Allocating Students to Multidisciplinary Capstone Projects Using Discrete Optimization

Thomas L. Magnanti[*] Karthik Natarajan[†]

First Version: July 2017, Revised Version: October 2017

Abstract

We discuss an allocation mechanism of capstone projects to senior year undergraduate students that has been implemented at the recently established Singapore University of Technology and Design (SUTD). A distinguishing feature of these projects is their multidisciplinarity - each project must involve students from at least two different disciplines. This is an instance of a bipartite many-to-one matching problem with one-sided preferences with additional lower and upper bounds on the number of students from the different disciplines that need to be matched to projects. This leads to challenges in applying many existing algorithms. We propose the use of discrete optimization to find an allocation that incorporates both efficiency and fairness considerations. This provides flexibility in incorporating side constraints that is often introduced in the final project allocation using inputs from the various stakeholders. Over a three year period from 2015 to 2017, the average rank of the project allocated to the student is roughly halfway between their top two choices, with around 78% of the students assigned to projects in their top three choices. We discuss practical design and optimization issues that arise in developing such an allocation.

Keywords: project allocation; discrete optimization; multidisciplinary

A capstone in architectural terms is the final stone placed in the center of an arch to hold the entire construction together in a stable manner. In an academic environment, a capstone course

[*] Singapore University of Technology and Design and MIT, Room 32-D784, Cambridge, MA 02139. Email: magnanti@sutd.edu.sg

[†] Engineering System and Design, Singapore University of Technology and Design, Singapore 487372. Email: karthik_natarajan@sutd.edu.sg

integrates the entire education of undergraduate students in a final culminating experience. Increasingly many universities have adopted a signature capstone course for students either during the last semester or the last two semesters of the undergraduate curriculum. Hauhart and Grahe [13] provide a detailed exposition on the challenges and best practices for the successful design and implementation of capstone courses, ranging from appropriate learning objectives, to format of the classes, to course assessment. They estimate that almost seventy five percent of American institutions today offer a capstone experience. Gorman [12] provides a thorough discussion on his course experience which offered live consulting style projects as capstones to undergraduate students in operations management at the University of Dayton's School of Business Administration. While such projects carry risks, he postulates from the experience over a eight year period that a well-designed capstone experience significantly benefits both the students and the companies. While the majority of the capstones offered in universities are discipline-specific, Hauhart and Grahe [13] estimate that around fifteen percent of the universities today offer multidisciplinary capstone projects. Rowles et al. [23] distinguish among the two types of capstone projects using the terminology of "magnets" (discipline-specific projects that act like a magnet in pulling together the richness of content from the discipline) and "mountaintops" (multidisciplinary projects where students from two or more disciplines ascend to the summit using the experience from different, disciplinary perspectives). Prior research in [13] and [23] also highlights that designing multidisciplinary capstones is more challenging than discipline specific capstones. In this paper, we focus on a related important and practical problem that arises in such a context - the allocation of students to multidisciplinary capstone projects.

Capstone Projects at SUTD

We discuss the capstone project allocation problem and some of the key challenges faced in its implementation at the Singapore University of Technology and Design (SUTD). SUTD ran its inaugaral capstone course in 2015 and currently offers the course once a year. The capstone course runs over the final two terms of the undergraduate program and is an important part of the graduation requirements of all undergraduate students at the university. SUTD is a recently established public university in Singapore. It enrolled its first cohort of undergraduate students in 2012. The university has four disciplines (also referred to as pillars) in which the undergraduate students major. These disciplines are Architecture and Sustainable Design (ASD), Engineering Product Development (EPD), Engineering Systems and Design (ESD) and Information Systems

Technology and Design (ISTD). The mission of the university is “to advance knowledge and nurture technically-grounded leaders and innovators to serve societal needs, with a focus on design, through an integrated multidisciplinary curriculum and multidisciplinary research”. SUTD partly achieves this by requiring that every capstone project brings together students from at least two disciplines to work on projects sourced primarily from companies operating in Singapore. By bringing together engineers from different disciplines and architects to work on real-world problems, the objective is to let students have an experience of the real work environment, develop innovative designs that might otherwise be inconceivable and develop collaboration and communication skills. A designated capstone office in the university sources projects from industry partners and works closely with a team of faculty and industry mentors to scope the projects to meet the learning outcomes of the capstone course. Around ninety percent of the projects in the capstone are sourced from companies while the remaining ten percent are entrepreneurial and proposed by students.

In this paper, we focus on the design of the capstone project allocation mechanism for the industry projects at SUTD, and highlight the flexibility of discrete optimization to solve this problem. We next discuss some of the key concerns and challenges that were identified from discussions with the various stakeholders which includes the university senior administration, the capstone office, the capstone faculty, students and industry mentors:

(a) Efficiency: Students at SUTD graduate with a basic degree from one of the four disciplines but also have the possibility of choosing focus tracks that provide further specialization opportunities within their discipline. Since the capstone projects range across a wide range of industries, students often tend to have strong preferences for some projects over others based on their specialization and interests. Hence, eliciting student preferences for projects and using it for allocation rather than just allocating projects randomly is critical for efficiency. At SUTD, we asked students to rank order their top ten preferred projects from the list of available projects. Clearly, if students are assigned to work on capstone projects that do not interest them, it is natural to expect these projects will be less successful. Hence, an important criterion is to have an efficient allocation which would maximize the chances of success of the capstone projects. While the industry mentors provided inputs on the mix of disciplines that are suitable for a particular project, they did not provide explicit preferences on individual students. Given the scale and the number of students in the undergraduate program, this was not deemed to be a feasible exercise in identifying an efficient allocation.

(b) Fairness: The project allocation should be perceived to be fair by the students. While fairness is a subjective criterion, it was important to give students equal chances in obtaining their preferred capstone projects. The capstone faculty decided not to use previous academic performance or other criteria such as gender or nationality of students in determining the final project allocation. Incorporating both efficiency and fairness considerations in resource allocation has been considered in other applications such as kidney allocation (see Bertsimas, Farias and Trichakis [5]) and also forms a key consideration of the capstone allocation at SUTD.

(c) Multidisciplinary projects: Every launched project must be multidisciplinary and involve students from at least two disciplines.

(d) Flexibility: The allocation method had to be designed to be flexible enough to incorporate any additional new constraints that might arise during the final project allocation meeting with the capstone committee.

