Improving Student Placement in IQP Centers via Preference Matching

A Major Qualifying Project submitted to the faculty of

WORCESTER POLYTECHNIC INSTITUTE

In partial fulfillment of the requirements for the degree of

Bachelor of Science

By

Camila Dias

Lin Jiang

Elizabeth Karpinski

ADVISOR:

Professor Andrew Trapp, Ph.D.

SPONSOR:

Interdisciplinary and Global Studies Division (IGSD)

Abstract

The goal of this MQP is to improve Worcester Polytechnic Institute's (WPI) Interdisciplinary and Global Studies Division (IGSD) off-campus IQP placement process. The IQP is a cornerstone of the WPI plan, however, placements have become increasingly complex and demanding as the number of applicants grow. Our team proposes an Excel-based decision support tool to improve the matching of students and IQP project center directors based on expressed preferences, thereby assisting IGSD by recommending student-IQP site placements. The decision-support tool uses VBA to build and solve an optimization model. In addition to reducing the time and effort required to place students, we believe that the decision-support tool will increase overall participant satisfaction with placements.

Acknowledgment

We would like to thank our sponsor IGSD for the support during the completion of this project. We especially would like to acknowledge Anne Ogilvie, Director of Global Operations, and Deborah Fusaro, Administrative Assistant, for their continuous encouragement and assistance, as well as their efforts in providing insight and information wherever necessary.

We would also like to thank both the students and the project center directors who participated in our surveys. The data from those questions allowed us to create a trial-run of our model, and demonstrate our results, which would not have been possible without their cooperation.

Additionally, we would like to offer our sincere gratitude to our advisor, Professor Andrew Trapp, for his ideas, guidance, and support.

Authorship

This report was equally written by Camila Dias, Lin Jiang and Elizabeth Karpinski with the exception of Sections 5. More Alternative Approaches, 6. More Alternative Approaches: Multiple Objectives, 7. More Alternative Approaches: Balancing, 8. Post-Optimality Analysis and 9. Sub-optimization written by Lin Jiang as part of her dual Industrial Engineering and Mathematical Sciences degree. Sections of this paper were drafted individually, however the final report was revised and edited in a combined effort.

Table of Contents

List of Figures

List of Tables

1. Introduction

Worcester Polytechnic Institute (WPI) is a private technological university with a project-based core curriculum (About WPI, 2016). As part of the undergraduate degree requirement, every WPI student participates in local or global projects that develop solutions to problems related to science and society. There are three important projects required for undergraduate students: Humanities & Arts (HUA), Interactive Qualifying Project (IQP), and Major Qualifying Project (MQP) (Global Projects Program, 2016). WPI's global projects are an important and well-known cornerstone of its project-based curriculum. Recently, WPI has been awarded the prestigious Bernard M. Gordon Prize from the United States National Academy of Engineering, which is the highest level of engineering honor there is in the United States, for Innovation in Engineering Technology (Bernard M Gordon Prize, 2016). The global projects have also been recognized by the Princeton Review as the most popular study abroad program in the nation (Most Popular Study Abroad Program, 2016).

Perhaps the most distinctive of projects at WPI is the IQP, which provides students an opportunity to apply their technical domain knowledge and skills to real-world projects. Unlike a traditional academic course, the IQP challenges students to engage with interdisciplinary peers and solve issues that matter to society. Through the IQP, student teams collaborate with faculty advisors and members of the community to make a social and economic impact locally, regionally, nationally, and often internationally (Interactive Qualifying Project, 2016). Thus, the IQP is a life-changing program and formative experience that develops future leaders by creating focus on

interdisciplinary approaches to problem solving.

To complete their IQP, undergraduate students have the opportunity to stay in Worcester, or go to an off-campus site, with approximately 31 project centers available around the globe. Those who choose an off-campus IQP go through a competitive selection process, followed by rigorous cultural preparation (Global Project Program, 2016). The Interdisciplinary and Global Studies Division (IGSD) administers this process, with the objective to help and guide students to complete their on or off campus projects (Interdisciplinary & Global Studies, 2016).

Due to the opportunities and popularity of WPI's Global IQP projects, the number of applicants has steadily increased year after year. In 2016, IGSD received almost 900 applications as compared to roughly half that number only a few years ago. Although the number and capacity of international project sites has increased, it has not kept up with increasing demand. The application process is still manually driven: even with 900 applications, stacks of student application papers are sent back and forth between the IGSD staff members and project center directors until decisions are made. This manual process is both time-consuming and labor intensive, and is prone to human error at many points. This makes it both ill-suited to keep up with an increasing demand, and an increasing strain on the IGSD's limited resources.

To address these limitations, we propose an Excel-based decision support tool to process, manage, and suggest placements for student applicants. Through Visual Basic for Applications (VBA) and integer linear programming, we have created a functioning spreadsheet-based model of the tool. The tool is designed to take as input data on the qualifications and preferences of students, and the criteria and capacities of sites; it formulates this information into an optimization

model, which then solves for a set of optimal student-site matches, which are outputted as recommended placements. To test it, we conducted a trial run of our tool as proof-of-concept with 104 students to 36 possible site and term locations. Our results show our tool can make WPI's IGSD IQP placement process faster and easier for students, project center directors, and IGSD staff alike. Furthermore, our tools' emphasis on improving the fit between student and project center helps to not only increase the number of students getting higher preference placements, but we believe lead to better quality matches between student and site as well.

2. Background

2.1 Student-Success-Centered Placement

IGSD is dedicated to helping students to understand and appreciate societal and technical issues by providing hands-on experience. Allowing students to gain exposure to different cultures and solve real-world problems helps them to improve soft skills such as teamwork and resiliency, as well as hard skills such as data analysis and programming. As a key purpose of WPI's global program is to enable WPI students to have an international experience, IGSD would like to allow as many students as possible the opportunity, without sacrificing quality of the program. Thus, the objective of any placement system used by IGSD must prioritize student-success-centered placement, that is, placements that reflect opportunities where students can be both successful and satisfied.

Quantifying what makes a student satisfied and what gives them to the potential to be successful is a difficult question. Proper assessment of the fit between students and project centers should take into account various levels of student potential, compatibility, experience and preference with and for different centers. This also has to be balanced with directors' own criteria to ensure students will be well-prepared to succeed at the project center. Therefore, IGSD's challenge is to design the student placement process to use the students' weighted matching scores based on quality and quantity of the matches, to deliver the most desirable student-project center placements.

2.2 Decision Support Systems

Decision support systems blend human judgment and intuition with the backing of computerized processing capabilities, combining the strengths of both. Krol (2013) defined decision support systems as "computer systems that provide users with support to analyze complex information and help make decisions." In some ways, decision support systems are an ideal antidote to human problems with repetitive, consistent decision making, as well as mistakes in intuitive quantitative analysis. Not only do they provide a consistent and customizable structure to analyze problems, but they can also be designed to have visual outputs, such as tables and graphs, to easily demonstrate the 'thought' process behind the model and recommended solutions.

For our purposes, a decision support system is a standardized, generally interactive method wherein a technological system is leveraged to provide computational analysis of data to support the human decision maker. This can include processing information for analytical tasks such as regression, simulation, and optimization, and producing useful output for human decision makers to interpret. It is largely intended to support the human decision maker, rather than as a replacement. Often decisions support systems allow users to express their own data and priorities into the calculations and analysis.

Decision support systems are also notable for their value in making technical processes accessible to people without a technical background. Many problems faced by industrial engineers require not just technical expertise to solve the problem, but the ability to structure the problem in an understandable manner for those without such skillsets. A decision support system, armed with the correct models and access to the appropriate data and domain knowledge, can fill this need.

This is often described as a difficult IT challenge, given that "it requires the availability of platforms which allowed [sic] the integration of various technologies (data, models, codes, etc.)" (Hanna, 2004). Hanna also recognizes Microsoft Excel, a user-friendly and popular platform among both among managers and engineers alike, as an ideal tool for decision support systems.
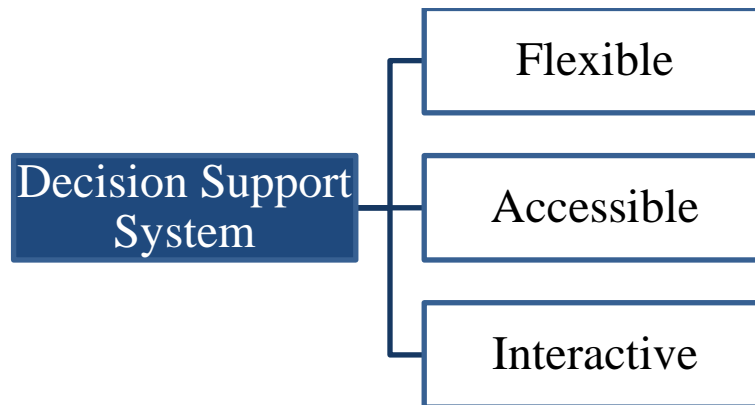
Figure 1 - Decision Support Systems

2.2.1 History of Decision Support Systems

Computerized decision support is a very recent field, but had been steadily gaining traction. It first became practical in the 1960's, the same decade that the first cases of academic literature and industry usage appear (Arnott, 2005). Researchers at the Carnegie Institute of Technology were instrumental in their recognition of different applications for the emerging technology; in one project, they even pioneered the use of decision support systems (DSS) in portfolio analysis and management (Krol, 2013). However, the lack of user-friendly software that could be operated at both the manager and the engineer level made it difficult for DSS to gain popularity. For decades, they remained at the fringes of business practice; yet, with the advent of ever increasing processing power, improved software, and generations more familiar and comfortable with technology, by

2005, they were considered "a mainstream commercial IT movement that all organizations engage" (Arnott 2005).

2.2.2 Human Decision Making

One often unconsidered aspect of human decision-making is called decision fatigue. Like willpower, research suggests that the ability to make informed decisions is depletable. The model of willpower as a muscle or exhaustible resource is well-established; previous studies have often showed a loss of 'willpower' after having to make decisions (Villarica, 2012). One worrying effect of this mental depletion is a reluctance to make trade-off decisions. This can push people into the so-called path of least resistance, and least commitment, often doing nothing at all (Tierny, 2011). This is especially concerning when considering its impact in a series of high trade-off decisions, such as rankings, or selecting students from a group, both of which the current IQP placement process relies on.

In many cases, even apparently benign background variables can hijack decision-making. In one study, researchers found that by manipulating the order of the options in which different car features were presented, they could increase the average customer's bill by $2,000 per car. By offering more expensive choices as the default later within the process, customers were swayed into taking them, even when it was not in their best interests (Tierny, 2011). These same insidious principles are at play in other areas as well. Another study of 1,100 judge rulings found that the timing of the request had a significant impact on the probability of a prisoner getting parole: the odds were significantly better for prisoners being considered early in the morning, or immediately after lunch (Tierny, 2011).

In many cases, the best antidote to human decision-making is to carefully structure and control it to remove the potential for random variation or hijacking. Difficult decisions are cognitively exhausting, and they add up over time to the detriment of the process of a whole. The unintentional influence of components such as time or order of applications in deciding placements is another concern. Thus, the structure and assistance of a decisions support system can be vital in maintaining consistency and quality in decision-making.

