted_identity	0	0	0	0	0	4	8	9	22	36	59	162	0	0.01	0.05	4.95
2012	(minimax)-weighted_identity	0	0	0	0	0	0	9	9	24	40	61	157	3	0.01	0.04	1.31
2012	(stable+minimax)-weighted_identity	0	0	0	0	0	0	12	8	19	46	60	155	0	0.02	0.06	5.64
2012	(stable)-weighted_exp	0	4	1	3	6	11	10	7	12	23	39	184	0	0.15	0.35	6.00
2012	(minimax)-weighted_exp	0	0	0	0	0	0	14	23	18	24	50	171	7	0.04	0.11	1.12
2012	(stable+minimax)-weighted_exp	0	0	0	0	0	0	16	20	22	24	48	170	0	0.05	0.14	5.29
2012	(stable)-greedy_max	1	3	1	2	5	13	13	7	14	19	37	185	0	0.17	0.38	76.90
2012	(minimax)-greedy_max	0	0	0	0	0	0	0	35	34	32	40	159	5	0.09	0.22	6.14
2012	(stable+minimax)-greedy_max	0	0	0	0	0	0	20	26	19	27	35	173	0	0.10	0.24	27.11
2013	(stable)-weighted_owa	0	0	0	0	0	1	8	18	18	50	107	153	0	0.04	0.08	6.91
2013	(stable)-weighted_identity	0	0	0	0	2	4	7	12	16	50	82	182	0	0.00	-0.00	6.44
2013	(minimax)-weighted_identity	0	0	0	0	0	5	7	13	19	48	80	183	7	0.00	-0.00	1.57
2013	(stable+minimax)-weighted_identity	0	0	0	0	0	5	7	13	19	49	80	182	0	0.00	0.00	6.04
2013	(stable)-weighted_exp	2	0	6	3	6	11	7	10	15	21	62	212	0	0.12	0.24	8.04
2013	(minimax)-weighted_exp	0	0	0	0	0	16	16	16	16	26	58	207	8	0.07	0.14	1.59
2013	(stable+minimax)-weighted_exp	0	0	0	0	0	16	12	14	17	30	64	202	0	0.05	0.10	6.68
2013	(stable)-greedy_max	2	1	7	4	9	14	13	14	8	21	41	219	0	0.26	0.53	406.92
2013	(minimax)-greedy_max	0	0	0	0	0	25	15	24	12	19	49	211	16	0.13	0.28	11.28
2013	(stable+minimax)-greedy_max	0	0	0	0	0	23	15	22	14	22	51	208	0	0.13	0.26	63.82
2014	(stable)-weighted_owa	0	0	0	0	0	0	1	20	32	46	102	170	0	0.04	0.09	7.35
2014	(stable)-weighted_identity	0	0	0	0	0	1	8	21	24	33	79	205	0	0.00	0.00	7.28
2014	(minimax)-weighted_identity	0	0	0	0	0	0	8	19	25	37	83	199	1	0.00	0.00	1.86
2014	(stable+minimax)-weighted_identity	0	0	0	0	0	0	8	20	27	31	85	200	0	0.00	0.00	6.66
2014	(stable)-weighted_exp	0	1	4	2	6	6	9	16	21	17	63	226	0	0.09	0.19	8.38
2014	(minimax)-weighted_exp	0	0	0	0	0	0	17	24	21	29	64	216	10	0.03	0.07	1.86
2014	(stable+minimax)-weighted_exp	0	0	0	0	0	0	15	26	23	25	68	214	0	0.03	0.07	7.27
2014	(stable)-greedy_max	0	0	4	3	7	17	15	18	16	13	44	234	0	0.18	0.39	123.35
2014	(minimax)-greedy_max	0	0	0	0	0	0	20	32	21	26	51	221	18	0.07	0.15	11.17
2014	(stable+minimax)-greedy_max	0	0	0	0	0	0	23	31	29	21	46	221	0	0.09	0.21	42.21
2015	(stable)-weighted_owa	0	0	0	0	0	0	0	0	0	52	141	229	0	0.04	-0.05	17.78
2015	(stable)-weighted_identity	0	0	0	0	0	0	1	1	10	40	103	267	0	0.01	-0.14	16.24
2015	(minimax)-weighted_identity	0	0	0	0	0	0	0	0	0	57	113	252	20	0.01	-0.12	2.77
2015	(stable+minimax)-weighted_identity	0	0	0	0	0	0	0	0	0	59	120	243	0	0.03	-0.09	11.77
2015	(stable)-weighted_exp	0	0	0	2	0	0	6	9	12	29	76	288	0	0.05	-0.03	15.36
2015	(minimax)-weighted_exp	0	0	0	0	0	0	0	0	0	65	99	258	21	0.02	-0.10	2.74
2015	(stable+minimax)-weighted_exp	0	0	0	0	0	0	0	0	0	67	105	250	0	0.03	-0.08	12.01
2015	(stable)-greedy_max	0	2	3	1	3	9	9	12	12	17	60	292	0	0.26	0.53	5157.09
2015	(minimax)-greedy_max	0	0	0	0	0	0	0	0	0	67	96	259	22	0.02	-0.10	8.23
2015	(stable+minimax)-greedy_max	0	0	0	0	0	0	0	0	0	71	100	251	0	0.04	-0.07	31.12
2016	(stable)-weighted_owa	0	0	0	0	0	0	0	1	3	33	120	259	0	0.04	-0.07	12.59
2016	(stable)-weighted_identity	0	0	0	0	0	1	1	2	9	24	81	298	0	0.00	-0.16	13.05
2016	(minimax)-weighted_identity	0	0	0	0	0	0	0	1	14	24	80	297	5	0.00	-0.17	3.17
2016	(stable+minimax)-weighted_identity	0	0	0	0	0	0	0	1	13	28	79	295	0	0.01	-0.16	12.28
2016	(stable)-weighted_exp	0	0	0	1	2	4	2	8	10	16	53	320	0	0.05	-0.01	16.53
2016	(minimax)-weighted_exp	0	0	0	0	0	0	0	14	15	18	54	315	11	0.03	-0.08	3.25
2016	(stable+minimax)-weighted_exp	0	0	0	0	0	0	0	10	15	23	57	311	0	0.02	-0.10	12.49
2016	(stable)-greedy_max	0	0	1	1	1	8	4	11	10	10	46	324	0	0.10	0.15	139.36
2016	(minimax)-greedy_max	0	0	0	0	0	0	0	16	16	19	48	317	15	0.04	-0.04	14.96
2016	(stable+minimax)-greedy_max	0	0	0	0	0	0	0	17	16	18	50	315	0	0.05	-0.03	48.33