Related Work

In this section, we review some of the successful applications of discrete optimization to solve project allocation problems at other universities while highlighting some of the key differences from our experience at SUTD. One such example is the allocation of internships to students at the MIT Leaders for Global Operations (LGO) program. Every student in the MIT LGO program works on an internship for six months. These projects are carefully scoped by a committee so as to meet the academic expectations from the MIT School of Engineering and the MIT Sloan MBA program. Each year around fifty students join the MIT LGO program and between fifty to sixty internships are scoped for the students to consider. Each student then interviews for fifteen to twenty five internships and ranks them in order of preference. Likewise the partner companies, rank the students who are interested in their internships. By using an optimization method that aims to maximize everyone's desire, the program has been able to ensure that around 75% of the time, students are allocated to one of their top three internship preferences[1]. While this is an example of a matching problem with two-sided preferences, the capstone allocation at SUTD differs in two ways. First, SUTD uses only one-sided preferences with students rating projects. This is mainly due to scale restrictions since we are dealing with hundreds of undergraduate students, rather than fifty postgraduate students. Obtaining companies preferences on undergraduate students is rather

[1] Source: https://lgo.mit.edu/curriculum/internships/

arduous in this case. Second, each project involves students from different disciplines and hence project mix is an important component of the capstone allocation in contrast to student internships which are primarily individual experiences. Table 1 summarizes other examples of universities using discrete optimization to allocate students to groups or projects. The last application in Table 1 is the closest to and shares some common features with the capstone allocation problem at SUTD. Their model allows for students to rank order their top five projects and assign three to six students from different disciplines to collaborate on a project. Their formulation uses penalty terms in the objective function to make a tradeoff between student preferences and violating the bounds on the number of students from a discipline allotted to a project. Thus it does not enforce that hard constraints on the project size is enforced. Furthermore, industry sponsors can select up to two students who they would like to have on the project based on an open house interaction. If students accept this invitation, then the model accounts for it in the allocation. However at SUTD, we are dealing with only one-sided preferences.

Table 1: This table provides a summary of some of the other universities where discrete optimization has been used for allocation purposes and the difference from the SUTD application.

Authors	University	Application	How does the application differ the SUTD application?
Baker and Powell [3]	Tuck School of Business, Dartmouth College	Assign students to sections and section members to study groups	Student preferences is not part of the input
Anwar & Bahaj [2]	Civil & Environmental Engineering, University of Southampton	Assign students to group projects in their fourth year of study	No hard constraints on multidisciplinarity
Kirkwood [17]	W. P. Carey School of Business, Arizona State University	Select project teams for MBA industrial projects course	Students from only a single discipline
Cutshall, Gavirneni & Schultz [10]	Kelley School of Business, Indiana University	Form teams across four disciplines for case studies in an integrated core	Student preferences is not part of the input
Lopes, Aronson, Carstensen & Smith [18]	College of Engineering, University of Arizona	Allocate senior design group projects	Soft constraints on multidisciplinarity with two sided preferences

A related stream of literature is the study of matching problems with preferences which has its origins in the pioneering work of Gale and Shapley [11] on the stable marriage problem. More closely related to this paper is the house allocation model proposed by Hylland and Zeckhauser [16] which allocates a set of houses (objects) to a set of residents (agents), with each agent having a preference list over the set of houses. Randomization is commonly used in such an allocation mechanism for fairness considerations. For example, consider a simple setting with two objects and two agents. Each agent prefers object 1 to object 2. Then a fair assignment is a randomized assignment where agents A and B get objects 1 and 2 with probability 0.5 and agents A and B get objects 2 and 1 with probability 0.5. Abdulkadiroğlu and Sönmez [1] proposed a random priority mechanism (also known as the random serial dictatorship mechanism) that chooses an order of agents from all possible order of agents with equal probability and then the top agent gets to pick his or her top choice, the second agent then gets to pick their top choice among the remaining objects, and so on.

Other techniques that have been proposed to tackle this problem include directly selecting lotteries that the agents should receive (see Hylland and Zeckhauser [16] and Bogmolnaia and Moulin [7]). These techniques however do not include any side constraints. In practical applications, several distributional constraints might arise. For example, when more than one agent can be assigned to an object, constraints might be enforced to ensure diversity on the type of agents that can be allocated to a object (see Budish et. al. [8] for several examples of such constraints in practical matching problems). Budish et. al. [8] propose allocation mechanisms that work with upper bound constraints on the group-specific quotas, but this does not generalize to lower bound constraints. Lower bounds, however, form an important aspect of the constraints in the capstone project allocation problem. Diversity is not optional in our setting, it is mandatory.

Several other matching criterion have also been considered for the house allocation model including a rank-maximal matching, popular matching, and fair matching. Algorithms have been developed to solve these models (see Manlove [20]). Hooker and Williams [15] developed a linear program for resource allocation problems that provides a tradeoff between conflicting objectives of utilitarianism and equity using a single threshold parameter. Our proposed approach for the capstone project allocation model also combines the utilitarian objective with an equity objective albeit in a different manner to find an allocation. Chen et. al. [9] recently introduced a linear programming formulation whose objective is to find a maximum weighted matching in a bipartite graph with degree constraints imposed on the vertices where the overall number of vertices of a particular type matched to a given vertex satisfies an upper bound constraint. However their model does not include lower bounds and fairness considerations which are important for capstone allocation.

Capstone Project Allocation Model

In this section, we discuss the formulation of the capstone project allocation model. We first provide a network representation and then discuss the main steps in identifying the project allocation. The detailed mathematical formulation is provided in the Appendix. To illustrate the network representation of the capstone allocation problem, we consider a simple example in Figure 1 with 6 students, 3 projects and 3 disciplines where we construct a node for each discipline, each student and each project and the solid arcs indicate the possible allocation of students to projects (see Appendix A for a more formal definition of the network). The goal is to match students to project subject to the various considerations listed next.

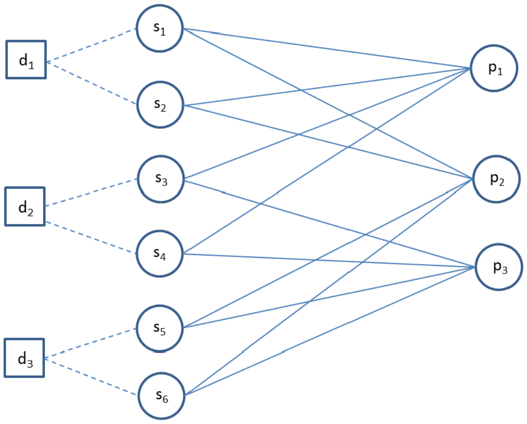


Figure 1: This graph provides a network representation of the capstone project allocation with 6 students, 3 projects and 3 disciplines. Here students s_1 and s_2 are from discipline d_1, students s_3 and s_4 are from discipline d_2 and students s_5 and s_6 are from disciplines d_3. This is denoted by the dashed lines. Project p_1 can be allocated only to students from disciplines d_1 and d_2, project p_2 can be allocated only to students from disciplines d_1 and d_3 while project p_3 can be allocated only to students from disciplines d_2 and d_3. This is denoted by the solid lines originating from all students of those disciplines to the projects.