Human Decision Making	Technical Systems
Difficulty making consistent decisions using numbers	Less capable of learning, adapting, and incorporating new information
More capable of learning, adapting, and incorporating new information	Able to handle complex numerical operations quickly, repetitively and accurately
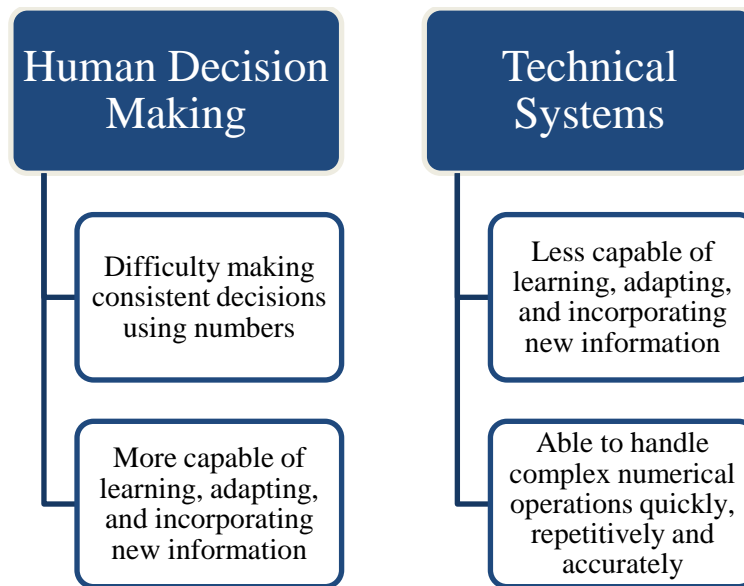
Figure 2 - Human Decision Making vs. Technical Systems

Another key concern in human decision-making is the human propensity to fall prey to inexact heuristics and biases. These are especially problematic where numbers are concerned, as people tend to fail at assessing probabilities objectively and pay little attention to prior probabilities and sample size (Tversky & Kahneman, 1974). A structured decision support system can remove the potential for subjectivity; once the human collaborators have agreed on the inputs, the analysis is outside of their control. Furthermore, with the creation of standardized inputs and outputs, the

risk of any unintentional quantitative error can be mitigated. With the exception of errors in data entry or interpretation, there is little room for the human in the process to make a mistake since they are not singularly responsible for deciding or performing the necessary calculations.

2.3 Use of Visual Basic for Applications (VBA)

Visual Basic for Applications (VBA) is a programming language used by Excel and other Microsoft Office programs. VBA allows users to record, run, and edit macros, which perform certain actions that need to be repeated within the programs. Recorded macros are saved as VBA macro files that can be applied to work in the future. One of the biggest advantages that VBA has is its ability to automate repetitive operations, which improves job efficiency. Another appealing feature of VBA is that it can be customized. For example, when people are creating the end-of-year profit report, it may be important to convince their manager that modifying the budget for some specific department will enhance the company's profit. To promote their ideas, they can customize the report for their manager by creating a custom budget button or an input box. The popular 'dashboard' concept is another iteration of this same idea. For easy interpretation of the data and better demonstration of the idea, a good user interface is essential for developing a user-friendly decision support system. Navigational buttons, function buttons, control buttons and user forms allow users to easily go back and forth in the spreadsheets, modify and manipulate the data (Hanna, 2004). This functionality allows VBA to be used to create and customize extensive applications, with the help of automated graph creation and form options (Sempf, 2016). Thus, the primary benefits of VBA are not only to enable people to effectively automate repetitive tasks, but also to modularize them for easier use and better distribution, especially for users who may lack a

technical background.

2.4 Optimization

Optimization is used to make the design or operation of systems as good as possible subject to some constraints that are imposed. There are many approaches to optimize systems depending on the types of systems as well as the system restrictions. Optimization is applied in virtually all areas including industrial design, scheduling system, resource allocation, staff planning and system controls (Pierre, 2014). However, the goal of all optimization procedures is always to obtain the best results possible satisfying the defined constraints.

According to Pierre, "while a system may be optimized by treating the system itself, by adjusting various parameters of the process in an effort to obtain better results, it generally is more economical to develop a model of the process and to analyze performance changes that result from adjustments in the model" (2014). Mathematical models are commonly used to formulate the optimization process. The framework for optimization includes defining a model structure with decision variables, constraints and an objective function to express "what's best".

2.5 Matching Problems

A common example of matching problems, often found in schools and universities, is students to courses. A paper concerning course match discussed a new mechanism to achieve a more efficient and fair student-course allocation (Budish, Cachon, Kessler and Othman, 2016). They introduced the A-CEEI mechanism to optimize fake-money income by assigning students to the available courses subject to the maximum capacity. This method requires input information

such as student budgets, courses' targets and maximum capacity. They also emphasize the three stages of computational methods, which were price vector search, eliminate oversubscription and reduce under subscription. This course matching mechanism can largely improve the effectiveness of course scheduling, and satisfaction of students.

2.6 Current IQP Matching Process

Currently, every WPI undergraduate student who wishes to apply to any off campus IQP is required to go through an application process. First, students have to attend an information session of their top IQP center choice. Students are also encouraged to go to sessions of other sites of interest. Then, students complete an online application through the Global Portal website, and rank sites from 1 through r, with r being the choice number r of the student. Applications include students' general information, resume, personal statement and transcript. Students who are either on probation or are otherwise not qualified for the projects are removed from process. After the deadline has passed, applicants are invited for an interview (only some sites require interviews) with their top choice project center. Once applications are reviewed and interviews are completed, applicants are either accepted, or not, to their first choice center. Students are then considered by the directors at their next ranked available center. If they are not accepted at their next-ranked center, their application proceeds to the next one, and so on, until all of their ranked sites have been considered. Students who are not accepted anywhere, including those who only ranked one site or are not strong candidates for the global projects program, are either moved to a waitlist, or removed altogether from the process. Once all decisions have been made and students have chosen whether to accept their placement, those in the waitlist can rank sites again according to new

availability.

There are two main problems of concern with the current process. First, the ranking system may not always reflect how students actually feel about the possibility of going to other top project centers. For example, a student may be equally interested in two popular and competitive project centers, A and B; under the current system, the student would be forced to rank them as first and second choice. If the student ranks project center A as his first choice, but is actually better suited for project center B, that student may not be accepted into either program: both popular project centers will fill up simultaneously in the first round, and the student will never have a chance to be considered for project center B, despite having a competitive candidacy. Second, as previously mentioned, much of the process is done manually. Applications are routed by hand or by inter-campus mail to project center directors and then back to the IGSD until all sites are filled. This is extraordinarily time consuming and labor-intensive for both the directors and IGSD. The students also suffer prolonged uncertainty as they wait for placements to be decided.
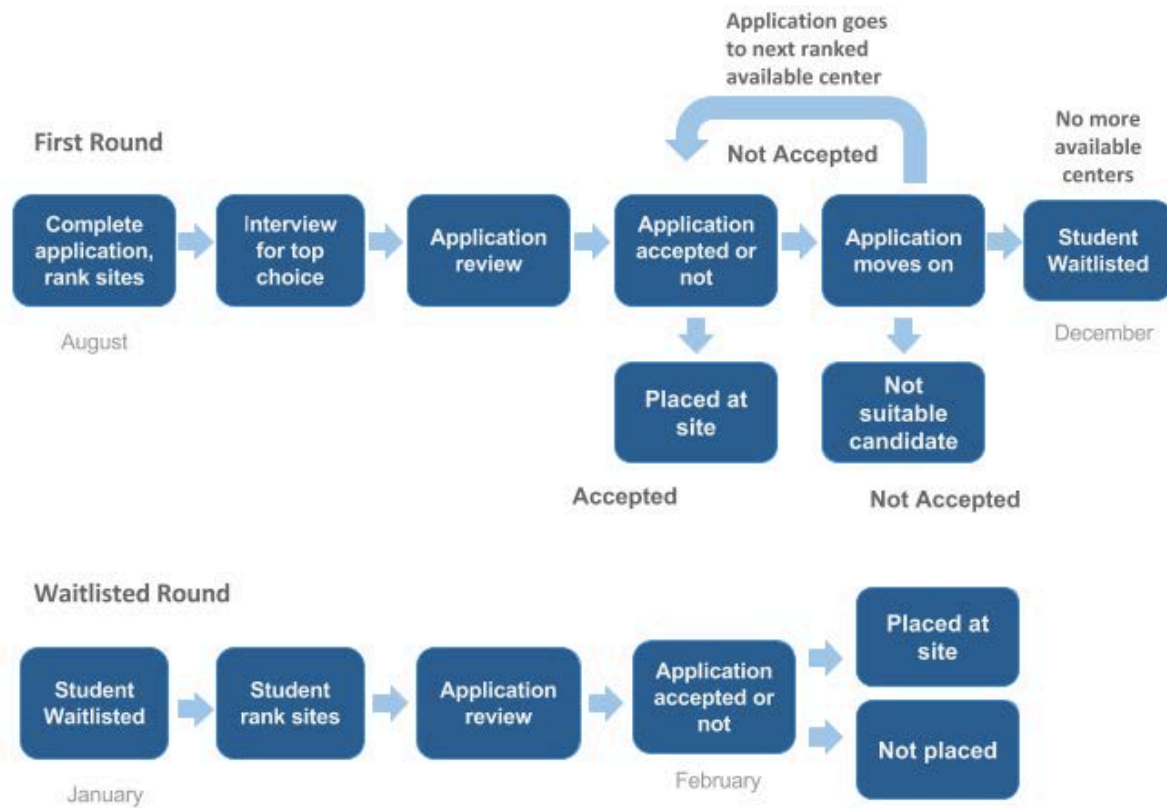
Figure 3 - IGSD Current Acceptance Process

2.7 Proposed IQP Matching Process

To improve the IQP match process we propose a decision support system that will help IGSD place students at project centers. The application process will mostly remain the same; however, the ranking of preference will be replaced with a rating system. Instead of ranking sites from 1 up to k (each student ranking a different number of sites), students will be able to rate centers between three options: "most want to go", "would like to go", and "do not want to go". These three options represent primary choices, secondary choices, and sites for which they have no preference. With the ranking system, students are able to mark as many sites as they would like as a primary or secondary choice, increasing their options and consequently, their chances to get

21

into competitive sites.

The selection process will also undergo some major modifications. Instead of a manual, paper-driven process, we are proposing an Excel-based optimization tool that will simultaneously match students with project centers. The decision support system will take into account information from both the project center directors and from the students to determine fit based on information gathered via surveys.