The last column reports the total solution time in seconds for the MILP solver

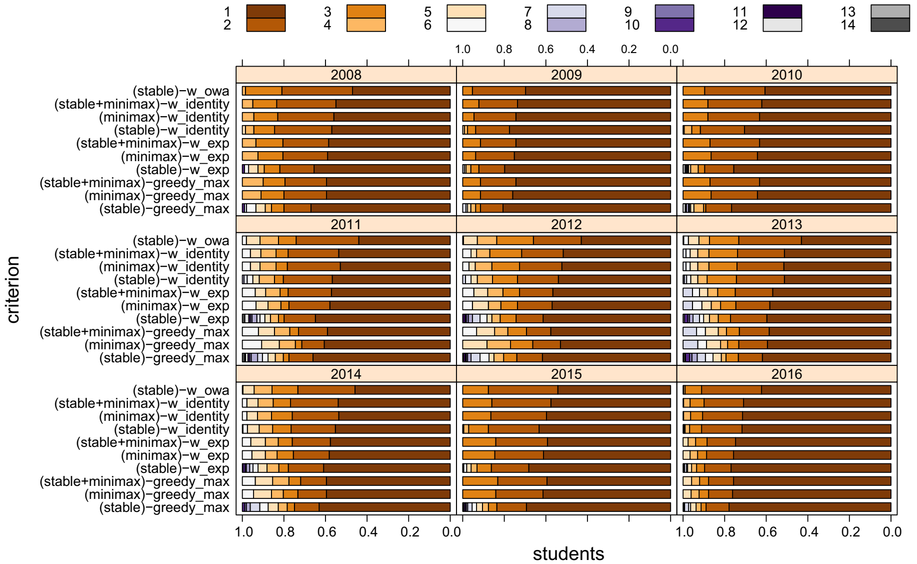

Fig. 2 A stacked barchart showing the proportion of students on each priority for the combined models

– The models a, b and c (or d) are all interesting and non-dominating each other. They do well also in terms of envy, which they do not optimize directly. They combine principles of fairness, stability, profile and weighted utility and provide three different solutions with similar characteristics but quite different distributions. The decision maker can then decide which of these three solutions to use.

It was with these last three models that we finally satisfied the administering committee. Their choice in the latest years fell on the assignment provided by the *(stable+minimax)-weighted_exp* (or *(stable+minimax)-greedy_max*) model.

6.3 Combined utilitarian-minimax model

Finally, we assess the combined utilitarian-minimax model by Hooker and Williams (2012). We include the constraint of stability in the model and test values for the threshold τ in $\{0, 1, 2, 3, 4, 5, 6, 7, 8\}$. A value of 0 for τ implies that all differences $z - v_{k\sigma(k)}$ are larger than the threshold and hence that each group contributes with its preference value, thus yielding a classical utilitarian criterion equivalent to *(stable)-weighted_identity*. On the other side, the value 8 for τ implies that all differences are below the threshold and that each group contributes with the worst preference value thus yielding a minimax approach.[5] We are interested in observing how solutions change between these two values of the threshold. The results are reported in Table 4 and visualized in Fig. 3.

– As expected *stable-hooker_0* leads to assignments of equal quality as *(stable)-weighted_identity* from Table 3 and *stable-hooker_8* to assignments similar to, for example, *(stable)-weighted_owa* (differences may be due to the fact that there are actually several optimal solutions with the same equal cost).

[5] Actually, a few students gave more than 7 preferences but this seems not to have an impact in our results.

Table 4 Combined utility-minimax models: for each year and model the table reports the distribution of students over the priorities, the deviation from stability (instability), and the relative deviation of total utility and total envy

Inst. year	Criterion	Students per priorities									Instab.	Utility	Envy	Gap	Seconds
		9	8	7	6	5	4	3	2	1					
2008	instab_hooker-0	0	0	0	0	3	8	21	53	115	0	0.00	0.07	0.00	2.58
2008	instab_hooker-1	0	0	0	0	0	16	18	53	113	0	0.02	0.12	1.20	3603.61
2008	instab_hooker-2	0	0	0	0	0	19	33	29	119	0	0.06	0.25	20.90	3603.49
2008	instab_hooker-3	0	0	0	0	0	30	53	52	65	0	0.35	0.91	0.00	2916.57
2008	instab_hooker-4	0	0	0	0	0	4	58	57	81	0	0.16	0.52	0.00	149.52
2008	instab_hooker-5	0	0	0	0	0	11	47	60	82	0	0.17	0.50	0.00	77.04
2008	instab_hooker-6	0	0	0	0	0	9	59	59	73	0	0.22	0.64	0.00	452.20
2008	instab_hooker-7	0	0	0	0	0	9	58	59	74	0	0.21	0.60	0.01	339.29
2008	instab_hooker-8	0	0	0	0	0	7	56	64	73	0	0.20	0.60	0.00	7.47
2009	instab_hooker-0	0	0	0	0	1	0	7	23	98	0	0.01	0.13	0.00	1.00
2009	instab_hooker-1	0	0	0	0	0	0	11	22	96	0	0.02	0.23	11.03	3602.54
2009	instab_hooker-2	0	0	0	0	0	0	24	38	67	0	0.27	1.35	0.00	5.56
2009	instab_hooker-3	0	0	0	0	0	0	15	46	68	0	0.21	0.87	0.00	32.87
2009	instab_hooker-4	0	0	0	0	0	0	8	33	88	0	0.05	0.26	0.00	130.15
2009	instab_hooker-5	0	0	0	0	0	0	10	34	85	0	0.08	0.45	0.00	41.76
2009	instab_hooker-6	0	0	0	0	0	0	8	37	84	0	0.08	0.45	0.00	142.46
2009	instab_hooker-7	0	0	0	0	0	0	15	26	88	0	0.09	0.39	0.00	101.49
2009	instab_hooker-8	0	0	0	0	0	0	10	38	81	0	0.11	0.39	0.01	113.83
2010	instab_hooker-0	0	0	0	0	1	7	8	41	136	0	0.00	0.01	0.00	1.86
2010	instab_hooker-1	0	0	0	0	0	0	28	43	122	0	0.07	0.22	7.09	3603.26
2010	instab_hooker-2	0	0	0	0	0	0	24	72	97	0	0.14	0.49	0.00	95.75