Our Approach

We provide the mathematical formulation in Appendix B for the capstone project allocation model. The inputs to the model are the network representation, the lower and upper bounds on the number of students of each discipline that is needed for each project and the student preferences for projects (see (1)). The student preferences are converted to utility values on the arcs in the network where the value is set to K for each student's topmost preferred project, $K-1$ for their second most preferred project, down to 1 for the project ranked the lowest in their list. We set the utility value to be negative infinity (a very large negative number denote by $-M$ for practical purposes) for all other projects that might be assigned to the student but the student does not rank (least preferred set of projects). At SUTD, we asked each student to provide a ranking of their top most preferred projects and hence set $K = 10$. The central capstone office collects this data on student preferences. Students are provided with a brief description of the projects and the mix of disciplines that each project needs. The upper and lower bound information on the number of students of the various disciplines needed for each project is however not shared with the students. This helps to partly prevent strategic behavior of students in forming groups before the project allocation since the upper bound on the number of students of each discipline needed for the project is unknown to them. The decision variables are two sets of binary variables which corresponds to which project is assigned to a student and which project is launched (see (2)-(3)). The objective function in (4) is the total utility (efficiency) given by the sum of the utilities of the projects assigned to students which is maximized. Constraints (4a) and (4e) ensures that every student is allocated to a single project. Constraints (4b) and (4f) ensures that a student is assigned to a project only if the project is launched. Constraints (4c) and (4d) ensures that the number of students of the different disciplines in a project that is allocated lies between the prescribed lower and upper bounds. We assume that the projects are scoped so that there is a feasible matching of students to projects that satisfies these constraints.

An important property that any optimal solution to the problem satisfies is that it is Pareto efficient. This is important from a practical perspective since it implies that once an optimal project allocation is found by solving the model, there is no swap among two students that makes both better off which implies the allocation is efficient and stable. An allocation of students to projects, denoted by $\mathcal{M}$, is said to be Pareto efficient if and only if there is no other feasible matching of students to projects denoted by $\mathcal{M}'$ such that no student is worse-off in $\mathcal{M}'$ in comparison to $\mathcal{M}$ and at least one student is better off in $\mathcal{M}'$ in comparison to $\mathcal{M}$. Such a property follows from classic results on the properties of utilitarianism in social welfare functions (see Hindriks and Myles [14]). For completeness, we provide the argument next. Assume that the optimal solution to the discrete optimization problem in (4) given by the matching $\mathcal{M}$ is not Pareto efficient. Then there exists another feasible matching of students to projects denoted by $\mathcal{M}'$ where no student is worse off in $\mathcal{M}'$ in comparison to $\mathcal{M}$ and at least one student is better off in $\mathcal{M}'$ than $\mathcal{M}$. The objective function of the new matching has to be strictly greater than the objective function of the old matching, thereby contradicting the optimality of the solution to (4).

Towards Fairness

In our experience working with the data on student preferences at SUTD, there are typically many Pareto-efficient optimal solutions. For example, assume that each student ranks three projects. In one maximum utility allocation, 4 students are given their top ranked projects, 4 students are given their second ranked project while 6 students are given their bottom ranked project. In a second allocation, 2 students are given their top ranked project, 8 students are given their second ranked project while 4 students are given their bottom ranked project. Both allocations have a total utility of 26. To distinguish among these allocations, we make use of the notion of the lexicographic max-min fairness criterion (see Luss [19], Ogryczak [22]) which prefers the second allocation to the first since fewer people are allocated to their bottom choice. Similarly, consider an example where two students of the same discipline are assigned to two distinct projects in a matching. If both students rank these projects identically, then swapping the project allocation among these two students would still be optimal. In this case, we use randomization to break ties. Our final allocation mechanism is then based on three steps as follows:

(a) Step 1 (Maximum utility): Solve the integer program (4) to find the maximum utility.

(b) Step 2 (Fairness): Start with the projects with utility of 1. Solve the integer program (5) to minimize the number of students assigned to the projects with utility 1 over the set of the maximum utility matchings. Next, solve the integer program (6) to minimize the number of students assigned to the projects with utility of $k = 2$ over the set of the maximum utility matchings with the smallest number of students assigned to less preferred projects and repeat this process up to $k = 9$.

(c) Step 3 (Randomization): In the final step, we break the symmetry in the optimal solution by performing a randomization step. This corresponds to choosing from one of the optimal solutions by randomly perturbing the objective function and solving (7).

Note that our approach needs the solution to $K + 1$ integer programs where $K = 10$ in this example. The integer program in step (1) ensures that the allocation chosen maximizes the total utilitarian social welfare function. In step (2), among the optimal utilitarian allocations from step (1), we first minimize the number of students allocated to the least preferred projects, then minimize the number of students allocated to their second least preferred project and so on, thus providing a allocation that is fair in a lexicographic order. Lastly in step (3), we perturb the objective function of the integer program using randomization with independent and identical random error terms to find a unique allocation with probability one.

Additional Constraints

One of the advantages of the discrete optimization approach is that it provides modeling flexibility in being able to capture side constraints that arise in actual implementations. In this section, we list some of the constraints that can be easily addressed by such a technique based on our own experience at SUTD (see Appendix C):

(a) At least one among a class of projects must be offered: Given that some of the companies provide multiple projects, it is often important to launch at least one of these projects to maintain continued interest from the company in providing capstone projects. This is easily modeled by adding in linear constraints of the type (8). Such a constraint can also model the requirement that a project that has been not launched in the previous year must be launched in the current year.

(b) Similar sized project groups: Given that the projects are performed in groups, it is often important to maintain similar sizes for the projects which can be incorporated with an explicit upper bound and lower bound on the number of students. This is modeled by a set of linear constraints in (9).

(c) Balanced project groups: Given that students from multiple disciplines work on a common project, it is sometimes useful to add in explicit constraints that ensure the number of students assigned to a project from the disciplines are relatively close to each other. For each project, this can be modeled by adding in the constraint (10) which can be modeled as a set of linear constraints in (11).

Actual Implementation

In this section, we provide a summary of the data and the results of the capstone project allocation for the years 2015, 2016 and 2017 at the SUTD.