Figure 4 - Proposed IQP Acceptance Process

3. Methodology

3.1 Director Survey

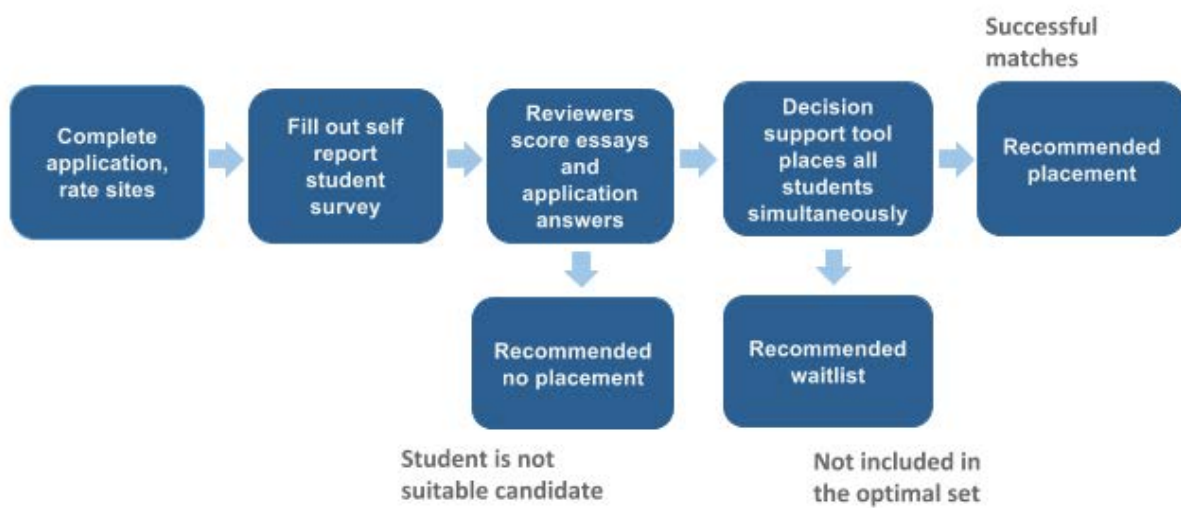We had two iterations of the director survey. In our initial survey, we asked directors approximately 40 broad questions, including an inquiry on the director's willingness to meet and provide more detailed information on their student IQP selection process. The goal of this first survey was to determine the aspects of student applications that are of interest to project center directors, and what criteria directors use to place students. From 13 responses, 7 directors agreed to meet with us. In the face-to-face meeting, we asked for feedback on the survey questions as well as additional information on their student evaluation. After interviews, we developed a second version of the survey. In the second survey we asked 18 questions in which directors had the chance to say how each evaluation criteria was important to their student selection process. The second time around we were able to get responses from 26 out of 30 directors. Once we had collected all the data from directors and developed a thorough understanding of their evaluative process, we developed the student survey. Director and student's survey are in Appendix A: Student's Survey and Appendix D: Consideration Matrix Calculation.

3.2 Criteria Development

In the course of this project, we developed different iterations of mathematical calculations to quantify student performance and director preferences. Our first model had 16 criteria and 59 sub-criteria on student attribute evaluation, which required an excessive amount of internal calculations and external resources. After interviewing project center directors and consulting with

our advisor Professor Andrew Trapp, we streamlined the model and narrowed it down to focus on

7 categories and 18 criteria, making the model more practical and easier to use while still including

the most meaningful data.

There are seven categories, which are used to classify students as shown in *Table 1*.

Table 1 - Model's Categorical Criteria

Category Name	Category Intention
Academics	Measures student academic performance
Teamwork/Cooperation	Measures how well students work in groups
Communication/Writing	Measures student ability to successfully communicate
Initiative/Motivation	Measures student ability and inclination to show consistent motivation and initiative
Openness/Resilience	Measures student openness to new challenges and experiences
Time Management/Responsibility	Measures how well student manages their responsibilities
Site-Specific Interest/Requirement	A last catch-all category under which site-specific requirements such as language skills fall

Within each category, we created a number of individual criteria which each count in a

weighted average toward one or more categories. Thus each criterion belongs to at least one

category, though most belong to more. For example, overall GPA, and consistency in passing

classes scores count towards the academics category. The criteria used in our trial run are: overall

GPA, consistency in passing classes, humanities grade average, major grade average, number of

challenging classes taken, grade improvement over time, extracurricular/volunteer, leadership positions, quality of essay writing, content of personal essay, PLA/TA/tutoring work, length of work experience, quality of work experience, international exposure/experience, Spanish, French, Mandarin and Cantonese.

To account for both criteria and sub-criteria in our model, we first developed a binary "consideration" matrix to indicate whether a sub-criteria is assigned to a "parent" category as shown in *Table 2*. For example, GPA Overall counts towards Academics, but not Teamwork/Cooperation. In the course of our experiments in the model, we tried several different category-criteria arrangements, including no category use at all. We believe that the increased complexity of including the categories and criteria relationships may allow for useful nuance in further development. To this end we have included functionality for the end user to customize not only the categories and criteria, but also the relationships between them, in our end product.

Table 2 – Partial View of Consideration Matrix

	Academic Excellence	Teamwork / Cooperation	Communication / Writing
Overall GPA	1	0	0
Consistency in Passing Classes	1	0	0
Humanities Grade Average	1	0	1
Major Grade Average	1	0	0
Number of Challenging Classes Taken	1	0	0
Grade Improvement Over Time	1	0	0
Extracurricular/Volunteer	0	1	0
Leadership Positions	0	1	0
Quality of Essay Writing	0	0	1

Figure 5 - User View of Consideration Matrix

The category ratings and the criteria ratings are combined according to the consideration matrix (*Table 2*), to result in a final set of normalized weights, which takes all three vectors into account. Example results are illustrated in *Table 3*, the full process can be found in Appendix D: Consideration Matrix Calculation.

Table 3 – Partial View of Final Preference Weights for Three Project Sites

Project Centers	Asuncion, Paraguay D18	Bangkok, Thailand C18	Bar Harbor, Maine E17
Consistency in Passing Classes	9%	11%	12%
Humanities Grade Average	7%	8%	7%
Major Grade Average	7%	7%	7%
Number of Challenging Classes Taken	7%	8%	7%
Grade Improvement Over Time	9%	8%	7%
Extracurricular/Volunteer	8%	7%	7%
Leadership Positions	6%	6%	10%
Quality of Essay Writing	6%	8%	7%

3.3 Student-Project Center Matching Computation

The first step of the computational process is quantifying the data. By using multiple-choice questions, we pre-emptively controlled the variability in the director survey data. This made it easy to convert each answer option ("Entirely Unimportant", "Somewhat Unimportant", "Neither Important nor Unimportant", "Somewhat important" or "Very Important") to a score from 1 to 5, respectively. The full director survey is available in Appendix B: Director's Survey. Once all the answers are converted to numbers, directors' scores are normalized as shown in *Table 5*.

Table 4 - Three Project Centers' Quantified Answers

Project Centers	Asuncion, Paraguay D18	Bangkok, Thailand C18	Bar Harbor, Maine E17
Academic Excellence	4	4.5	4
Teamwork / Cooperation	5	5	5
Communication / Writing	4	4	4
Initiative / Motivation	5	5	5
Openness to New Experience	5	5	5
Time Management/Responsibility	4	4	4
Site-specific Language Requirement	5	1	4

Table 5 - Three Project Centers' Normalized Data

Project Centers	Asuncion, Paraguay D18	Bangkok, Thailand C18	Bar Harbor, Maine E17
Academic Excellence	11%	14%	11%
Teamwork / Cooperation	14%	15%	14%
Communication / Writing	11%	12%	11%
Initiative / Motivation	14%	15%	14%
Openness to New Experience	14%	15%	14%
Time Management/Responsibility	11%	12%	11%
Site-specific Language Requirement	14%	3%	11%

Similarly to the director survey, multiple-choice answers from the student survey are converted to scores from 1 to 5, with 5 being full points for a criterion. To give the students the chance to elaborate some answers, we also developed a few open-ended questions. Two reviewers manually scored those written answers and students' essays, and then the average score from both reviewers was entered into the model.

To calculate the percentage of the student-project center match, the VBA code multiplies the normalized director preferences to the student scores for each criterion, creating essentially a weighted average based on director preferences. Since student performances are originally rated out of 5, to reduce it back into an easily understood percentage score, each match is divided by 5. An illustration of the results is shown in *Table 6*.

Table 6 – Sample of Student-Project Center Match Score

Criteria	Project Center	Students		
	Asuncion, Paraguay D18	Znggurj Ovbaqv	Qnivq Obivpu	Wbfrcuvar Objra
Overall GPA	9%	46%	37%	37%
Passing Classes	7%	37%	37%	37%
Humanities Grades	7%	37%	37%	22%
Major Average	7%	37%	37%	29%
Number of Challenging Classes	9%	18%	9%	9%
Grade Improvement	8%	25%	25%	33%
Extracurricular or Volunteering	6%	19%	25%	28%
Leadership Positions	6%	19%	22%	25%
Quality of Essay Writing	6%	19%	19%	19%
Content of Personal Essay	6%	17%	17%	19%
PLA/TA/Tutoring Work	7%	7%	7%	7%
Length of Work Experience	7%	22%	37%	37%
Quality of Work Experience	6%	19%	22%	25%
International Exposure	9%	37%	28%	9%
Total		359%	357%	336%
Average Score (divided by 5)		72%	71%	67%

Student project center preferences are also an essential consideration for the student placement tool, as students should not be placed at project sites where they have no intention of going to, even if they are a desirable match. Therefore, in the student survey, applicants indicate only their primary and secondary preferences, with all other sites as no preference (i.e., student placement there is undesirable). Their responses are converted to scores of 1, 0.5, and 0 respectively as shown in *Table 7*.

Table 7 - Students' Project Center Preference Matrix

Student Preferences	Znggurj Ovbaqv	Qnivq Obivpu	Wbfrcuvar Objen
Cape Town, South Africa B17	1	1	1
Copenhagen, Denmark A17	0.5	0.5	0.5
Copenhagen, Denmark D18	0.5	1	0
London, England D18	0	1	0
London, England E17	0	0.5	0
Melbourne, Australia B17	0.5	1	0.5
Melbourne, Australia D18	0.5	1	0

Our final match scores balanced the weighted-average of the director-student match with the student preferences, by multiplying the director-student match by the quantified student preference. Thus, no matter the director's preferences for the student, the final match score is a zero at any site the student does not want to go.

Table 8 - Partial Matrix of Final Matching Scores

Objective Function Coefficients	Znggurj Ovbaqv	Qnivq Obivpu	Wbfrcuvar Objen
Cape Town, South Africa B17	71%	72%	72%
Copenhagen, Denmark A17	34%	35%	36%
Copenhagen, Denmark D18	34%	70%	0%
London, England D18	0%	74%	0%
London, England E17	0%	37%	0%
Melbourne, Australia B17	36%	75%	37%
Melbourne, Australia D18	36%	75%	0%

3.4 Optimization Model

In our optimization model, our objective function value is obtained as the sum of the product of each matching score to each final placement (*Table 8*). Each placement is represented by our decision variables in the model, which are binary: 1 representing placement, and 0 representing no placement, for each student, for each site. The model has two constraints sets: each student can be placed k times, with $k = 1$ being the default (but the model can also be run to generate multiple site, $k > 1$ recommendations for each student), and the number of students assigned to each site must be equal to or less than the capacity. Other important considerations such as students' preferences and their qualifications, as well as project directors' preferences, are incorporated into the objective function, so as to not immediately disqualify a student from a placement. The problem is to assign a set of students S to a set of project centers P such that the overall weighted matching score is maximized.