Table 4 continued

Inst. year	Criterion	Students per priorities									Instab.	Utility	Envy	Gap	Seconds
		9	8	7	6	5	4	3	2	1					
2010	instab_hooker-3	0	0	0	0	0	0	27	66	100	0	0.14	0.50	0.00	206.33
2010	instab_hooker-4	0	0	0	0	0	0	37	62	94	0	0.20	0.67	0.00	94.11
2010	instab_hooker-5	0	0	0	0	0	0	31	68	94	0	0.18	0.62	0.01	67.71
2010	instab_hooker-6	0	0	0	0	0	0	28	65	100	0	0.15	0.51	0.00	109.50
2010	instab_hooker-7	0	0	0	0	0	0	27	64	102	0	0.14	0.46	0.00	62.72
2010	instab_hooker-8	0	0	0	0	0	0	52	62	79	0	0.31	1.04	0.00	46.51
2011	instab_hooker-0	0	2	4	6	9	18	15	57	148	0	0.00	0.01	0.00	3.56
2011	instab_hooker-1	0	0	0	13	10	18	14	66	138	0	0.02	0.04	0.01	455.28
2011	instab_hooker-2	0	0	0	16	13	15	11	58	146	0	0.03	0.06	9.92	3604.55
2011	instab_hooker-3	0	0	0	14	16	17	8	59	145	0	0.03	0.07	24.01	3604.46
2011	instab_hooker-4	0	0	0	14	25	21	17	31	151	0	0.11	0.24	29.11	3604.83
2011	instab_hooker-5	0	0	0	18	20	25	41	42	113	0	0.25	0.54	0.00	1545.97
2011	instab_hooker-6	0	0	0	9	18	25	23	60	124	0	0.11	0.22	0.00	1305.65
2011	instab_hooker-7	0	0	0	8	32	33	35	35	116	0	0.25	0.53	0.00	1660.42
2011	instab_hooker-8	0	0	0	10	18	27	40	52	112	0	0.18	0.38	0.00	1407.81
2012	instab_hooker-0	0	0	4	3	8	18	18	45	110	0	0.01	0.05	0.00	4.51
2012	instab_hooker-1	0	0	0	7	8	19	24	39	109	0	0.02	0.06	0.01	639.00
2012	instab_hooker-2	0	0	0	10	10	18	22	35	111	0	0.03	0.09	7.31	3605.28
2012	instab_hooker-3	0	0	0	9	12	20	16	32	117	0	0.05	0.13	28.76	3455.36
2012	instab_hooker-4	0	0	0	14	18	31	32	39	166	0	0.08	0.19	30.59	3601.18

Table 4 continued

Inst. year	Criterion	Students per priorities									Instab.	Utility	Envy	Gap	Seconds
		9	8	7	6	5	4	3	2	1					
2012	instab_hooker-5	0	0	0	16	39	46	62	64	73	0	0.46	0.96	0.00	1837.99
2012	instab_hooker-6	0	0	0	2	42	43	49	53	111	0	0.28	0.60	3133.93	3605.14
2012	instab_hooker-7	0	0	0	6	40	47	46	47	114	0	0.31	0.63	102.39	3605.24
2012	instab_hooker-8	0	0	0	9	44	52	47	58	90	0	0.41	0.84	50.56	3605.22
2013	instab_hooker-0	0	2	3	7	14	18	47	80	184	0	0.00	− 0.00	0.00	5.78
2013	instab_hooker-1	0	0	7	7	12	16	48	84	181	0	0.01	0.00	0.01	2503.27
2013	instab_hooker-2	0	0	10	8	10	16	43	84	184	0	0.01	0.01	17.80	3606.37
2013	instab_hooker-3	0	0	14	11	15	13	35	69	198	0	0.04	0.08	37.02	3606.18
2013	instab_hooker-4	0	0	18	15	13	17	24	68	200	0	0.08	0.14	40.27	3606.39
2013	instab_hooker-5	0	0	23	17	23	21	27	39	205	0	0.17	0.35	29.41	3606.44
2013	instab_hooker-6	0	0	39	42	58	39	53	62	62	0	0.87	1.71	69.37	3606.33
2013	instab_hooker-7	0	0	9	48	43	44	60	59	92	0	0.61	1.19	0.00	2368.31
2013	instab_hooker-8	0	0	4	26	38	36	55	73	123	0	0.35	0.69	97.96	3606.28
2014	instab_hooker-0	0	0	1	8	20	25	31	84	202	0	0.00	0.00	0.00	6.42
2014	instab_hooker-1	0	0	0	10	21	24	32	79	205	0	0.00	0.01	0.01	94.66
2014	instab_hooker-2	0	0	0	12	23	20	28	84	204	0	0.01	0.01	2.85	3607.19
2014	instab_hooker-3	0	0	0	15	28	25	16	77	210	0	0.03	0.08	11.02	3607.35
2014	instab_hooker-4	0	0	0	19	43	31	26	33	219	0	0.14	0.30	27.87	3607.40
2014	instab_hooker-5	0	0	0	17	43	61	69	72	109	0	0.42	0.84	0.00	1105.55
2014	instab_hooker-6	0	0	0	4	43	53	53	76	142	0	0.26	0.54	0.00	200.90
2014	instab_hooker-7	0	0	0	1	48	50	75	79	118	0	0.32	0.59	0.00	72.30
2014	instab_hooker-8	0	0	0	5	46	47	49	80	144	0	0.25	0.53	0.00	1020.60

Table 4 continued

Inst. year	Criterion	Students per priorities									Instab.	Utility	Envy	Gap	Seconds
		9	8	7	6	5	4	3	2	1					
2015	instab_hooker-0	0	0	0	1	1	12	37	103	268	0	0.01	− 0.14	0.00	16.16
2015	instab_hooker-1	0	0	0	0	0	0	82	89	251	0	0.05	− 0.02	0.01	304.59
2015	instab_hooker-2	0	0	0	0	0	0	162	112	148	0	0.34	0.66	0.00	14.59
2015	instab_hooker-3	0	0	0	0	0	0	147	128	147	0	0.32	0.63	0.00	13.70
2015	instab_hooker-4	0	0	0	0	0	0	157	130	135	0	0.35	0.66	0.00	13.73
2015	instab_hooker-5	0	0	0	0	0	0	128	138	156	0	0.28	0.49	0.00	14.08
2015	instab_hooker-6	0	0	0	0	0	0	147	129	146	0	0.32	0.60	0.00	13.66
2015	instab_hooker-7	0	0	0	0	0	0	137	130	155	0	0.29	0.57	0.00	15.18
2015	instab_hooker-8	0	0	0	0	0	0	147	134	141	0	0.33	0.59	0.00	14.44
2016	instab_hooker-0	0	0	1	1	3	10	22	78	301	0	0.00	− 0.16	0.00	12.36
2016	instab_hooker-1	0	0	2	1	4	8	23	73	305	0	0.00	− 0.15	0.00	50.88
2016	instab_hooker-2	0	0	0	0	14	14	14	68	306	0	0.03	− 0.08	0.00	60.77
2016	instab_hooker-3	0	0	0	0	23	22	23	33	315	0	0.11	0.15	0.01	71.82
2016	instab_hooker-4	0	0	0	0	14	24	33	88	257	0	0.18	0.30	0.00	15.45
2016	instab_hooker-5	0	0	0	0	19	21	30	79	267	0	0.18	0.32	0.00	12.81
2016	instab_hooker-6	0	0	0	0	17	24	46	85	244	0	0.24	0.46	0.00	13.68
2016	instab_hooker-7	0	0	0	0	21	35	56	98	206	0	0.38	0.91	0.00	12.79
2016	instab_hooker-8	0	0	0	0	15	28	48	71	254	0	0.23	0.44	0.00	11.96