Data

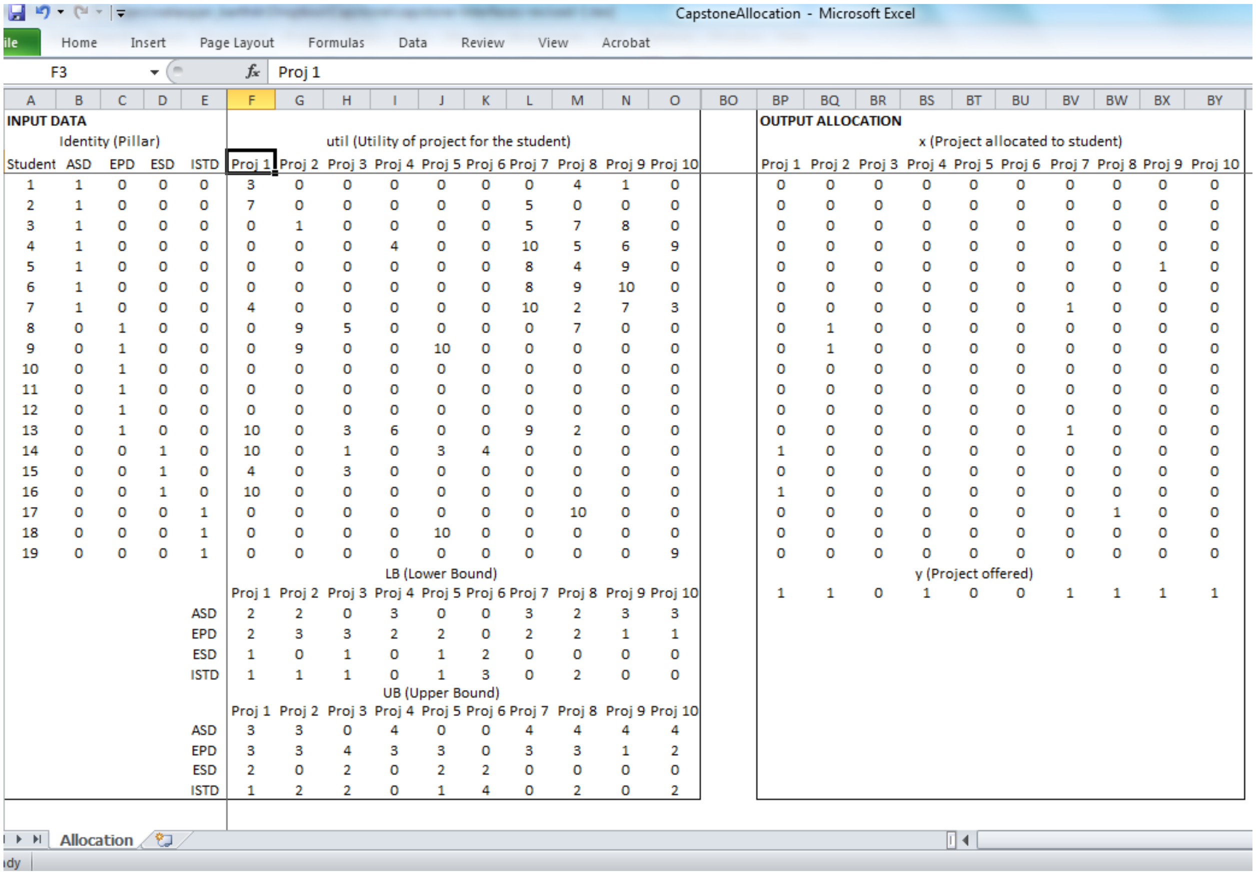
In Table 2, we provide a summary of the details of the students and projects. The data in the table includes the total number of students, the number of students from each of the four disciplines (denoted by ASD (Architecture and Sustainable Design), EPD (Engineering Product Development), ESD (Engineering Systems and Design) and ISTD (Information Systems Technology and Design)), the total number of projects, and the distribution of the projects sizes that integrates multiple disciplines. As the table illustrates, every year there are projects that can involve students from all four disciplines.

Table 2: This table provides data on the students and projects to be assigned for the years 2015 to 2017.

Year	2015	2016	2017
Total number of students to be assigned	219	170	238
Total number of students in ASD	67	59	63
Total number of students in EPD	56	62	76
Total number of students in ESD	67	28	56
Total number of students in ISTD	29	21	43
Total number of projects available	98	61	63
Number of projects that involves two disciplines exactly	39	36	52
Number of projects that involves three disciplines exactly	29	19	8
Number of projects that involves four disciplines exactly	7	3	3
Number of projects that could involve either two or three disciplines	11	3	0
Number of projects that could involve either three or four disciplines	11	0	0
Number of projects that could involve either two, three or four disciplines	1	0	0

The integer programs were solved using the optimization solver CPLEX version 12.6.2.0 with the SolverStudio Excel interface (see Mason [21]). In our experience, the use of the Excel based interface in SolverStudio was particularly useful in communicating the results with the capstone committee. The advantage is that the data and the results are displayed on the same spreadsheet helping the capstone committee better visualize the allocation and provides additional flexibility in incorporating new constraints and to check their influence on the structure of the optimal solution. In Figure 2, we provide a snapshot of the input data and output allocation in the Excel interface and in Figure 3, we provide an illustration of this model developed using the AMPL language and solved with CPLEX using Solver Studio in Excel.

In Figure 4, we display the distribution of the number of students interested in the various



INPUT DATA

	Identity (Pillar)				util (Utility of project for the student)									
Student	ASD	EPD	ESD	ISTD	Proj 1	Proj 2	Proj 3	Proj 4	Proj 5	Proj 6	Proj 7	Proj 8	Proj 9	Proj 10
1	1	0	0	0	3	0	0	0	0	0	0	4	1	0
2	1	0	0	0	7	0	0	0	0	0	5	0	0	0
3	1	0	0	0	0	1	0	0	0	0	5	7	8	0
4	1	0	0	0	0	0	0	4	0	0	10	5	6	9
5	1	0	0	0	0	0	0	0	0	0	8	4	9	0
6	1	0	0	0	0	0	0	0	0	0	8	9	10	0
7	1	0	0	0	4	0	0	0	0	0	10	2	7	3
8	0	1	0	0	0	9	5	0	0	0	0	7	0	0
9	0	1	0	0	0	9	0	0	10	0	0	0	0	0
10	0	1	0	0	0	0	0	0	0	0	0	0	0	0
11	0	1	0	0	0	0	0	0	0	0	0	0	0	0
12	0	1	0	0	0	0	0	0	0	0	0	0	0	0
13	0	1	0	0	10	0	3	6	0	0	9	2	0	0
14	0	0	1	0	10	0	1	0	3	4	0	0	0	0
15	0	0	1	0	4	0	3	0	0	0	0	0	0	0
16	0	0	1	0	10	0	0	0	0	0	0	0	0	0
17	0	0	0	1	0	0	0	0	0	0	0	10	0	0
18	0	0	0	1	0	0	0	0	10	0	0	0	0	0
19	0	0	0	1	0	0	0	0	0	0	0	0	0	9

LB (Lower Bound)

	Proj 1	Proj 2	Proj 3	Proj 4	Proj 5	Proj 6	Proj 7	Proj 8	Proj 9	Proj 10
ASD	2	2	0	3	0	0	3	2	3	3
EPD	2	3	3	2	2	0	2	2	1	1
ESD	1	0	1	0	1	2	0	0	0	0
ISTD	1	1	1	0	1	3	0	2	0	0

UB (Upper Bound)

	Proj 1	Proj 2	Proj 3	Proj 4	Proj 5	Proj 6	Proj 7	Proj 8	Proj 9	Proj 10
ASD	3	3	0	4	0	0	4	4	4	4
EPD	3	3	4	3	3	0	3	3	1	2
ESD	2	0	2	0	2	2	0	0	0	0
ISTD	1	2	2	0	1	4	0	2	0	2