3.4.1 Set definition

S: Set of students eligible to be placed in the global projects program, indexed by s

P: Set of project centers that have capacity to place students, indexed by p

3.4.2 Parameter definition

W_{sp} is the weight assigned to matching s ∈ S with project center p ∈ P

C_p is the capacity (in terms of number of students) for project center p ∈ P

3.4.3 Assumptions

    a. Assume all the student information we collected are correct

b. Assume our sample of students' information can reflect the behavior of general student population

3.4.4 Mathematical Programming Formulation

Variable Definitions:

Binary variables

$$x_{sp} = \begin{cases} 1 \text{ if student s is assigned to project center p} \\ 0 \text{ otherwise} \end{cases}$$

Objective Function Terms:

The objective is to maximize the total weighted score from matching students to project centers:

$$\text{Maximize} \sum_{s=1}^{|S|} \sum_{p=1}^{|p|} W_{sp} * x_{sp} \tag{1}$$

Constraints:

1. Each student is assigned to at most k, a user-specified number of centers:

$$\sum_{p=1}^{|p|} x_{sp} \leq k \, \forall \, s \in S \tag{2}$$

2. Each project center must remain within capacity:

$$\sum_{s=1}^{|S|} x_{sp} \leq C_p \forall \, p \in P \tag{3}$$

3. Placement variables must be binary:

$$x_{sp} \in \{0,1\} \, \forall \, s \in S , \, \forall \, p \in P \tag{4}$$

Objective function (1) is to maximize the sum-product of the matching weight and the student placement variables, meaning to maximize the total matching scores for all the students and sites. Constraint (2) indicates that every student can be assigned to as most one project center. Constraint (3) indicates that every project center must remain within capacity. The variable

definition (4) ensures that student placement variables are binary since they either get placed or they don't.

The result from solving the optimization model that maximizes the total weighted matching score over all placed students is presented in *Table 9*. By looking at *Table 9*, decisions can be made regarding which student goes to which project site. For example, student Ehfuqv Nohnyunvwn was placed at Rabat, Morocco, and student Avpubynf Oretfgebz was placed in Melbourne, Australia.

Table 9 - Partial Selection of Decision Matrix

Sites	Znggurj Ovbaqv	Qnivq Obivpu	Wbfrcuvar Objra	Students Placed	Capacity
Cape Town, South Africa B17	0	0	1	1	1
Copenhagen, Denmark A17	0	0	0	1	5
Copenhagen, Denmark D18	0	0	0	2	5
London, England D18	0	1	0	2	3
London, England E17	0	0	0	2	3
Melbourne, Australia B17	1	0	0	5	5
Melbourne, Australia D18	0	0	0	0	0

3.5 Procedure for Proof of Concept Trial Run

To conduct a trial run of our process and evaluate its effectiveness, we first had to collect sufficient information to mimic the real process from both students and project center directors. Student transcript and multiple choice survey answers provided objective and easily quantifiable information such as GPA. Subjective information about qualitative criteria such as work experience and on campus involvement were described by students in open-ended questions, or included in their IGSD application. This information was then scored by a pair of IGSD appointed

33

reviewers. To protect student privacy, IGSD staff anonymized student names using a cypher before passing the data off to our team. IGSD also incentivized student responses through Amazon gift cards to increase participation. In total from the survey, we received almost 120 responses, 104 of which we used in our trial run. Students who rated project centers we had no data for were dropped from the dataset for internal consistency.

3.6 Student – Director IQP Matching Tool

Our decision support tool recommends placements based on the information from directors and students. It uses optimization to find a set of "quality fits" between applicants and project centers. The tool is straightforward to use and it runs off of a spreadsheet, using OpenSolver and Visual Basic for Applications (VBA) code. To use the tool, IGSD will first decide which parameters to use in the model (categories and criteria), and then transfer directors' and students' survey answers to specifics worksheets. The data must be standardized and quantified ahead of time. Once all data is placed in the tool, the end user only needs to hit the single "run model" button, and analysis will be conducted and recommendations generated. A preview of the tool is shown on *Figure 6* and *Figure 7.* See Appendix C: Matching Tool Instructions for instructions detailed.

Figure 6 - Student-Director IQP Matching Tool Welcome Sheet

Figure 7 - Student-Director IQP Matching Tool Parameters

4. Results and Analysis

Once all necessary information had been collected, the team input the data into the optimization model, and solved for the optimal placements. To evaluate multiple variations of the model, the problem was set up and then re-solved under several different conditions. Each set of student-site placements were then evaluated by two criteria. First, by how many pairs matched the actual results in 2016; in these criteria, our team felt that the closer the model's results were to the human-created results, the better, as it would demonstrate an easier but accurate reproduction of the same outcome. Second, the matching preferences for both students and directors' placements are compared with IGSD's placements. This is to help the team gauge if the places in which the model differs from human results also correspond to a drop in or increase of satisfaction on its subjects. To thoroughly evaluate how our model performed under different parameters, we tested the results of the trial-run using two different measures: *accuracy with respect to IGSD's placements*, and *placements to preferred sites*. Director preferences being relatively homogenous, we placed priority on student satisfaction.

4.1 Actual IGSD Outcomes in Sample

Using the rankings of each site that our sampled students used in their IGSD application, we constructed *Table 10* to indicate student outcomes. In the majority of cases, students are placed in their first choice. However, students are more likely to be waitlisted and eventually accept a placement that they did not originally rank than to get their second choice.

Table 10 - Student Outcomes based on Actual IGSD Placement

Preference of Site	Number (Percentage) of Students from Sample Placed
1st Choice	79 (76%)
2nd Choice	5 (5%)
3rd Choice	4 (4%)
4th Choice	2 (2%)
5th Choice	3 (3%)
6th Choice	2 (2%)
7th Choice	3 (3%)
Placement Not Ranked	6 (6%)

4.2 Ranking Model Comparison to IGSD Placements

In order to best compare our model's results with IGSD's, we first used the actual IGSD application rankings as student preferences. To scale the rankings as student preferences, we set each preference value to $\frac{1}{r}$, where **r** represents the **r**th ranking. For example, if a student had ranked 5 project sites 1 through 5, the corresponding preference values would be 1, 0.5, 0.33, 0.25 and 0.2 respectively. Their placement results with the calculated student preference coefficients in the model are detailed in *Table 11.*

Table 11 - Ranking Formulation and Results

Ranking	Coefficient	Model Placements	IGSD Placements
1st Choice	1	79	79
2nd Choice	0.5	8	5
3rd Choice	0.33	7	4
4th Choice	0.25	2	2
5th Choice	0.2	1	3
6th Choice	0.17	1	2
7th Choice	0.14	1	3
8th choice	0.13	2	0
Placement Not Ranked	0	3	6

As can be seen in *Figure 8*, our ranking model placed more students at higher ranked sites. The same numbers of students receive their first choice, and more receive their second and third choices as well. In addition, fewer students are waitlisted and asked to reconsider the remaining options, which they previously did not rank.
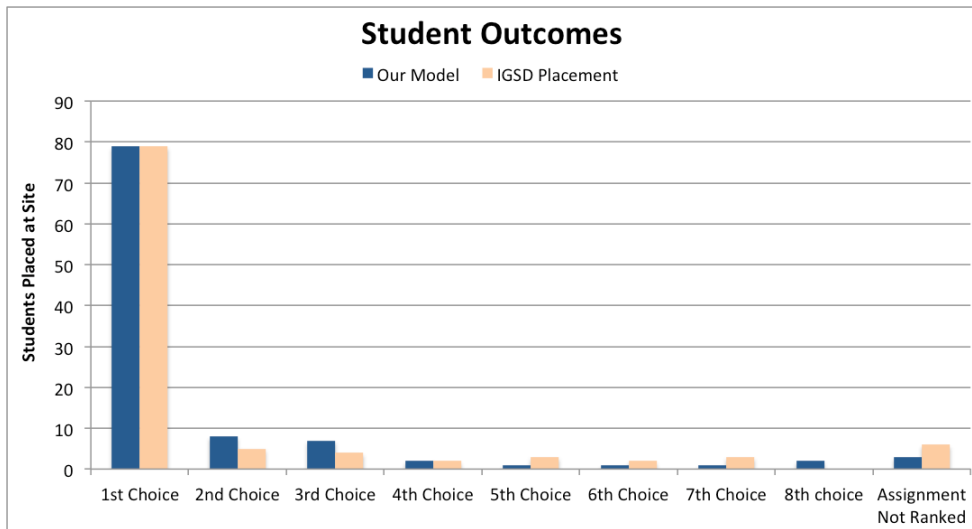
Figure 8 - Ranking Model Outcomes Compared with IGSD Placement

4.3 Rating Model Comparison to IGSD Placements

To test how rating might impact the optimization model in allowing greater flexibility of placements, we surveyed students and asked them to rate each site as "most want to go", "would like to go" and "would not go". In our model, these became student preference coefficients of 1, 0.5, and 0, respectively. To evaluate how IGSD placements corresponded to the rating system, we cross-referenced the actual IGSD placements with what students rated on the survey (see *Table 1*).

Table 12 - Rating Formulation and Results

Rating	Coefficient	Model Placements	IGSD Placements
Primary Choice	1	96	78
Secondary Choice	0.5	7	17
Placement Not Ranked	0	1	9

Using the ratings model, 93% of students were placed into primary choice sites, and only 7% were placed in secondary sites. Only one student was waitlisted in our rating model, compared to nine who were in the IGSD process. Theoretically, if the model is scaled up to include full site capacities, the increased number of possibilities and flexibility will make it possible to further improve the 93% placement in primary sites.
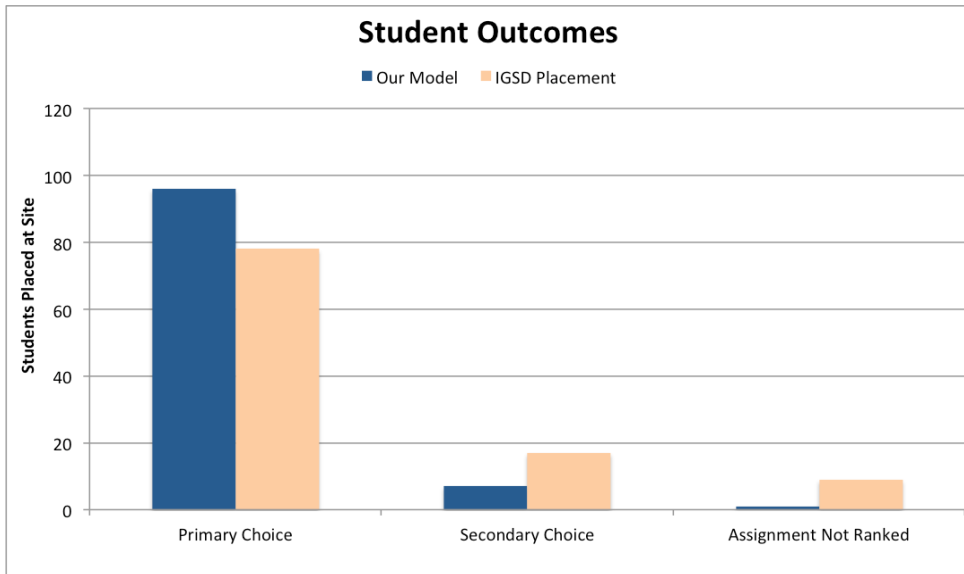
Figure 9 - Rating Model Outcomes Compared with IGSD Placement