The last two columns report the optimality gap when the solver exceeded the time limit of 3600 s and the total running time in seconds of the MILP

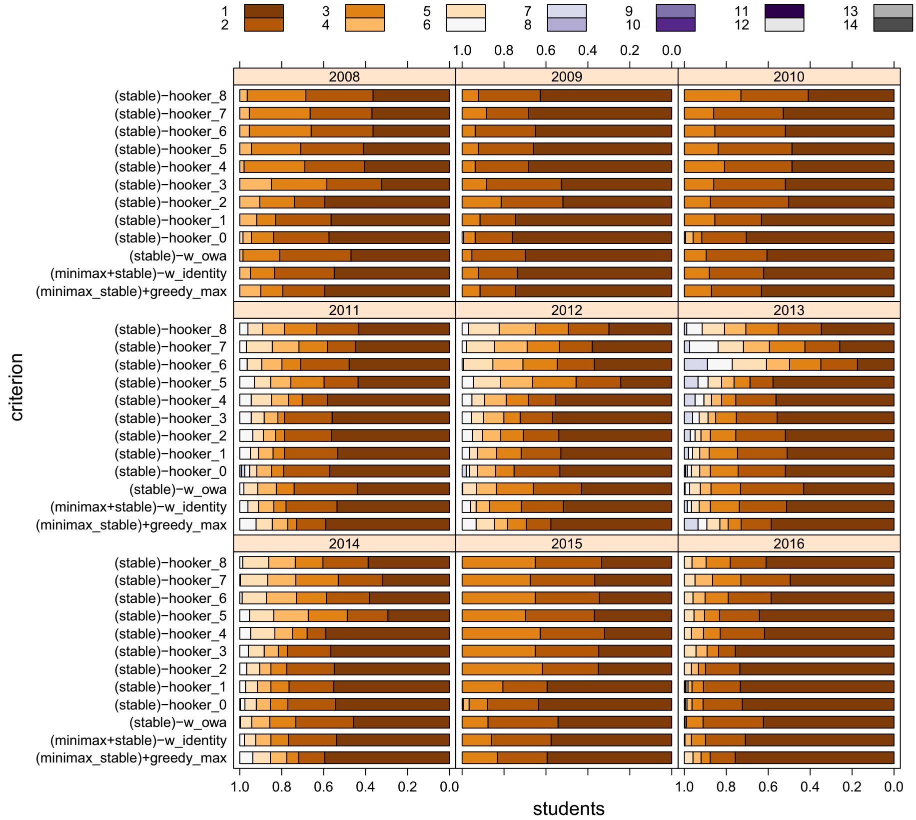

Fig. 3 A stacked barchart showing the proportion of students on each priority for the combined utilitarian-minimax models

- The global utility of the assignments grows for increasing values of τ until a certain value and then it falls again. This pattern was observed also by Hooker and Williams (2012). However, it is hard to predict which value of τ will give the best global satisfaction. (Remember that for the way we defined utility, the global satisfaction is high for low values of global utility.)
- Configurations with values for τ equal to 2, 1, 1, 4, 4, 5, 4, 1, 3 for the instances from 2008 to 2016, respectively, seem to have the peculiar property of maximizing the number of students in the first priority while setting fewer students in the second priority. This is perhaps easier to see from Fig. 3. They are in this sense similar to the exponential weighting scheme and to the greedy maximum matchings (*(stable+minimax)-greedy_max* in Fig. 3).
- With the exception of $\tau = 0$ all configurations seem to achieve competitive results. Specifically, the worst preference value is always the same as found by the *minimax* model (except for $\tau = 0$) and differences among configurations are only to be observed on the distributions and relative utility deviations. Overall it seems that the best models from this table are anyway of similar quality if not worse than those of the three selected combinations from the previous section. Most importantly, in many cases the runs are

truncated before optimality is proved when we used a time limit of 3600 seconds, which shows the higher computational cost of these models.

In conclusion, it is hard to see advantages for the combined utilitarian-minimax models with respect to the best three models of the previous section. Moreover, in our experience, the models are not easy to control via the parameter τ since it is hard to predict the outcome for values of τ different from 0 and 8.

6.4 Enumerating all solutions

Finding all solutions for a certain model may be interesting from a point of view of fairness and robustness. For a guarantee of fairness the selection among the set of optimal solutions has to be done at random with a uniformly distributed probability. In practice, one can resort to take the first solution found, but depending on the implementation of the solver there might be some bias in this case, which is not easy to spot. Also, having several solutions available can be interesting to analyze the difference according to some other measure of quality and to direct the selection accordingly. In Table 5, we show the number of optimal solutions for the three selected models. They can be obtained by iteratively adding cuts (or no-good constraints) to the models once an optimal solution has been found. The cuts enforce that a new optimal solution must have the same quality and differ by the value of at least one variable among the binary variables x_{kj} that define solutions in the model (1)–(5). Given an optimal solution $\bar{x}$, let $\bar{O} := \{(k, j) \mid G_k \in \mathcal{G}, p_j \in L_k, \bar{x}_{kj} = 1\}$ and $\bar{Z} := \{(k, j) \mid G_k \in \mathcal{G}, p_j \in L_k, \bar{x}_{kj} = 0\}$. The constraint to add to the models is then

$$\sum_{(k,j)\in\bar{O}} (1 - x_{kj}) + \sum_{(k,j)\in\bar{Z}} x_{kj} \geq 1.$$

The results, reported in the table, indicate that the number of optimal solutions varies considerably between instances. There is quite some variation with respect to computation times, indicating that it might not always be easy to find all solutions. As expected, the number of solutions admitted by *(stable+minimax)-weighted_identity* is always larger than *(stable+minimax)-weighted_exp*. The model that provides the least number of solutions is *(stable)-weighted_owa*. We expect *(stable+minimax)-greedy_max* to behave similarly. On the other side, we observe that the distributions of the solutions can vary for *(stable+minimax)-weighted_identity* and, to a smaller extent, also for *(stable+minimax)-weighted_exp*.