OUTPUT ALLOCATION

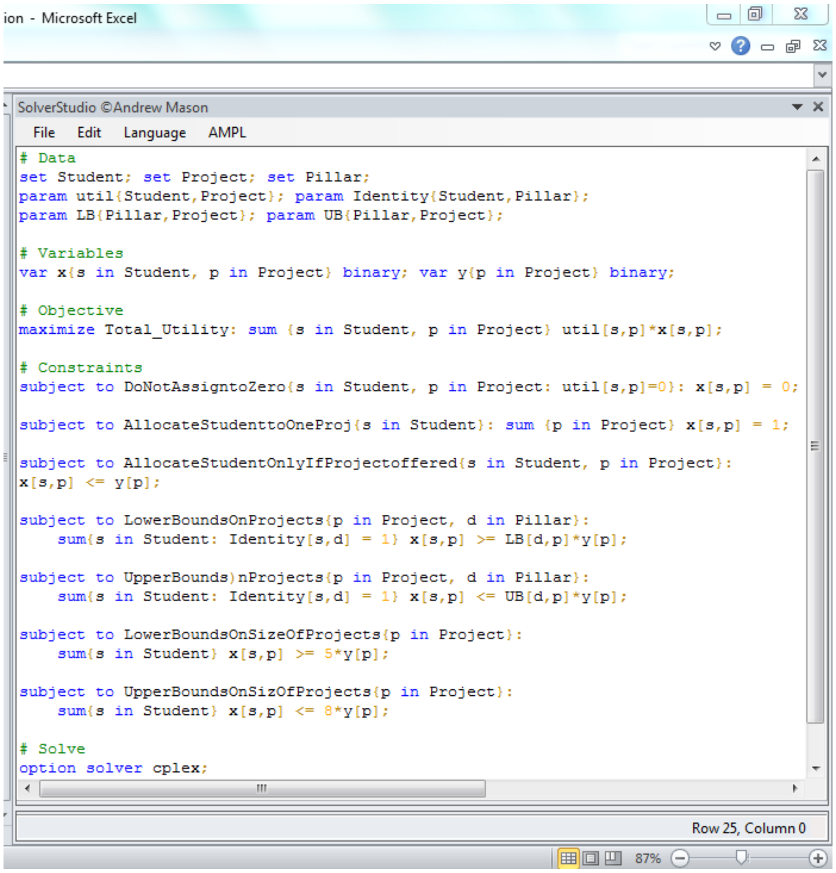
x (Project allocated to student)

Proj 1	Proj 2	Proj 3	Proj 4	Proj 5	Proj 6	Proj 7	Proj 8	Proj 9	Proj 10
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	1	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0
0	1	0	0	0	0	0	0	0	0
0	1	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	1	0	0	0
1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0
0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0

y (Project offered)

Proj 1	Proj 2	Proj 3	Proj 4	Proj 5	Proj 6	Proj 7	Proj 8	Proj 9	Proj 10
1	1	0	1	0	0	1	1	1	1

Figure 2: This spreadsheet illustrates the Excel interface for a snapshot of 19 students and 10 projects from the year 2016. Columns B to E identifies the pillars of the students and columns F to O capture the utility preferences of students for the projects. The rows for LB and UB provide the lower bound and upper bound on the number of students needed from each pillar for the projects. Columns BP to BY show the output allocation from the integer program where the x variables specify the project a student is allocated to and the y variables indicates if a project is launched. For example, in the optimal allocation, project 7 is launched with student 7 from ASD and student 13 from EPD are assigned to it. In addition, 3 more students from ASD and 1 student from EPD (not displayed in the sheet) were allocated to this project to meet the bound constraints. Project 3 on the other hand is not allocated in the final solution.



```
# Data
set Student; set Project; set Pillar;
param util{Student,Project}; param Identity{Student,Pillar};
param LB{Pillar,Project}; param UB{Pillar,Project};

# Variables
var x{s in Student, p in Project} binary; var y{p in Project} binary;

# Objective
maximize Total_Utility: sum {s in Student, p in Project} util[s,p]*x[s,p];

# Constraints
subject to DoNotAssigntoZero{s in Student, p in Project: util[s,p]=0}: x[s,p] = 0;

subject to AllocateStudenttoOneProj{s in Student}: sum {p in Project} x[s,p] = 1;

subject to AllocateStudentOnlyIfProjectoffered{s in Student, p in Project}:
x[s,p] <= y[p];

subject to LowerBoundsOnProjects{p in Project, d in Pillar}:
    sum{s in Student: Identity[s,d] = 1} x[s,p] >= LB[d,p]*y[p];

subject to UpperBounds)nProjects{p in Project, d in Pillar}:
    sum{s in Student: Identity[s,d] = 1} x[s,p] <= UB[d,p]*y[p];

subject to LowerBoundsOnSizeOfProjects{p in Project}:
    sum{s in Student} x[s,p] >= 5*y[p];

subject to UpperBoundsOnSizOfProjects{p in Project}:
    sum{s in Student} x[s,p] <= 8*y[p];

# Solve
option solver cplex;
```

Figure 3: This figure provides a snapshot of the Solver Studio Excel Interface. The optimization model was coded in AMPL and solved using CPLEX.

projects for a particular year 2015. Each student was allowed to indicate their top 10 preferred projects. In this year, there were zero students interested in one project while a maximum of eighty two students interested in another project. As the figure indicates, there is significant variation in interest among the projects with a few of the projects being extremely popular.

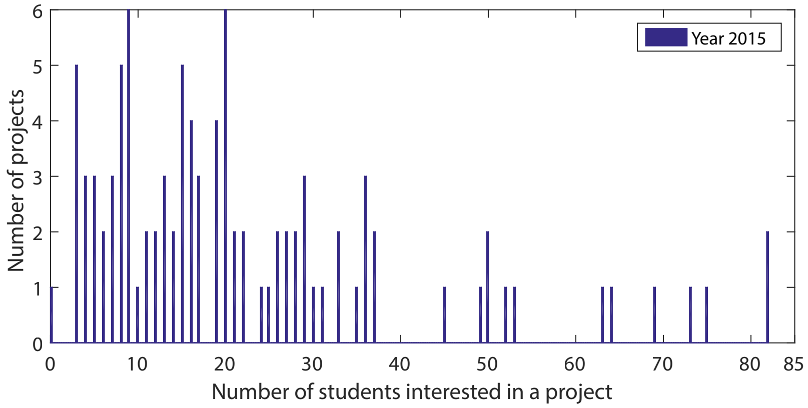


Figure 4: This figure illustrates the distribution of the number of students interested in a project for the year 2015. The horizontal axis shows the number of students who ranked a project in their top 10 list and the vertical axis shows the number of such projects.