4.4 Rankings vs. Ratings

One curious factor we found in our analysis is that student ratings in our MQP group's survey did not always reflect how the students had ranked project sites in their IGSD application. In fact, only 90% of students marked their top ranked site as a primary preference on the survey while almost 20% of students wound up disowning a 5th, 6th, or 7th ranked choice as no preference at all. This leads us to believe that students' stated preferences might be significantly different than their revealed preferences, which makes the interpretation of their satisfaction data difficult. Therefore, we conservatively look to group summaries to demonstrate the utility of the optimization model to improve matching outcomes. The general trends visible in *Figure 10* can clearly show that the rankings and student ratings are not entirely discrete; almost half of students gave their 2nd ranked choice a top rating, and nearly 10% even gave their 7th ranked site "most want to go" status. The distribution for a "would like to go" rating varied more widely, with a high

of 66% clustered around 4th choice rankings. For a significant portion of students, perhaps as high

as 40%, the 2nd through 6th ranked choices all carry about the same preference weight. For this

reason, we believe that ratings may capture true student preferences better than rankings.
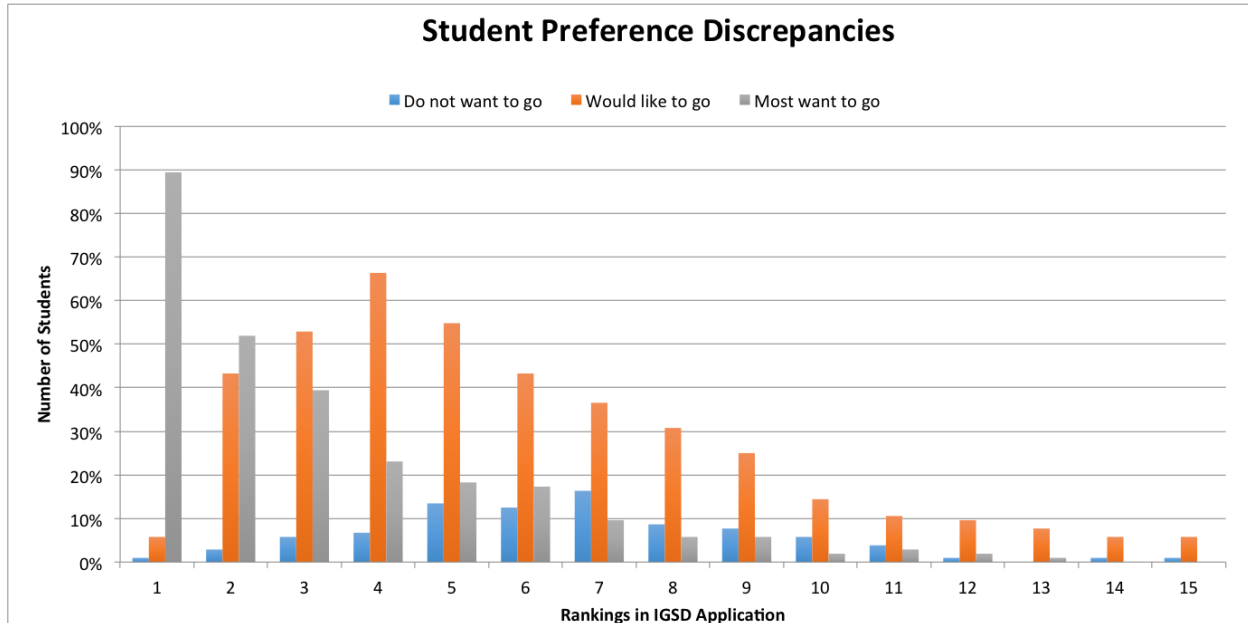
Figure 10 - Student Preference Discrepancies

Our model results from the student rating data were very different from the student ranking

data. Perhaps further reflecting the discrepancies in student preferences, while our inverse ranking

model shared 74% of outcomes with IGSD, our rating based model only shared 44%. However,

evidence suggests it bears a considerable mathematical and even psychological improvement on

student satisfaction. In the ratings model, the average student preference for their placement was

96%; IGSD's placements have an average preference of 82%. Furthermore, using the ratings, 92%

of students can be placed into a primary choice center, using the flexibility offered by students'

flexible preferences to create better outcomes for the group as a whole. In contrast, with only a

41

single first choice, no more than 76% of students were placed into their top center. Relying further on students' survey ratings, 97% of students in the ratings model received a placement they rated equal to or greater than IGSD's placement.

4.5 Benefits of the Tool

The matching tool provides students, directors and IGSD staff a variety of benefits, which it will be discussed in the following sections.

4.5.1 Student Benefits

The optimization model offers extensive benefits for student success centered placement. In addition to increasing the number of students who are placed in a primary choice site, and greatly reducing the wait list, it will also allow for more students to be considered in the program. Since so many sites fill up during the first round of applications, if a student is rejected from their first choice, they may have limited or no remaining options. Thus, under the current sequential placement process, even if a student applies to many sites, they may only be considered by few or one. Our proposed tool is not subject to this: the optimization model allows for simultaneous consideration all possible placements for all students, before any commitments are made. Simultaneous consideration would be infeasible to perform without an optimization model since it would exponentially increase the number of applications each director would be responsible for evaluating.

Changes to the placement process will also net a number of psychological benefits for the students. Under the sequential placement model, students are explicitly rejected from every site

they rank, until one selects them. For example, even if a student is selected by a site, if it is their 5th choice, they are still aware that they have been rejected by sites 1 through 4. This unnecessary rejection is avoided under our system; students are given a placement based on their preferences and their fit, without any explicit rejection from other sites.

Adoption of the system would also create more transparency for the students. IGSD could publicize the director preferences on their website to aid students in their application center. Through this, they would be able to know in advance of their applications what each project center director values, and how the decision will be made. This could better inform their decisions during, and better prepare students for, the application process. This would also enable them to tailor their application to emphasize specific qualities for specific sites, much like how applicants do in the job market.

4.5.2 Director Benefits

Currently, project center directors have to devote large amounts of time to evaluating and selecting student applications. For directors with very popular sites, there can be eighty or ninety applications, each one with individual essays and sometimes interviews. All of this takes a great deal of time and effort on the director's end. The sequential process also means that project center directors may miss out on promising candidates by accident of ranking, since they never fully see the pool of potential candidates.

4.5.3 IGSD Benefits

In addition to students and directors, our tool also provides benefits for the IGSD staff. The tool eliminates second round meetings and application review, ranking data collection review and analyzes, reference checking and possibly wait-list process. As a result, it decreases the amount of labor-hour needed in the current process. Although, the tool will still need the support of manual entries and evaluation such as during essay assessment, the amount of hours spent in the process will reduce drastically. The tool has the possibility of reducing 14 minutes per application, which when multiplied by 900, it saves IGSD and the directors 5 weeks of full time 8 hour work.

Another benefit of having an optimization-matching tool is the flexibility of generating different matching scenarios at any moment desired. In other words, our tool gives IGSD the ability to evaluate possible scenarios relatively quickly and inexpensively, without risk to the actual process. For example, currently students who are not placed in the first round go to a waitlist. Once all first round accepted students either accept or reject their placement, waitlisted students have the opportunity to re-apply to open spots, which their application needs to be revised by directors. With the tool, once waitlisted students re-rate available project centers, IGSD will only need to rerun the model with the new student data, as director preferences are already available in the model. This same ability to re-run scenarios could help IGSD to develop invaluable insights. They could use the model as a simulation tool in order to answer questions such as "what capacities do we need in which project centers for all students to get a primary choice site" or to see how placements change in response to considering different criteria. These experiments would be nigh impossible to conduct with the current manual process, which takes the involvement of dozens of

people the better part of a year; the decision support tool could run simulations of placements in order to answer questions or test ideas in only a matter of hours.

5. More Alternative Approaches: Diversity Focus

As we discussed in our Max Matching model (1) – (4), our objective is to maximize the total weighted matching score. However, people might have different focuses other than having the highest matching score. We explored other alternative approaches, and we focused on maximizing diversity in this section.

5.1 Maximizing Diversity in Forming Student Groups

Forming a collaborative team may benefit from having the best people in the team. However, there are a lot of ways to define "best". For example, sports teams need players with physical strength and executive teams need people with creative, applicable ideas. Companies are trying to become more competitive by integrating a variety of talents in their heterogeneous employee populations (Workforce 2000, 1987). An increase in diversity will benefit industries by increasing organizational effectiveness as well as enhancing employee problem solving and innovation skills (Maass et al., 2015). Therefore, it is important to understand the influence of the dynamics of workforce diversity on team outcomes (Jehn, 1995). Researchers at the University of Minnesota did examinations on papers with the topic of team diversity published from 1980 to present. They found that attributes highly related to employment such as functional expertise, education, and industry background had a stronger impact on team performance than less job-related attributes such as gender and ethnicity (Horwitz, 2005). Researchers concluded that bio-demographic attributes such as age and gender are less likely to affect team's performance. However, job-related attributes are more relevant to the team's performance. Thus, an important

point, according to Horwitz, is that simply creating diverse teams will not make them more effective; the success of teamwork depends largely on the right composition of the individual attributes.

5.2 Background

There is a study at California State University about how to develop a student assignment that takes students' diverse background into consideration (Bhadury et al., 1999). The management and finance department wanted to allocate MBA students to different project groups based on their educational backgrounds. Their objective was to help professors to develop an assignment that can reflect the most desirable mix of students' knowledge and skills (Bhadury et al., 1999). The authors used the concepts of max-flow and dining problem, where the dining problem is the challenge of deciding a seating arrangement at a restaurant table, so that adjacent persons have different backgrounds and personalities to improve social networking. For the study we are discussing here, they used a modified dining problem algorithm in a way that the number of students with same background that are assigned to any one of the project teams is as small as possible. The maximally diverse grouping problem (MDGP) focuses on assigning a set of elements to mutually disjoint groups in a way that the diversity among the elements in each group is maximized (Feo and Khellaf, 1990). The objective function of MDGP is to maximize the sum of the diversity score of all the groups.