6.5 The impact of group registrations

In this section we try to study the impact of group registrations on the priority distributions. In particular, we try to assess experimentally two possible conjectures: group registrations worsen the overall quality of the allocation; and students that register in groups have higher chances to obtain a high priority allocation. It was on the basis of these conjectures, together with other considerations of pedagogical character and complexity of the process of group formation (Lu and Boutilier 2012), that the administering committee finally decided to remove the possibility for group registrations in the years 2015 and 2016.

In Fig. 4, we visualize the results of the solutions of the selected combined models when solving the instances with (w) and without (wo) the constraint that students who registered in groups must go in the same team. We observe that there are only minimal differences. For example, in the instances 2012 and 2013 there is a slight improvement of all models when solving the instance without group registrations. Note, however, that in these instances students had to express the same list of preference as the other components of the group.

Table 5 All optimal solutions: for the three selected models the table report the number of solutions and the total time in seconds to find them all

Inst. Year	Criterion	# sol.	sec.	Variation range of stds per priority 6	5	4	3	2	1
2008	(stable)-weighted_owa	1	4.09	0	0	0	0	0	0
2008	(stable+minimax)-weighted_identity	303	3614.89	0	0	2	6	9	5
2008	(stable+minimax)-weighted_exp	336	3602.43	0	0	0	0	0	0
2009	(stable)-weighted_owa	1	2.25	0	0	0	0	0	0
2009	(stable+minimax)-weighted_identity	474	3610.22	0	0	0	3	6	3
2009	(stable+minimax)-weighted_exp	94	114.93	0	0	0	0	0	0
2010	(stable)-weighted_owa	2	5.03	0	0	0	0	0	0
2010	(stable+minimax)-weighted_identity	342	3601.54	0	0	0	5	10	5
2010	(stable+minimax)-weighted_exp	390	3611.88	0	0	0	0	0	0
2011	(stable)-weighted_owa	1	6.72	0	0	0	0	0	0
2011	(stable+minimax)-weighted_identity	239	3623.01	2	3	4	8	11	7
2011	(stable+minimax)-weighted_exp	285	3610.32	0	2	3	1	0	0
2012	(stable)-weighted_owa	1	9.95	0	0	0	0	0	0
2012	(stable+minimax)-weighted_identity	43	200.86	0	0	3	4	1	2
2012	(stable+minimax)-weighted_exp	15	47.26	0	0	0	0	0	0
2013	(stable)-weighted_owa	2	22.07	0	0	0	0	0	0
2013	(stable+minimax)-weighted_identity	170	3608.18	1	1	8	9	9	8
2013	(stable+minimax)-weighted_exp	13	107.11	2	1	0	2	2	1
2014	(stable)-weighted_owa	1	15.29	0	0	0	0	0	0
2014	(stable+minimax)-weighted_identity	166	3620.66	1	2	3	10	18	10
2014	(stable+minimax)-weighted_exp	53	611.72	0	0	0	0	0	0
2015	(stable)-weighted_owa	1	26.46	0	0	0	0	0	0
2015	(stable+minimax)-weighted_identity	149	3639.46	0	0	0	6	12	6
2015	(stable+minimax)-weighted_exp	180	3620.64	0	0	0	2	3	1
2016	(stable)-weighted_owa	2	47.57	0	0	0	0	0	0
2016	(stable+minimax)-weighted_identity	23	298.29	0	1	4	7	14	8
2016	(stable+minimax)-weighted_exp	24	380.95	0	0	0	0	0	0

The last six columns report the range of variation in number of students per priority

Hence, the flexibility of not having group registrations might be not fully exploited with these instances even removing the group constraint, because students might have expressed their preferences in compliance with the group rather than with their actual desires. Therefore, we include in the figure also the instances of 2015 and 2016. In these instances, group registrations were not allowed. Comparing with the distributions of the other instances it might seem that the overall situation indeed improves. However, we cannot know whether the improved situation is due to the removal of groups or to an inherent easiness of the instances. Indeed, instances 2009 and 2010 seem also to yield good solutions as those in 2015 and 2016.

In Fig. 5, we analyze the distributions of students in the solutions of the model with the group constraint. In particular, we separate students according to the size of the group under which they registered. It does seem that the distributions of students in groups of size larger than one are better. Indeed, these distributions seem to use less priorities and their cumulative fractions of students are more shifted towards left. There are however exceptions, like on instance 2012.

7 Discussion

An aspect in the design of allocation mechanisms, that we did not take into account but that is usually very important in economics literature, is the manipulability of the assignment process. Except for the lottery algorithm, none of the optimization criteria discussed has

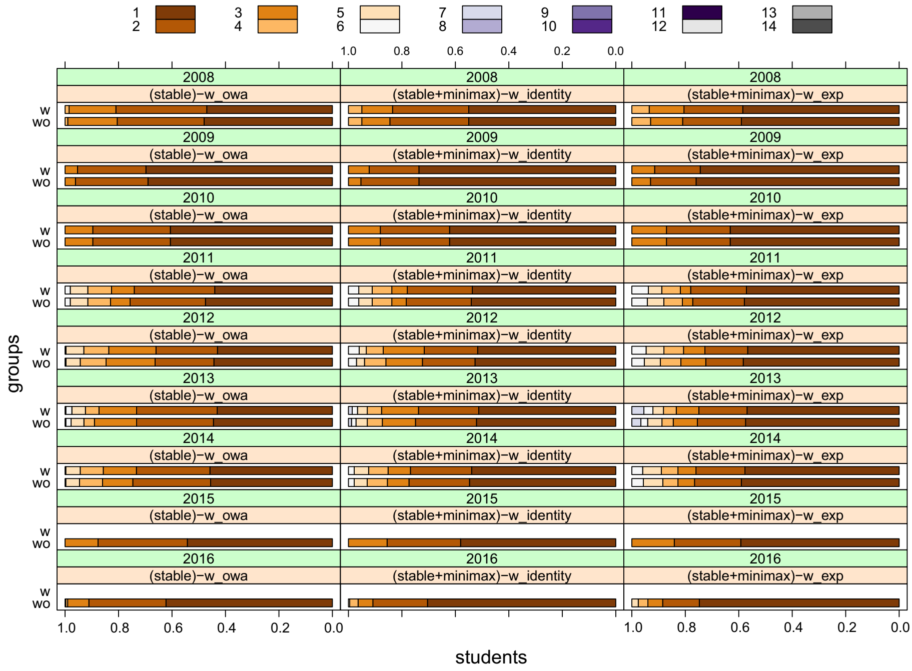

Fig. 4 A stacked barchart to assess the impact of group registrations on the distributions of students for the selected combined models. On the y-axis a "w" indicates the models that are solved with the group constraint active and a "wo" the models are solved without that constraint