Results

Table 3 summarizes the results of the project allocation using discrete optimization. The table shows the maximum total utility (average utility among those who were allocated projects in their top 10) and the distribution of the number of students assigned to their various preferred projects obtained from the optimization model for the years 2015 to 2017. The average utility of the allocated projects for students who obtained projects in the top 10 was 7.97, 8.47 and 9.16 for the three years. This corresponds to the students on average getting a project at a rank of around 1.5 with around 78% of the students assigned to projects in their top three choices. In the year 2015, one student could not be assigned to a project among his top 10 choices. There were no such students in the years 2016 and 2017. The running time to solve the integer programs to obtain these allocations was under a couple of minutes which is very reasonable for this application. At steady state the university is expected to have around 1000 students in the capstone projects each year and a reasonable goal in this case would be to develop a method that should be able to obtain allocations for problems of this size in a few minutes up to possibly to an hour. We believe that given the significant improvement in the solution times of integer programming solvers such as CPLEX and Gurobi, this approach is feasible even in the long run as the university grows. Table 4 specifies the distribution of the size of the projects that were launched across the different disciplines. As illustrated, the projects offered range from two to four disciplines thus capturing the original objective of having multidisciplinary capstone projects. Most of the three and four discipline projects were chosen in the final allocation (note that in 2016 only a total of 3 projects involved all four disciplines and all of these were offered in the optimal allocation while in 2017, 2 of the 3 projects that involved four disciplines were offered) which indicates that the students tend to like projects involving many disciplines and the integer program also helps allocate this without forcing it as a hard constraint. In Figure 5, we plot the number of students who expressed an interest in a project versus whether the project is offered in the final allocation. As should be expected from the figure, projects that have more student interest has an higher chance of being offered, validating the quality of the allocation.

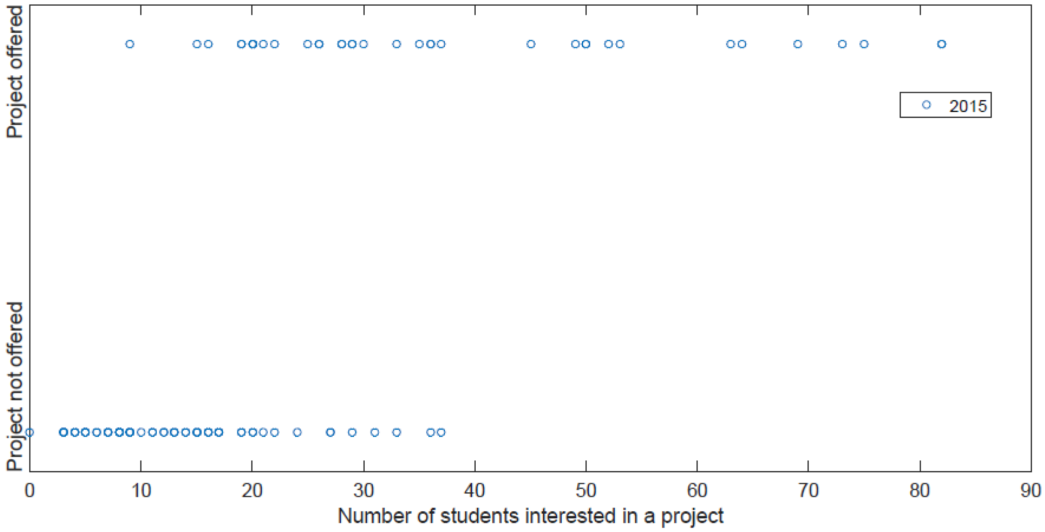


Figure 5: The plot indicates the projects launched by the algorithm and the number of students interested in the project. Each dot in the graph corresponds to project where the x-axis plots the number of students who ranked this project in their top 10 and the y-axis indicates if the project was launched in the final allocation. As seen from the figure, more of the popular projects are launched while fewer of the less popular projects are launched.

Table 3: This table provides the maximum total utility and distribution of ranks from the solution to the integer programs. Across the three years, there is only one student in 2015 who did not receive a project in the top ten. On average, students roughly obtained projects halfway between their top two choices, with around 78% of the students assigned to projects in their top three choices. Fewer students are assigned to projects at the bottom of their lists by incorporating fairness considerations in the model.

Year	2015	2016	2017
Total utility (average utility among top 10)	1739-M (7.97)	1440 (8.47)	2181 (9.16)
Utility = 10	65	67	128
Utility = 9	49	42	66
Utility = 8	29	22	22
Utility = 7	33	11	10
Utility = 6	16	10	5
Utility = 5	8	11	4
Utility = 4	5	4	2
Utility = 3	5	2	1
Utility = 2	6	1	0
Utility = 1	2	0	0
Utility = -M	1	0	0

Table 4: This table provides the distribution of the allocation of students to projects across disciplines.

Year	2015	2016	2017
Total number of projects launched	37	29	41
Number of projects allocated with two disciplines exactly	15	14	31
Number of projects allocated with three disciplines exactly	14	12	8
Number of projects allocated with four disciplines exactly	8	3	2

Comparison With Alternate Approaches

In this section, we also compare the results with two alternate allocation mechanisms. In Approach (a), we compute the maximum total utility using step 1 of the algorithm as before. However we replace step 2 in the algorithm to find a rank maximal matching rather than a fair matching. To find the rank maximal matching, the maximum number of students are assigned to their most preferred project and given this, the maximum number of students are assigned to their second most preferred project and so on. We use integer programming as before to find this matching. In Table 5, we compare the distribution of the students assigned to the different ranks with such an approach for the year 2015. As indicated in the table, in Approach (a), we can assign 70 students to their top ranked choice instead of 65 students in the year 2015. However this comes at the cost of potentially having more students assigned to lower ranked projects. For example, this leads to 8 students being assigned to projects to their second choice from bottom instead of 6 as in our proposed approach. In Approach (b), we find the allocation by dropping step 1 which finds a maximum utility matching, but rather directly find a fair matching solution (step 2). The results of such an approach is show in Table 5 where the matching is fair but it comes at the cost of loss of efficiency. For example, from a fairness perspective, 2 students are moved from being assigned to their bottom ranked project in Approach (b). However, the average utility of the projects allocated to students in their top 10 choices in Approach (b) decreases to 7.32 from 7.97. This reduction has been called as the price of fairness in related literature (see Bertsimas, Farias and Trichakis [4]). This can be partly visualized by a fairly significant drop in the number of students who are assigned their top project which drops from 65 in our proposed approach to 34 in Approach (b). This indicates that in our dataset, there appears to be sufficient scope to find a relatively fair matching among the maximum utility solutions since the price of fairness might be too high for such an application.

Discussion

In this section, we discuss some of the student feedback on the capstone project allotted to them and identify what worked well and what could be improved on for further iterations of the project allocation. Currently, the end of term survey of the capstone course at the university focuses on faculty and course evaluation but does not capture student feedback on their happiness/unhappiness with the final project allocated. Another aspect that we identified as important was to try and

Table 5: This table provides the distribution of ranks from the different approaches for 2015. The table compares the allocation from the three approaches - Our Proposed Approach which combines efficiency and fairness, Approach (a) finds a rank maximal matching from the efficient solutions and Approach (b) finds a fair matching without a focus on efficiency. From Approach (a), more students obtain lower ranked projects while from Approach (b), fewer students obtain their top ranked projects. Our Proposed Approach tackles both these issues.

Approach	Our Proposed Approach	Approach (a)	Approach (b)
Utility = 10	65	70	34
Utility = 9	49	47	48
Utility = 8	29	24	30
Utility = 7	33	35	41
Utility = 6	16	16	20
Utility = 5	8	6	18
Utility = 4	5	5	13
Utility = 3	5	5	8
Utility = 2	6	8	6
Utility = 1	2	2	0
Utility = -M	1	1	1

engage the students more in the design process of the allocation mechanism and to provide greater transparency to students on how projects were being allocated to them.