In another context, there may be multiple objectives to optimize at the same time. A method known as diversity maximization approach (DMA) can be used for generating the efficient frontier for such multi-objective problems. This approach finds Pareto optimal solutions by making sure

that every time the new solution found is the most diverse/far away from the existing efficient points (Masin and Bukchin, 2008). This DMA approach can be used to solve both single and multi-objective problems. Furthermore, complex diversity maximization problems often involve the use of computational methods; in the paper "Workgroups Diversity Maximization", a model and a hybrid algorithm for maximizing workgroup diversity is presented, with the goal of finding the most diverse assignment by maximizing inter-group heterogeneity (Caserta and Voß, 2013). Last, a method known as minimizing similar attributes (MSA) is discussed in the paper "Maximizing Diversity in the Engineering Global Leadership Cultural Families" (Maass et al., 2015). The objective of MSA is to minimize the penalty scores due to deviation from user-solicited target input values, thereby maximizing the diversity in a group. The MSA approach is different from MDGP because it allows users to input the target number of times each attribute is present in a specific group. Another benefit with MSA is that this approach evenly balances the group size, which seems to be suitable for our case, where we try to place WPI students in different project centers.

5.3 Maximizing Diversity Using MSA

As discussed in the paper "Maximizing Diversity in the Engineering Global Leadership Cultural Families", MSA approach was applied in an academic setting. Researchers used the MSA model to assign 64 Engineering Global Leadership (EGL) students to the 2013 Cultural Families Program in a way that students have different attributes for each of the characteristics such as cultural core, religion and languages. They emphasized the importance and benefit to place students in diverse project teams in order to maximize student learning experience. Our situation

is also an academic setting. Even though we are trying to match students with IQP project sites instead of families program, the ideas are the same. So we decided to use this MSA approach along with our collected student and director information to solve for a solution which maximizes the diversity of students attributes.

Instead of having the actual attributes for each characteristic like Junior for Year and Christianity for Religion, we have assigned scores of scale 1 to 5 for each attribute per student. For example, a student has a score 4 in Leadership, which means this student is a very good leader. Since the problem setup is different, a modified version of the mathematical formulation of the model is shown in section 5.4.

5.4 Mathematical Formulation

The problem is to assign students to project centers such that the overall diversity is maximized by minimizing the similarity penalty scores.

5.4.1 Set Definition

S : Set of students, indexed by $s \in S$

P: Set of project centers, indexed by $p \in P$

K: Set of student attributes, indexed by $k \in K$

A_k: Set of scores of attributes, indexed by $a \in A_k$

5.4.2 Parameter Definition

$v_{akp} = $ Target number of score $a \in A_k$ in $k \in K$ for site $p \in P$

$f_{sp} = $ The preference of each student $s \in S$ for each site $p \in P$

$\overline{w}_{akp} = $ Penalty weight for over using score $a \in A_k$ for $k \in K$ for site $p \in P$

$\underline{w}_{akp}$ = Penalty weight for under using score $a \in A_k$ for $k \in K$ for site $p \in P$

$$b_{sak} = \begin{cases} 1 & if \ student \ s \ has \ score \ a \in A_k \ for \ attribute \ k \in K \\ 0 & otherwise \end{cases}$$

c_p = Capacity for each project center $p \in P$

5.4.3 Decision Variables

$$y_{sp} = \begin{cases} 1 & if \ student \ s \ is \ in \ project \ center \ p \in P \\ 0 & otherwise \end{cases}$$

$\overline{e}_{akp}$ = Number of additional over-usage of score $a \in A_k$ for $k \in K$ for site $p \in P$

$\underline{e}_{akp}$ = Number of additional under-usage of score $a \in A_k$ for $k \in K$ for site $p \in P$

5.4.4 Formulation

$$Minimize \ \sum_{k=1}^{|K|} \sum_{a=1}^{|A_k|} \sum_{p=1}^{|P|} \overline{w}_{akp} * \overline{e}_{akp} + \underline{w}_{akp} * \underline{e}_{akp} \tag{5}$$

Subject to

$$\sum_{s=1}^{|S|} y_{sp} \leq c_p \ \forall \ p \in P \tag{6}$$

$$\sum_{p=1}^{|P|} y_{sp} \leq 1 \ \forall \ s \in S \tag{7}$$

$$\sum_{s=1}^{|S|} b_{sak} * y_{sp} + f\left(\overline{e}_{akp}, \underline{e}_{akp}\right) = v_{akp} \ \forall \ k \in K, a \in A_k, \ p \in P \tag{8}$$

$$y_{sp} \leq 2 * f_{sp} \ \forall \ s \in S, \forall \ p \in P \tag{9}$$

$$\overline{e}_{akp} \geq 0 , \underline{e}_{akp} \geq 0 \ and \ y_{sp} \in \{0,1\} \ \forall \ k \in K, a \in A_k, s \in S, p \in P \tag{10}$$

The objective function (5) minimizes the sum of penalty scores over all attributes and project centers. Since we want to maximize the diversity, we want to minimize the deviation between the actual usage of students' attributes and our desirable level of usage. Constraint (6) makes sure the number of students assigned to each project center does not exceed its site capacity.

Constraint (7) states that each student can be placed at most once. Constraint (8) decides the values of placement variables. This constraint, in conjunction with the objective function, uses the idea of goal programming, and the right hand side of this equation specifies the goal value. This is the only constraint that involves the E variables, and it means we want the actual number of attribute scores usage to be equal to the target number of usage set by users by adding or subtracting E values. Deviation variable $\overline{e}_{akp}$ and $\underline{e}_{akp}$ are used to represent the possible deviations above or below the target value v_{akp}. In (4), $f(\overline{e}_{akp}, \underline{e}_{akp})$ is a function of $\overline{e}_{akp}$ and $\underline{e}_{akp}$, in our MSA model, we used $f(\overline{e}_{akp}, \underline{e}_{akp}) = -\overline{e}_{akp} + \underline{e}_{akp}$, but other functions like quadratic penalty functions could also be used. Constraint (9) means that if a student rate a site as "Do not want to go" then this student will not be sent to this site no matter what. We have three levels in our current ratings, 0, 0.5 and 1, and multiplying by 2 will make them 0, 1 and 2. Therefore, for those "do not want to go" sites, the constraint "$0 \leq 0$" will make sure no placement can be made for these sites. For all other sites, this constraint will have no effect at all. Last, constraint (10) is the variable declaration: $\overline{e}_{akp}, \underline{e}_{akp}$ are the number of over-usage or under-usage of student attributes so they need to be non-negative, while y_{sp} are the binary placement variables.

Formulation (5) – (10) modifies the original MSA formulation as follows. First, there is no lower bound for project site capacity. Second, we took out $y_{11} = 1$ and $w_f \geq w_{f+1}$ because there's no need to order student placements by sites.

5.5 User Control and Optimization Model

Our MSA model requires user input to enter target numbers V_{akp} of scores for each

attribute for each site. For instance, Denmark project site director wants only 2 students who have scores of 4 for leadership in his/her site. Some intuition here might be there is no need to have many students with good leadership skills because as they may not work well together. While this not being the case in general, it demonstrates the strong flexibility of the user-control target numbers. Other user input includes the penalty weight for over- or under-using attribute scores, which can be easily adjusted by users depending on different scenarios.

The MSA model differs from our original formulation (1) of maximizing the cumulative matching score, because it minimizes the total weighted penalty of deviating from the target numbers set by the directors.

6. More Alternative Approaches: Multiple Objectives

Our original formulation (1) - (4) emphasizes placing students with the highest matching scores to respective project sites. However, having students with the highest overall match scores may be shortsighted, because project centers may end up with students having many of same characteristics. This could be problematic because one of the goals of the IQP experience is for students with varying cultural background and personal skills are to go abroad and learn from one another. On the other hand, the MSA formulation (5) – (10) emphasizes diverse student placement for all the project centers. Therefore, a method to combine both of our objectives, maximizing matching scores and maximizing diversity, may be beneficial.

Our first approach is to use a ratio of matching scores and penalty scores. Let M be a

function representing matching scores, and P a function of penalty scores. Our objective will be to maximize the ratio $\frac{M}{P}$, because an increase in matching scores and a decrease in penalty scores will lead to an increase in the ratio as a whole, which is what we desire. While this objective function is theoretically appropriate, one drawback is it leads to a nonlinear objective function. In general, nonlinearities can be either convex or non-convex. While optimization problems with convex nonlinearities can be efficiently solved to identify a global optimal solution, there are no known approaches to handle general non-convex nonlinearities. As global optima are desirable, we would like to use a linear optimization model instead of a nonlinear one. Therefore, an alternative approach is to have a linear combination of these two objectives: $\alpha M - \beta P$. This function also ensures that we have to increase matching score or decrease penalty scores in order to maximize our objective function. By using this objective function, we can let project center directors to decide the values of α and β. For example, a project center director that would like to prioritize diversity of students can set α to be 30% and β to be 70%.

7. More Alternative Approaches: Balancing

Consider a student assignment problem where we have a project center and three students. In one situation we could have students with matching scores 10%, 30% and 60% assigned to this project center. In the other situation, we could have 30%, 35% and 35%. Most people would prefer the second assignment, which we call it a balanced solution.

7.1 Balancing vs. Best

The objective of any optimization problem is always to maximize or minimize a function, thereby representing the "best" result. However, we sometimes want to have a balanced solution instead of the best one. For a general example, managers in industries need to figure out a way to assign work to each worker. Each worker has different skills and has high efficiency doing some specific task. They want to not only best utilize workers' skills, but also make sure that the workload for each worker is about the same in order to avoid complaints due to unfairness. An example in our case could be a solution that might suggest sending two students with matching scores 80% and 20% to a specific project site, whereas the other solution might suggest sending two students with scores 50% and 50%. Obviously, the second solution is more balanced even though the total matching scores for the site are the same. We believe this "balancing" concept is helpful to think about depending on the kind of solution project directors want, do they only care about the total matching scores of all students accepted? Or, do they also care about individual matches, and whether there is a student with a matching score that is much lower than other accepted students with the site?

The concept of *maximin* or *minimax* is introduced to achieve this goal. Solving a balancing problem means to make the difference between the largest and smallest values as small as possible (Martello and Pulleyblank, 1984). In the previous example, this means the workload is almost evenly distributed among workers while making sure tasks are done efficiently by assigning the right workers to do the right jobs.

7.2 Implementation in Model

In our situation, IGSD wants to send the best students to project centers based on both students' and project directors' preferences. In the previous optimization models (1) – (4), we discussed maximizing the cumulative matching scores and the diversity of students. As we are solving the solutions simultaneously for all project sites, we want to have a balanced student placement solution for all these sites, which means we want to ensure that the smallest score for all the sites is as large as possible. This balancing problem has similar model as the Maximizing Matching Score model, except that it has a different object function and two more constraint sets.

Since we already shown the formulation for the Maximizing Matching Score Model, we would focus on the difference here. Recall our maximizing matching score model (1) – (4), all the sets and variables remain the same. We added in two more constraint sets and an objective function.

$$w_{sp} * y_{sp} + 1 * \left(1 - y_{sp}\right) \geq Q \ \forall \, s \in S, \ \forall \, p \in P \tag{11}$$

$$\sum_{s=1}^{|S|} y_{sp} \geq k\% * c_p \ \forall \, p \in P \tag{12}$$

Maximize Q (13)

In (11), **Q** is the lower bound of the matching scores for all the project sites. Constraint (11) ensures that the left hand side gets 100% when the student is not placed so it wouldn't affect or put any limitations on the value of **Q**. Constraint (12) indicates that we need at least **k**% of the students to be placed into project centers, and k could be a number in [80,90]. For example, it's reasonable to have more than 80% of students placed into project centers. We will have Q always be 100% if we don't have constraint (12). Because our ultimate goal is to achieve the highest matching score and a balanced solution at the same time, which means we want the lowest match score to be as large as possible, therefore this lower bound should be as large as possible as well.