strategy-proof guarantees. That is, we have no element to exclude that students could be better off by submitting false preferences. For example, suppose that two students are competing for a place at a medium popular topic that they both prefer the most. If there is also a very popular topic, then it can perhaps be a good strategy for one of these students to put that very popular topic first in his/her list, as it is unlikely that the optimal solution would assign that topic to the manipulating student. However, by this manipulation his/her chances to get the medium popular topic can increase compared to the other student who submits his/her true preference list.

In our practical context, we argue that the following elements make at least difficult the design of a manipulation strategy: (i) the lack of historical data on the popularity of topics, due to the fact that topics are renewed every year, and the fact that preferences on the current year are not made public; (ii) the difficulty in predicting the assigned priority due to the NP-hardness of the problem and the randomization of the selection among solutions. As a further element, the identity weighting scheme may be constituting a safer alternative with respect to strategy proof, since it provides a larger set of solutions among which random selection can take place.

Note that strategy-proofness comes at the cost of a loss in collective welfare (Budish and Cantillon 2012). It is however a serious issue and, in practice, strategy proof solutions have been preferred to Pareto optimal solutions (Abdulkadirolu et al. 2009). It might then be interesting as a future work to include in the analysis assignment mechanisms that try to improve over this trade-off. See, for example, Budish (2011) and Budish and Cantillon (2012).

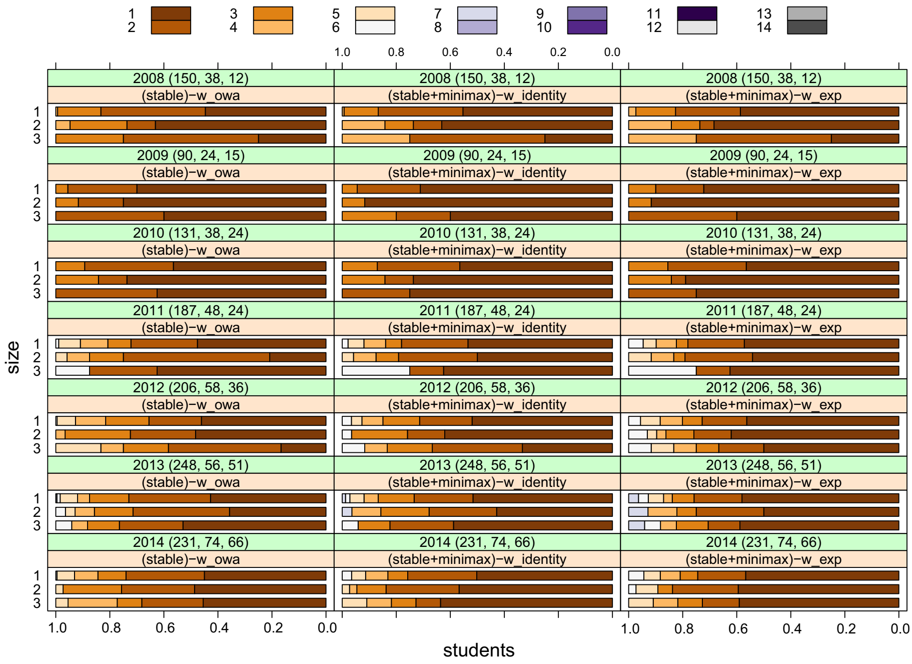

Fig. 5 A stacked barchart to assess the impact of group registrations on the likelihood of getting a certain priority. In the strip text in parenthesis close to the year we give the total number of students who registered in groups of size 1, 2 and 3, respectively. We removed the instances without group registrations

Another aspect to be mentioned is that a different definition of fairness is possible with respect to the one we have adopted in this paper. Indeed, fairness can be defined also as the criterion that two students with the same preference list should have the same chance of getting their preferred topics. There is no deterministic mechanism for fair allocation in this sense, but there are stochastic procedures that can provide ex ante fairness. The lottery based greedy algorithm described in Sect. 4, is a kind of stochastic process that might be ex ante fair, but has serious issues in terms of overall welfare of the solution. However, Bogomolnaia and Moulin (2001) have shown that there are several possible mechanisms for ex ante fair solutions, where the lottery selects solution randomly over the set of efficient assignments by keeping the marginal probabilities of the students for getting their specific choices. Moreover, recent papers show that the lotteries over matchings can be chosen in such a way that further aims are fostered, such as cohesion in group choice (Ashlagi and Shi 2014). Therefore, it might be interesting to design and test alternative ways of combining the criteria used in this paper with stochastic selection, improving the guarantees of equal treatment of equals.

8 Conclusions and future work

We studied the problem of allocating students into teams working on project topics. The number of teams available is limited and there are constraints on the team formation due to team capacity and group registrations (i.e., students willing to be in the same team). The problem arose at our institution and has an important impact in the satisfaction of students for their education. The main challenge of the problem is the requirement of maximizing individual and collective welfare. Our focus has been on comparing, in terms of computational cost and solution quality, a few alternative criteria for obtaining such a goal.

As common in social choice theory, we formulated individual welfare as a minimax utility criterion and collective welfare as a total weighted student utility criterion. An alternative approach that subsumes the minimax approach and uses an alternative way of achieving collective welfare is lexicographic optimization. We formulated a leximin approach that tries to minimize the number of students with assignments in the worse priorities finding generous maximum matchings. Using the result by Yager (1997), we reformulated the leximin approach as an ordering weighted average approach, which simplifies the solution process, requiring to solve a single MILP problem instead of a series of such problems. At the other end of the spectrum we considered another lexicographic approach that greedily tries to maximize the number of students in the first priority then on the second and so on, thus finding greedy maximum matchings. Weighted utility approaches allow to span in the range that has the generous maximum assignments and greedy maximum assignments as extremes. Further, we added and adapted the concepts of stability and envy-freeness, which are well-known in the literature on the related problems of bipartite matching with two-sided preferences and of resource allocation in multi-agent systems, respectively. These criteria aim at increasing the acceptability of the consequent assignments by preventing local improvements that an individual might easily spot looking only at the current situation and his/her preferences. In our practical experience, stability has been a very important factor for the acceptance by the students of the assignments that we delivered. Finally, we looked at Hooker and Williams' alternative proposal to address the trade-off between equality and global utility.