Towards this in 2017, a two hour class was conducted where the discrete optimization model that was being used to perform capstone allocation was introduced to the students. At the end of this session, a survey was run. While the session was conducted for a small random sample of twenty students, we obtained some useful inputs from this session as documented next. Two questions were posed to the 20 students - one quantitative and the other qualitative. The first question was as follows: "How happy or unhappy were you with the project that was allocated to you?" The response to this question was as follows: Very Unhappy (0 students), Unhappy (1 student), Neutral (4 students), Happy (4 students), Very Happy (11 students). The response to this question is encouraging since the survey was conducted at the end of the term and hence also reflects the student's actual experience with the capstone project.

The second question was as follows: "Provide any inputs you have on the capstone allocation mechanism." Some of the key inputs from students were as follows. Some students wanted the allocation mechanism to go beyond using just the degree specialization to using sub-specializations in allocating projects. The feasibility of doing so needs to be considered early on in the project scoping session and faces some challenges due to the open-ended nature of some of the design projects. We will be working with the capstone office to explore the feasibility of such an approach for future years. Another suggestion was to possibly consider alternate strategies for capturing project preferences by allowing students to bid on projects by providing a budget of virtual currency or providing students with an opportunity to express group preferences. This provides more strategic opportunities for students in their bidding process but complicates the data collection process. Another important extension to the model would be allocating faculty to the capstone projects. In the feedback, students also expressed happiness that the actual optimization approach being used in the capstone allocation was shared with them, since it provided transparency and made them realize that their capstone experience was not being determined by a mysterious black box. Reflecting on this, we found that this application provided a great opportunity to showcase the power of discrete optimization methods and more generally prescriptive analytics in using models to go from data to decisions. Our experience in interacting with administrators, faculty and students across disciplines on the capstone was that it helped showcase the importance of such tools to those who might not necessarily be familiar with them in solving important allocation challenges that often arise in universities.

Acknowledgements

The research of the first author was partly supported by the SUTD-MIT IDC grant number IDG21700101 Design of the Last Mile Transportation System: What Does the Customer Really Want?. The authors would like to thank Pey Kin Leong, the Associate Provost of Education at SUTD, the faculty leads for the capstone project which include Katja Holtta-Otto, Bige Tuncer, Lucienne Blessing and administrative staff which includes Shanty Coomaraswamy and Foo Yu-Chiann for their feedback, help and support in implementing the capstone project allocation mechanism at the university. The authors would also like to thank the Editor-in-Chief Michael Gorman, the Associate Editor and two anonymous reviewers for their insightful comments on improving the paper.

Appendix A: Network Representation

Define a bipartite graph $G(\mathcal{S} \cup \mathcal{P}, \mathcal{E})$, with $\mathcal{S} = \{s_1, \ldots, s_m\}$, the set of student nodes and $\mathcal{P} = \{p_1, \ldots, p_n\}$, the set of project nodes. We let $\mathcal{D} = \{d_1, \ldots, d_t\}$ denote the set of possible types (disciplines) of the students. Associated with each student $s \in \mathcal{S}$ is a particular student type $d(s) \in \mathcal{D}$. Associated with each project $p \in \mathcal{P}$ is a set of possible student types $\mathcal{D}(p) \subseteq \mathcal{D}$ that the project accepts. An edge (s, p) is present if student s can be allocated to project p, namely $d(s) \in \mathcal{D}(p)$. We let $\mathcal{E} \subseteq \mathcal{S} \times \mathcal{P}$ denote the set of undirected edges of the graph. The cardinality of each set $|\mathcal{D}(p)|$ is either two, three or four in our context since each project must have students from at least two disciplines and at most four disciplines.

Appendix B: Mathematical Formulation

We define the parameters as follows:

$$\begin{aligned} \text{UB}(p,d) &= \text{Upper bound on the number of students of type d needed for project p} \\ \text{LB}(p,d) &= \text{Lower bound on the number of students of type d needed for project p} \\ \text{util}(s,p) &= \text{Utility of project p for student s in the network} \end{aligned} \tag{1}$$

We define two sets of decision variables as follows:

$$x_{sp} = \begin{cases} 1 & \text{if student s is allocated to project p,} \\ 0 & \text{otherwise,} \end{cases} \tag{2}$$

$$y_p = \begin{cases} 1 & \text{if project p is offered,} \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

We now formulate a maximum utility discrete optimization formulation for the capstone allocation problem as follows:

$$
\begin{aligned}
\max \quad & \sum_{(s,p)\in\mathcal{E}} \text{util}(s,p)x_{sp} & & \\
\text{s.t.} \quad & \sum_{p\in\mathcal{P}:(s,p)\in\mathcal{E}} x_{sp} = 1, & \forall s \in \mathcal{S}, \quad & \text{(a)} \\
& x_{sp} \le y_p, & \forall (s,p) \in \mathcal{E} \subseteq \mathcal{S}\times\mathcal{P}, \quad & \text{(b)} \\
& \sum_{s\in\mathcal{S}:(s,p)\in\mathcal{E},d(s)=d} x_{sp} \ge \text{LB}(p,d)y_p, & \forall d \in \mathcal{D}(p), \forall p \in \mathcal{P}, \quad & \text{(c)} \\
& \sum_{s\in\mathcal{S}:(s,p)\in\mathcal{E},d(s)=d} x_{sp} \le \text{UB}(p,d)y_p, & \forall d \in \mathcal{D}(p), \forall p \in \mathcal{P}, \quad & \text{(d)} \\
& x_{sp} \in \{0,1\}, & \forall (s,p) \in \mathcal{E} \subseteq \mathcal{S}\times\mathcal{P}, \quad & \text{(e)} \\
& y_p \in \{0,1\}, & \forall p \in \mathcal{P}. \quad & \text{(f)}
\end{aligned}
\tag{4}
$$

Appendix C: Towards Fairness

(a) Step 1 (Maximum utility):

Solve the integer program in (4). Let the optimal objective value be denoted by z^*.

(b) Step 2 (Fairness):

Sub-iteration $k = 1$:

For the projects with utility of 1, solve the following integer program to minimize the number of students assigned to the projects that are the least preferred over the set of the maximum utility matchings:

$$
\begin{aligned}
\min \quad & \sum_{(s,p)\in\mathcal{E}:\text{util}(s,p)=1} x_{sp} \\
\text{s.t.} \quad & \text{(3a)-(3f)}, \\
& \sum_{(s,p)\in\mathcal{E}} \text{util}(s,p)x_{sp} = z^*.
\end{aligned}
\tag{5}
$$

Let the optimal objective value to (5) be denoted by z_1^*.