8. Post-Optimality Analysis

Since we used multiple methods to solve this student placement optimization problem, we need a way to analyze the solutions and summarize the results. *Figure 11* below demonstrates the performance of the original objective function (1) and alternate objective functions (5), (13) and the IGSD placement according to a variety of metrics.

Model\Metric	Student Satisfaction	Director Satisfaction	Diversity Penalty Score	Avg. Std Match Score	Avg. Std Attribute Score	SD of Std Match Scores	Student get First Choice	Lowest Accepted Match
IGSD Manual Placement	86%	55%	5577	59%	49	0.22	78%	32.5
Max Matching	**97%**	**64%**	5654	**67%**	49	0.14	**93%**	32.5
MSA	65%	46%	**4575**	39%	49	0.21	26%	32.5
Balancing	100%	68%	5695	63%	50	0.22	89%	**39**

Figure 11 – Post Optimality Analysis Result

The comparison metrics reported in *Figure 11* are the student's satisfaction, directors' satisfaction, diversity penalty score, average placed student match score, average placed student attribute score, standard deviation of student match scores, percentage of students who get their first choice and the lowest accepted match score. Since IGSD's goal is to allow as many students as possible to experience global culture and apply their in-class knowledge and skills in solving real-world problems, they care about student satisfaction and so sending students to most preferred project centers is desirable. On the other side, the satisfaction of project center directors is also important. Directors have expectations and preferences for students with certain backgrounds. Moreover, student team diversity may be important because it will potentially increase the total value students bring in as a group and help them learn from each other with a variety of backgrounds. The average placed student score indicates the average matching level of the students who get placed based on students and professors' preferences. However, we believe it is helpful to see the placed student attributes scores regardless of students and professors preferences. We want to prevent a high standard deviation because it indicates that the solution points are spread out over a wide range of values. Furthermore, knowing the percentage of students who get their first choice is a useful way to understand how well each model works. Finally, the lowest accepted match gives us information about what is the lowest score among all the accepted students.

Students and directors' satisfaction are the most important factors when we match students with project centers. By looking at the *Figure 11* above, we proved again that our MaxMatching model (1) – (4) has the highest percentage in both student and director satisfaction. The balancing model has the highest number in satisfactions. However, the limitation in this model is that we

57

assumed we must assign k% of the students to project centers, in our case we set k to be 90. The diversity penalty score emphasizes on the diversity of student attributes. Note that small diversity penalty scores indicate greater diversity in the solution. The MSA model (5) – (10) yields a solution with a diversity penalty score 21% lower than the IGSD placement, 21.5% lower than the balancing model (13), 22% lower than the MaxMatching model (1) – (4). These numbers mean our MSA approach gives a solution pool of students that is 21%, 21.5% and 22% more diverse than the IGSD, balancing and MaxMatching models respectively.

We want to look at the average student match and attribute scores because they can represent how good the students are for all the sites in general. In *Figure 11* we can see that the MaxMatching model (1) – (4) has the highest average student match score of 67%. The average student attribute scores are almost the same for solutions of the four models, which are around 50 when 49 is also the average student attribute scores for all students.

For the standard deviation of student matching scores, our balancing model (13) has a relatively small number, which makes sense because this model tends to give us the most balanced solution. But we can also see that the MaxMatching model (1) – (4) does a good job as well. We also care about the percentage of students who get first choice. We can see in the table that our MaxMatching model gets the highest percentage, which proved that we should use this model for maximizing student first choices. Last, the balancing model (13) has the highest lowest accepted match score of 39, which is 20% higher than other models. This makes sense because the balancing model is optimizing for maximizing the minimum score.

The MaxMatching (1) – (4), MSA (5) – (10) and Balancing models (11) – (13) are useful

approaches that we can continue to test and improve. These three models have three different objective functions and therefore have different focuses. Based on the analysis, IGSD can use MaxMatching model (1) – (4) when they want to maximize student and directors 'satisfaction and aim for the students with highest matching scores. In the situation when IGSD wants to have a diverse student assignment, they can apply the MSA (5) – (10) model so that a project center can have students who have different backgrounds. If IGSD wants to improve on the "worst" accepted student, they can use the balancing model (11) – (13) to increase the lowest match score. In our analysis, we also notice that the current IGSD placement is doing good overall, but it didn't stand out in any of the eight metrics. Therefore, this current placement can be improved by using our suggested methods.

9. Sub-optimization

A lot of the times we would like to think about what is the second best option even when we already have the best option. In our case, after our model gives a solution of sending which students to a specific project center, project center directors might want to know the selected students better and take them to an interview or just dig in their application materials deeper. Suboptimal solution, the second best placement, will be very useful for those project directors who are not one hundred percent satisfied with the current selection of students and want to see an alternative option.

Optimization models can also be used as a laboratory for exploring systems (Camm, 2014).

In addition to the best solution, the best alternative solutions may also be found for users. To do this, we can add one more constraint in our algebraic formulation (11) - (13). Recall in our MSA (5) – (10) model, y_{sp} is the binary placement variable for student s and site p. Define y_{sp}^* to be the optimal solution and y_{sp} to be any other feasible solution. Consider the following constraint:

$$\sum_{s,p:y_{sp}^*=0} y_{sp} + \sum_{s,p:y_{sp}^*=1}(1 - y_{sp}) \geq 1. \tag{14}$$

Note that when $y = y^*$, $\sum_{s,p:y_{sp}^*=0} y_{sp}$ equals to zero and $\sum_{s,p:y_{sp}^*=1}(1 - y_{sp})$ equals to 0, which means the left hand side equals to 0. In fact, this is the only situation when we have the left hand side to be 0. Therefore, adding this constraint will ensure that the former optimal solution is not found, which means we will obtain a next-best optimal solution.

This method has successfully been tested in the MSA (5) – (10) and balancing model (13) and it can also be applied to Max Matching model simply by adding the constraint (14). Then the Max Matching model will have the following formulation:

$$\text{Maximize} \sum_{s=1}^{|S|} \sum_{p=1}^{|p|} W_{sp} * x_{sp} \tag{1}$$

Subject to

$$\sum_{p=1}^{|p|} x_{sp} \leq k \, \forall \, s \in S \tag{2}$$

$$\sum_{s=1}^{|S|} x_{sp} \leq C_p \forall \, p \in P \tag{3}$$

$$x_{sp} \in \{0,1\} \, \forall \, s \in S , \, \forall \, p \in P \tag{4}$$

$$\sum_{s,p:x_{sp}^*=0} x_{sp} + \sum_{s,p:x_{sp}^*=1}(1 - x_{sp}) \geq 1. \tag{14}$$

x_{sp} is our placement variable for student s and project center p in the Max Matching

model. When we plug in the original optimal solution into constraint (14), we have $\sum_{s,p:x^*_{sp}=0} x_{sp}$

equal to 0 and $\sum_{s,p:x^*_{sp}=1}(1 - x_{sp})$ also equal to 0. Therefore, we have the left hand side equal to

0. Since this is the only case when the left hand side equals to 0, making it not be 0 will give us a

different optimal solution, which is the next-best optimal solution that we want to obtain here.

10. Recommendations and Suggestions

10.1 Tool Implementation

The first step we suggest IGSD to take on in the implementation phase is to communicate to key stakeholders the reasons behind shifting from a manual to an automated IQP matching process. When communication the changes, it is important that IGSD is able to clearly and accurately emphasize the benefits of the tool, so all stakeholders are aware of its capabilities, are engaged, and are on board with the changes. Having all stakeholders who are part of the process on the same page will make the transition process smoother and easier to manage. To facilitate IGSD with the process transition, we completed many analyses to illustrate the benefits of our tool to directors as well as students as detailed in Section 4. Results and Analysis.

Moving from a paper-driven process to an automatic one can be challenging as people might have difficulty completely trusting technology. Therefore, we suggest IGSD gradually implement the tool and present results along the way to build trust in the system among stakeholders. An option is to implement a blended system next year, having the tool running beside the manual process. Comparing the tool results with the manual process throughout next year selection process might get directors and other stakeholders that might have been taken it back by adopting the technology at first, to actually trust and adopt the new tool. Another possibility is to have an MQP team next year running the tool alongside the current process, making enhancements to the tool while developing a marketing strategy to promote the tool. To successfully implementing the Decision Support System, all stakeholders need to embrace and use the new

technology. We believe that once the tool is fully implemented it will have a great and positive impact in the WPI community.

10.2 Standardization of Data

While very powerful, some VBA formulas are sensitive to unstandardized data. Therefore, to get the most effective results from the tool we recommend IGSD to standardize all the data input. For example, whenever dealing with the directors' preference data, we suggest IGSD to keep the names consistent in all locations. Our first suggestion is to use the project center location names instead of the directors' names in the data since different directors might advise the same project site. Additionally, project center names should be followed by the term in which the IQP will occur. For example, "Melbourne, Australia B17" and "Melbourne, Australia D18". It is important to concatenate the project center and the term and treat each one as different inputs since they have different capacities and different students applying to them. Our second recommendation is to keep the students first and last name consistent. For example, if IGSD decides to use first name followed by last name as "Whna Pnenonyyb", they should keep the same format in the entire input data.

10.3 Challenges to Proposed Process

Despite the strength of our proposed process, a number of challenges remain. Of utmost concern is that stakeholders will need to cultivate trust in and embrace the technology. The more the tool is trusted, the more useful it can become. A secondary concern is the lack of special circumstances or exceptions: the tool will treat all students the same, and any sort of preference

(for example, for a certain fraternity) will have to be explicitly added into the model. This also is a present limitation of the tool: its total dependence on honest preference reporting. Both students and directors must honestly record their preferences as ratings, since the tool will only act to satisfy stated preferences.

11. Reflections

Our team faced many challenges throughout our project, which we overcame by communicating well with our sponsors and advisor as well as by adapting to different conditions.

11.1 Design of Project Scope

Our first obstacle was designing the scope of our project. Originally this project was not going to be our Major Qualifying Project (MQP) but only an independent study with IGSD. However, as the scope of the project grew from a simple VBA tool to a powerful exercise in optimization and matching, it became clear that it would be very well suited to an MQP in the industrial engineering field. As the project unfolded we realized the IGSD student placement problem was a great opportunity for us to expand our understanding of theories we have been exposed to in courses such as optimization and apply it in a real-world setting. As a capstone design experience, the goal of the project was clear: create a decision support tool to assist IGSD in the student-director placement process. To design a functional tool, we had to understand the current placement process, search for similar models and approaches, develop a detailed mathematical description of our model, design a VBA code, and test and evaluate tool results. One of the greatest challenges during the design of the tool was quantifying qualitative data and converting it to numbers. There were lots of considerations during the quantifying process, such as how to represent each student by using numerical measures, and how to accommodate students' and directors' preferences. We had different discussions with IGSD staff, directors and our advisor, and tried different iterations of our model to finally come up with the current model design.