In the analysis of results we focused on the distribution of students over the assigned preference values, on measures of stability and envy, and on the running time. Individually taken, all criteria exhibit some pitfalls with respect to the criteria that they are not addressing. But the empirical evidence collected on the implications of these basic criteria provided us valuable information for the design of novel combinations of these criteria. The weighted utility with identity and exponential weighting schemes and the greedy maximum matching were combined with the minimax approach and with stability. The leximin approach, solved via ordering weighted average, was also combined with stability. Instead, we discarded the envy-freeness criterion because its model turned out computationally much harder to solve to optimality with respect to the others and because it anyway provided assignments very similar to those of the identity weighting scheme in the weighted approach. The way to combine utilitarianism and minimax proposed by Hooker and Williams (2012) led also to models that are computationally difficult to solve. Moreover, we experienced that the effect of the threshold parameter was rather difficult to predict making it hard for us to decide its value a-priori.

The combination of criteria incurred a price in terms of student distributions over preference values. However, in our case, this loss was less important than the advantages gained in terms of fairness and stability. We actually eneded up treating stability as a constraint, because it was always achievable in our instances. Similarly, we used the results of the minimax model as a strict constraint on the weighted utility models.

The inclusion of stability and minimax in the constraint set led to solutions with different distribution profiles for the three types of weighting schemes used in the objective function, i.e., OWA, identity, or exponential schemes, while the greedy maximum approach led to distribution profiles very similar to those of the exponential scheme. The three solutions of the weighting schemes do not dominate each other and we presented them all to the administering committee. Although they all met the satisfaction of the administering committee, the committee ended up choosing the ones returned by exponential weighting scheme with the reasoning that students may perceive as more important to go from the second to the first priority rather than going from the sixth to the fifth. The generous maximum matchings approach in the guise of OWA scheme was deemed less attractive because it always ended up with fewer students in their first priority.

For our instances with up to 422 students and 160 project teams the three models are easily solvable in the order of tens of seconds with a state-of-the-art MILP commercial solver. Note that in some recent articles (Harper et al. 2005; Geiger and Wenger 2010; Srinivasan and Rachmawati 2008) the problem was instead judged too hard for exact approaches. A further advantage of a MILP approach with respect to other approaches like specialized algorithms or heuristics is that, it made it easy to model a few other side constraints not mentioned so far in this paper. For example, we had to deal with the fact that projects and students are grouped into types and that there are compatibility constraints between these types (this constraint was indeed implemented in the experiments reported in this paper). On the other hand, a MILP approach entails a loss of transparency. While the lottery procedure could be described entirely in the official course description cited in Sect. 4, a MILP approach cannot. Our approach was to substitute the procedural description with a declarative presentation of how assignments must look like. We finally opted for describing the procedure for finding greedy maximum matchings: from the pool of feasible assignments at each stage h for $h = 1, \dots, \Delta$ the assignments that do not maximize the students in the hth priority are discarded; among the assignments left after all these removals, the assignment to be put in operation is one selected at random (randomness is entailed by the reshuffling of the input data to the MILP solvers).

We believe that the learnings summarized above can be helpful also in other practical applications involving preferences. For example, it would be interesting to extend this work to the paper-reviewer assignment problem. In Garg et al. (2010), the authors provide an approximation algorithm and it would be interesting to observe how a MILP approach to finding optimal solutions would perform in that context. Moreover, the combined criteria reviewed here could be used also in that problem improving the practical appeal of the solutions. More generally, we have collected ideas from different scientific areas and have shown how these ideas can be applied in practice. In particular, we described the relationship between weighted utilitarian, lexicographic and utilitarian-minimax approaches. The main learning here is that at least two solutions might be worth considering, the generous maximum matching and the greedy maximum matching. They can be attained by solving a series of MILP problems or by using an appropriate weighting scheme in a single MILP problem. Other weighting schemes and versions of utilitarian-minimax approaches constitute different variations of these schemes and provide solutions between those two extremes.

We did not analyze the manipulability of student assignments in theoretical terms. However, we argue that the NP-hardness of the assignment problem and the random selection among solutions of equal quality (as defined by the criteria used) make at least difficult the design of a malicious strategy in the expression of preferences. We analyzed experimentally the conjecture that registering in groups rather than individually should increase the chances of obtaining a high priority topic. We found some small evidence for this conjecture while perhaps more evidence that students attain better worst priority when registered in groups than when registered individually. However, our experiments in this regard are not sufficient to provide statistical significance.

There are a few issues that might deserve further investigation. From a design point of view, it would be interesting to allow for a partial order rather than a strict order in the expression of preferences. A partial order should lead to solutions of better quality and thus improve the perceived satisfaction by students (Fragiadakis and Troyan 2014). Another issue that might deserve consideration is disruption management: it has happened that after the assignment has been made public, a project could not be offered anymore due to the unforeseen absence of a supervisor. In this case, the goal becomes finding an alternative good solution with the minimum amount of changes.

References

Abdulkadirolu, A., Pathak, P. A., & Roth, A. E. (2009). Strategy-proofness versus efficiency in matching with indifferences: Redesigning the NYC high school match. *American Economic Review*, *99*(5), 1954–78. https://doi.org/10.1257/aer.99.5.1954.

Abraham, D. J., Irving, R. W., & Manlove, D. F. (2007). Two algorithms for the student-project allocation problem. *Journal of Discrete Algorithms*, *5*(1), 73–90. https://doi.org/10.1016/j.jda.2006.03.006.

Anwar, A. A., & Bahaj, A. (2003). Student project allocation using integer programming. *IEEE Transactions on Education*, *46*(3), 359–367. https://doi.org/10.1109/TE.2003.811038.

Arulselvan, A., Cseh, Á., Groß, M., Manlove, D.F., & Matuschke, J. (2016). Matchings with lower quotas: Algorithms and complexity. CoRR arxiv:1412.0325. Preliminary version appeared at ISAAC 2015.

Ashlagi, I., & Shi, P. (2014). Improving community cohesion in school choice via correlated-lottery implementation. Working paper.

Atkinson, A. B. (1970). On the measurement of inequality. *Journal of Economic Theory*, *2*(3), 244–263.

Bertsimas, D., Farias, V. F., & Trichakis, N. (2012). On the efficiency-fairness trade-off. *Management Science*, *58*(12), 2234–2250. https://doi.org/10.1287/mnsc.1120.1549.