Sub-iteration $k = 2, \ldots, K$:

Solve the following integer program to minimize the number of students assigned to the projects with utility of k over the set of the maximum utility matchings with the smallest number of students assigned to less preferred projects:

$$\begin{aligned} \min \quad & \sum_{(s,p)\in\mathcal{E}:\text{util}(s,p)=k} x_{sp} \\ \text{s.t.} \quad & \text{(3a)-(3f)}, \\ & \sum_{(s,p)\in\mathcal{E}} \text{util}(s,p)x_{sp} = z^*, \\ & \sum_{(s,p)\in\mathcal{E}:\text{util}(s,p)=t} x_{sp} = z_t^*, \quad \forall t = 1,\dots,k-1. \end{aligned} \tag{6}$$

Let the optimal objective value to (7) be denoted by z_k^*. Go to the next sub-iteration till $k = K$.

(c) Step 3 (Randomization):

Choose independent random numbers $\epsilon_{s,p}$ say normally distributed with mean 0 and variance 1 for each student-project pair and solve the integer program:

$$\begin{aligned} \max \quad & \sum_{(s,p)\in\mathcal{E}} (\text{util}(s,p) + \epsilon_{s,p})\, x_{sp} \\ \text{s.t.} \quad & \text{(3a)-(3f)}, \\ & \sum_{(s,p)\in\mathcal{E}} \text{util}(s,p)x_{sp} = z^*, \\ & \sum_{(s,p)\in\mathcal{E}:\text{util}(s,p)=k} x_{sp} = z_k^*, \quad \forall k = 1,\dots,K. \end{aligned} \tag{7}$$

Appendix C: Additional Constraints

(a) Let $\mathcal{P}'$ be the set of projects offered by that particular company. Then,

$$\sum_{p\in\mathcal{P}'\subseteq\mathcal{P}} y_p \geq 1, \tag{8}$$

(b) Let LB and UB denote the lower and upper bound on number of students per project. Then,

$$\begin{aligned} \sum_{s\in\mathcal{S}:(s,p)\in\mathcal{E}} x_{sp} \geq \text{LB}y_p, \\ \sum_{s\in\mathcal{S}:(s,p)\in\mathcal{E}} x_{sp} \leq \text{UB}y_p. \end{aligned} \tag{9}$$

(c) Let B be the maximum difference in the number of students from two different disciplines allowed in a project. Then,

$$\left| \sum_{s\in\mathcal{S}:(s,p)\in\mathcal{E},d(s)=d_1} x_{sp} - \sum_{s\in\mathcal{S}:(s,p)\in\mathcal{E},d(s)=d_2} x_{sp} \right| \leq \text{B}y_p, \quad \forall d_1 \neq d_2 \in \mathcal{D}(p), \tag{10}$$

This can equivalently be reformulated as the set of linear constraints:

$$\begin{aligned}
\sum_{s\in\mathcal{S}:(s,p)\in\mathcal{E},d(s)=d_1} x_{sp} - \sum_{s\in\mathcal{S}:(s,p)\in\mathcal{E},d(s)=d_2} x_{sp} \leq \mathrm{B}y_p, \quad \forall d_1 \neq d_2 \in \mathcal{D}(p),\\
\sum_{s\in\mathcal{S}:(s,p)\in\mathcal{E},d(s)=d_2} x_{sp} - \sum_{s\in\mathcal{S}:(s,p)\in\mathcal{E},d(s)=d_1} x_{sp} \leq \mathrm{B}y_p, \quad \forall d_1 \neq d_2 \in \mathcal{D}(p).
\end{aligned} \tag{11}$$

References

[1] Abdulkadiroğlu, A., Sönmez, T. (1998). Random serial dictatorship and the core from random endowments in house allocation problems. *Econometrica*, 66(3), 689-701.

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

[3] Baker, K. R., Powell, S. G. (2002). Methods for assigning students to groups: a study of alternative objective functions. *Journal of the Operational Research Sociey*, 53, 397-404.

[4] Bertsimas, D., Farias, V. F., Trichakis, N. (2011). The price of fairness. *Operations Research*, 59(1), 17-31.

[5] Bertsimas, D., Farias, V. F., Trichakis, N. (2013). Fairness, efficiency, and flexibility in organ allocation for kidney transplantation. *Operations Research*, 61(1), 73-87.

[6] Budish, J. N., Cachon, G. P., Kessler, J. B., Othman, A. (2017). Course match: A large-scale implementation of approximate competitive equilibrium from equal incomes for combinatorial allocation. *Operations Research*, 65(2), 314-336.

[7] Bogomolnaia, A., Moulin, H. (2001). A new solution to the random assignment problem. *Journal of Economic Theory*, 100, 295-328.

[8] Budish, E., Che, Y-K., Kojima, F., Miglrom, P. (2013). Designing random allocation mechanisms: Theory and Applications. *American Economic Review*, 103(2), 585-623.

[9] Chen, C., Chester, C., Srinivasan, V., Wu, K., Thomo, K. (2016). Group-aware weighted bipartite b-matching. *Proceedings of 25th International Conference on Information and Knowledge Management*, 459468.

[10] Cutshall, R., Gavirneni, S., Schultz, K. (2007). Indiana Universitys Kelley School of Business uses integer programming to form equitable, cohesive student teams. *Interfaces*, 37(3), 265-276.

[11] Gale, D., Shapley, L. (1962). College admissions and the stability of marriage. *American Mathematical Monthly*, 69, 9-14.

[12] Gorman, M. F. (2010). The University of Dayton operations management capstone course: Undergraduate student field consulting applies theory to practice. *Interfaces*, 40(6), 432443.

[13] Hauhart, R. C., Grahe, J. E. (2014). Designing and teaching undergraduate capstone courses. Wiley and Sons Inc.

[14] Hindriks, J., Myles, G. D. (2006). Intermediate Public Economics. MIT Press.

[15] Hooker, J. N., Williams, H. P. (2012). Combining equity and utilitarianism in a mathematical programming model. *Management Science*, 58(9), 1682-1693.

[16] Hylland, A., Zeckhauser, R. (1979). The efficient allocation of individuals to positions. *Journal of Political Economy*, 87(2), 293-314.

[17] Kirkwood, C. W. (2004). Selecting student project teams when it really matters: An optimization-based approach using internet resources. *INFORMS Transactions on Education*, 4(3), 9-27.

[18] Lopes, L., Aronson, M., Carstensen, G., Smith, C. (2008). Optimization support for senior design project assignments. *Interfaces*, 38(6),448-464.

[19] Luss, H. (1999). On equitable resource allocation problems: A lexicographic minimax approach. *Operations Research*, 47(3), 361-378.

[20] Manlove, D. F. (2013). Algorithmics of matching under preferences. *Series on Theoretical Computer Science: Volume 2*, World Scientific.

[21] Mason, A. J. (2013). SolverStudio: A new tool for better optimisation and simulation modelling in Excel. *INFORMS Transactions on Education*, 14(1), 4552.

[22] Ogryczak, W. (2005). On the lexicographic minimax approach to location problems. *European Journal of Operational Research*, 100(3), 566585. 100(3), 566-585.

[23] Rowles, C. J., Koch, D. C., Hundley, S. P., Hamilton, S. J. Toward a model for capstone experiences: Mountaintops, magnets, and mandates. *Assessment Update*, 16(1), 115.