Biró, P., Fleiner, T., Irving, R. W., & Manlove, D. F. (2010). The college admissions problem with lower and common quotas. *Theoretical Computer Science*, *411*(34), 3136–3153. https://doi.org/10.1016/j.tcs.2010.05.005.

Biró, P., & McDermid, E. (2014). Matching with sizes (or scheduling with processing set restrictions). *Discrete Applied Mathematics*, *164*(Part 1), 61–67. https://doi.org/10.1016/j.dam.2011.11.003.

Bogomolnaia, A., & Moulin, H. (2001). A new solution to the random assignment problem. *Journal of Economic Theory*, *100*(2), 295–328. https://doi.org/10.1006/jeth.2000.2710.

Bouveret, S., & Lang, J. (2008). Efficiency and envy-freeness in fair division of indivisible goods: Logical representation and complexity. *Journal of Artificial Intelligence Research*, *32*, 525–564. https://doi.org/10.1613/jair.2467.

Brams, S. J., & Taylor, A. D. (1996). *Fair division: From cake-cutting to dispute resolution*. Cambridge: Cambridge University Press.

Budish, E. (2011). The combinatorial assignment problem: Approximate competitive equilibrium from equal incomes. *Journal of Political Economy*, *119*(6), 1061–1103. https://doi.org/10.1086/664613.

Budish, E., & Cantillon, E. (2012). The multi-unit assignment problem: Theory and evidence from course allocation at Harvard. *American Economic Review*, *102*(5), 2237–71. https://doi.org/10.1257/aer.102.5.2237.

Dye, J. (2001). A constraint logic programming approach to the stable marriage problem and its application to student-project allocation. In *Proceedings of the sixth international workshop on computer-aided software engineering*.

El-Atta, A., & Moussa, M.I. (2009). Student project allocation with preference lists over (student, project) pairs. In *Second international conference on computer and electrical engineering, 2009. ICCEE '09* (Vol. 1, pp. 375–379). https://doi.org/10.1109/ICCEE.2009.63.

Fragiadakis, D. E., & Troyan, P. (2014). Improving welfare in assignment problems: An experimental investigation. http://erl.tamu.edu/working-papers/. Working paper at Economic Research Laboratory, Texas A&M University.

Garg, N., Kavitha, T., Kumar, A., Mehlhorn, K., & Mestre, J. (2010). Assigning papers to referees. *Algorithmica*, *58*(1), 119–136. https://doi.org/10.1007/s00453-009-9386-0.

Geiger, M. J., & Wenger, W. (2010). On the assignment of students to topics: A variable neighborhood search approach. *Socio-Economic Planning Sciences*, *44*(1), 25–34. https://doi.org/10.1016/j.seps.2009.03.001.

Gusfield, D., & Irving, R. (1989). *The stable marriage problem: Structure and algorithms*. Cambridge: MIT Press.

Harper, P. R., de Senna, V., Vieira, I. T., & Shahani, A. K. (2005). A genetic algorithm for the project assignment problem. *Computers & Operations Research*, *32*(5), 1255–1265. https://doi.org/10.1016/j.cor.2003.11.003.

Hooker, J. N., & Williams, H. P. (2012). Combining equity and utilitarianism in a mathematical programming model. *Management Science*, *58*(9), 1682–1693. https://doi.org/10.1287/mnsc.1120.1515.

Iwama, K., & Miyazaki, S. (2008). A survey of the stable marriage problem and its variants. In *International conference on informatics education and research for knowledge-circulating society (ICKS 2008)* (pp. 131–136). https://doi.org/10.1109/ICKS.2008.7.

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. https://doi.org/10.1016/j.jda.2012.02.001.

Kagel, J. H., & Roth, A. E. (Eds.). (1997). *The handbook of experimental economics*. Princeton: Princeton University Press.

Lu, T., & Boutilier, C.E. (2012). Matching models for preference-sensitive group purchasing. In *Proceedings of the 13th ACM conference on electronic commerce, EC '12* (pp. 723–740). ACM, New York, NY, USA. https://doi.org/10.1145/2229012.2229068.

Manlove, D. F. (2013). *Algorithmics of matching under preferences. Series on theoretical computer science* (Vol. 2). Singapore: World Scientific.

Manlove, D. F., & O'Malley, G. (2008). Student-project allocation with preferences over projects. *Journal of Discrete Algorithms*, *6*(4), 553–560. https://doi.org/10.1016/j.jda.2008.07.003.

Moulin, H. (2003). *Fair division and collective welfare*. Cambridge: The MIT Press.

Ogryczak, W., Pióro, M., & Tomaszewski, A. (2005). Telecommunications network design and max–min optimization problem. *Journal of Telecommunications and Information Technology*, *3*, 43–56.

Perach, N., Polak, J., & Rothblum, U. G. (2008). A stable matching model with an entrance criterion applied to the assignment of students to dormitories at the technion. *International Journal of Game Theory*, *36*(3), 519–535. https://doi.org/10.1007/s00182-007-0083-4.

Proll, L. G. (1972). A simple method of assigning projects to students. *Operational Research Quarterly (1970–1977)*, *23*(2), 195–201.

Rawls, J. (1971). *A theory of justice*. Cambridge, MA: Harvard University Press.

Roth, A. E. (1984). The evolution of the labor market for medical interns and residents: A case study in game theory. *Journal of Political Economy*, *92*(6), 991–1016.

Srinivasan, D., & Rachmawati, L. (2008). Efficient fuzzy evolutionary algorithm-based approach for solving the student project allocation problem. *IEEE Transactions on Education*, *51*(4), 439–447. https://doi.org/10.1109/TE.2007.912537.

Tempkin, L. (1993). *Inequality*. New York: Oxford University Press.

Teo, C., & Ho, D. J. (1998). A systematic approach to the implementation of final year project in an electrical engineering undergraduate course. *IEEE Transactions on Education*, *41*(1), 25–30. https://doi.org/10.1109/13.660783.

Williams, H. P. (2013). *Model building in mathematical programming* (5th ed.). Chichester: Wiley.

Yager, R. R. (1996). Constrained OWA aggregation. *Fuzzy Sets and Systems*, *81*(1), 89–101. https://doi.org/10.1016/0165-0114(95)00242-1.

Yager, R. R. (1997). On the analytic representation of the leximin ordering and its application to flexible constraint propagation. *European Journal of Operational Research*, *102*(1), 176–192. https://doi.org/10.1016/S0377-2217(96)00217-2.

Young, P. (1995). *Equity: In theory and practice*. Princeton, NJ: Princeton University Press.