It was quite a challenge to finalize the categories and criteria, and deciding on their relationships. We struggled to decide which categories to use in order to better incorporate all the attributes of the students. For example, we originally considered 16 categories and 59 criteria, which is 75 in total. However, after a few iterations of the model we reduced to 25 categories and criteria in total, creating a more straightforward and concise model while including all the meaningful students' attributes that the project directors look for. To overcome this challenge and collect accurate information, we interviewed project directors for their insights about which criteria they use to select students. Although it was difficult to incorporate every director's different opinion and recommendation into our model, we believe we created a satisfactory methodology.

11.2 Constraints and Limitations in the Design

Our project was subject to a number of limitations. Our first issue was during the data collection process, which is a common problem for most projects. We had initial difficulty obtaining survey responses from both directors and students. We simplified our director survey after we revised our model, and sent it out along with our student survey at the end of B-term, which caused some response delays because people were busy with final exams.

Additionally, the actual obtained student placement, site capacity and student preferences ranking data from IGSD contained some inaccuracies. We first used the Rot13 cypher to anonymize student names to ensure that their information was kept confidential, and then we used the Excel function VLOOKUP to match student information to the anonymized student names. However, there were some discrepancies in this process, as students used different versions of the

same name in their application and survey. To overcome this challenge, we kept in touch with IGSD staff, Deborah Fusaro, who was able to correct the data and give us the right information.

Furthermore, we had some limitations in data representation when running the model. In our trial run, we had a sample of 104 students competing against each other for 35 sites. However, in the actual 2016-2017 IGSD acceptance process, there were more than 800 students competing simultaneously, which means the student tool placement results had limitations to accurately reflect the actual placement. However, we believe that the placement results obtained from the tool are still representative. Data limitations also include data format inconsistency between ours and IGSD files, which we overcame by standardizing data in all files to get the metrics calculations correct. Moreover, there was also gap between the stated and revealed preferences. Directors and students sometimes expressed one preference but acted on another. For example, the project director might claim that he/she wants students with top GPAs, but he/she actually accepted lots of students with average GPA but diverse cultural background or other standout personalities.

Another major challenge we faced in our MQP project was writing, testing and debugging our VBA code. We designed an Excel-based model with VBA code to create a user-friendly and flexible tool that could accommodate the changing number of directors and applicants each year. To make our code adjust to changing and updating sheets and cells as formulas do, we had to write a dynamic code. Parts of the code were designed individually at first, and then combined as a group. However, we each wrote the code in different formats. Therefore, we spent a good amount of time making sure that our VBA code was cohesive, dynamic and flexible to change. We also faced many difficulties during the debugging and testing process. To overcome this challenge, we

met as a group and took a closer look at the code to see whether or not we could find the problems. Whenever, we were unable to solve the issue, we stepped out of the code and came back to it in a later time.

11.3 Life-Long Learning

The tool that we have designed is worth exploring and developing to the next level because it has the potential to solve large similar problems. Universities always have this student placement problem. For example, they can use our model to assign students to robotics competition. But of course the goal and constraints here will be different. Our model is powerful because it can be broadly applied to not only academic settings, but also industrial or other type of companies. It focuses on solving a common problem, which is finding the best placement subject to limited resources. For applying it in other settings with the same essential idea but different constraints and objectives, this model needs to be adjusted to fit the new existing problem.

Overall, the project was a great learning and practicing experience, giving us the opportunity to prepare for the business world as well as strengthen our technical, communication, and interpersonal skills. While our team faced many challenges throughout the project, we overcame by communicating well with each other and our sponsor as well as keeping a list of tasks to accomplish during the project. The developing tool process enhanced our critical thinking and judgment skills. As we understood more about the users' needs and expectations, we designed a user-friendly tool to assist the decision makers in the student-director matching process.

11.4 Interdisciplinary Aspects

Our project utilized many interdisciplinary aspects. Our background research relied heavily on our English and research skills, while our process mapping drew from our business backgrounds. In analyzing the survey data, we used our knowledge from statistics classes to understand the information we were seeing. We applied our theoretical math knowledge in setting up the linear integer programming for the optimization model itself. For writing efficient and comprehensive VBA codes, we applied our computer science knowledge we learned in class and group projects. We believe blending these interdisciplinary aspects of the project have really added to both its breadth and depth, making for a stronger project and educational experience overall.

Bibliography

"About WPI." (2016). Retrieved October 24, 2016, from https://www.wpi.edu/about

"Bernard M. Gordon Prize." (n.d.). Retrieved March 03, 2017, from
http://wp.wpi.edu/projectbasedlearning/proven-pedagogy/bernard-m-gordon-award/

Bhadury, Joyendu, E.Joy Mighty, and Hario Damar. "Maximizing workforce diversity in project
teams: a network flow approach." *OMEGA: International Journal of Management
Science*, 28, pp 143-153 (1999)

Budish, E., Cachon, G. P., Kessler, J. B., & Othman, A. (2016). "Course match: A large-scale
implementation of approximate competitive equilibrium from equal incomes for
combinatorial allocation." *Operations Research*., in press.

Camm, Jeffrey D. (2014) "ASP, The Art and Science of Practice: A (Very) Short Course in
Suboptimization." *Interfaces* 2014, 428-431.

Caserta M., Voß S. (2013) "Workgroups Diversity Maximization: A Metaheuristic Approach."
In: Blesa M.J., Blum C., Festa P., Roli A., Sampels M. (eds) Hybrid Metaheuristics. HM
2013. *Lecture Notes in Computer Science*, vol 7919. Springer, Berlin, Heidelberg.

David Arnott, Graham Pervan. (2005) "A critical analysis of decision support systems research."
Journal of Information Technology. 20(2):67-87.

Dios M, Molina-Pariente J, Fernandez-Viagas V, Andrade-Pineda J, Framinan JM. (2015) "A
Decision support system for operating room scheduling." *Computer & Industrial
Engineering*. 88:430-443.

Feo, Thomas A., and Mallek Khellaf. (1990) "A class of bounded approximation algorithms for
graph partitioning." *Networks*. 20(2), 181–195.

"Global Projects Program." (2016). Retrieved October 24, 2016, from
https://www.wpi.edu/academics/undergraduate/project-based-learning/global-project-
program

Hanna M (2004). Principles of designing and developing spreadsheet-based decision support
systems. University of Florida, Dissertation.

Horwitz, Sujin K. "The Compositional Impact of Team Diversity on Performance: Theoretical
Considerations." Integrative Literature Review, June 2005.

"Interactive Qualifying Project." (2016). Retrieved October 24, 2016, from
https://www.wpi.edu/academics/undergraduate/project-based-learning/interactive-
qualifying-project

"Interdisciplinary & Global Studies." (2016). Retrieved October 24, 2016, from
https://www.wpi.edu/academics/departments/interdisciplinary-global-studies

Jehn, Karen. (1995) "A Multimethod Examination of the Benefits and Detriments of Intragroup
Conflict." *Administrative Science Quarterly*. 40(2): 256-285.

Krol M (2013). Decision Support Systems. In: Lerner KL, Lerner BW, eds. Computer sciences. 2nd ed. ed. Detroit: Macmillan Reference USA.

Maass, Kayse Lee, Vera Mann Hey Lo, Anne Weiss, and Mark S. Daskin. (2015) "Maximizing Diversity in the Engineering Global Leadership Cultural Families." *Interfaces*. 45(4), 293-304.

Mason, A.J., (2012) "OpenSolver – An OpenSource Add-in to Solve Linear and Integer Progammes in Excel", Operations Research Proceedings 2011, eds. Klatte, Diethard, Lüthi, Hans-Jakob, Schmedders, Karl, Springer Berlin Heidelberg: 401-406. Retrieved October 29, 2016, from http://dx.doi.org/10.1007/978-3-642-29210-1_64, http://opensolver.org

Martello, S., and W. R. Pulleyblank. (1984) "Balanced optimization problems." *Operations Research Letters* 3.5: 275-78.

Masin, Michael, and Yossi Bukchin. (2008) "Diversity Maximization Approach for Multiobjective Optimization." *Operations Research*. 56(2): 411-424.

"Most Popular Study Abroad Program." (2017). Retrieved February 10, 2017, from https://www.princetonreview.com/college-rankings?rankings=most-popular-study-abroad-program

Pierre, D. M. (2014). "Optimization." Retrieved October 29, 2016, from http://www.accessscience.com.ezproxy.wpi.edu/content/474000

Tierney J. "Do you suffer from decision fatigue?" The New York Times. (2011). Retrieved October 29, 2016, from http://www.nytimes.com/2011/08/21/magazine/do-you-suffer-from-decisionfatigue

Trapp, Andrew and Konrad, Renata. (2015) "Finding diverse optima and near-optima to binary integer programs." *IIE Transactions* 47, 1300–1312.

Tversky A, Kahneman D. (2011) Judgment under uncertainty: Heuristics and biases. In: *Managerial Decision Making*.117-24.

Villarica H (2012). "The chocolate-and-radish experiment that birthed the modern conception of willpower." The Atlantic.

Workforce 2000. "Work and workers in the 21st century." New York: Hudson Institute. 1987.

Appendix A: Student's Survey

IQP Student Survey

This survey along with other data from your application will be used to test a new proposed method of placing students at IQP project centers. The testing of the new proposed model will take place only after formal placement decisions for 2017-18 Global Projects Program have been decided and announced. The information you provide will be anonymized by IGSD and kept confidential. Please, fill out all the questions to the best of your knowledge, completely and honestly. There are 16 questions in total. It should take 5 to 8 minutes to complete. Thank you very much for your time.

*Required

Email address *

Your email address

What is your first name? *

Your answer

What is you last name? *

Your answer

Please rate all projects centers as Most want to go, Would like to go, or Do not want to go. *

	Most want to go	Would like to go	Do not want to go
Bar Harbor, Maine E17	○	○	○
London, England E17	○	○	○
Windhoek, Namibia E17	○	○	○
Worcester/Boston E17	○	○	○
Boston, Massachusetts A17	○	○	○

Copenhagen, Denmark A17	○	○	○
Moscow, Russia A17	○	○	○
Panama City, Panama A17	○	○	○
Santa Fe, New Mexico A17	○	○	○
Switzerland A17	○	○	○
Cape Town, South Africa B17	○	○	○
Hangzhou, China B17	○	○	○
Melbourne, Australia B17	○	○	○
Nantucket, Massachusetts B17	○	○	○
San Juan, Puerto Rico B17	○	○	○
Tirana, Albania B17	○	○	○
Venice, Italy B17	○	○	○
Washington D.C. B17	○	○	○
Worcester Center for Community Empowerment and Environmental Responsibility B17	○	○	○
Bangkok, Thailand C18	○	○	○
Cuenca, Ecuador C18	○	○	○
Hong Kong, China C18	○	○	○
Pioneer Valley, Massachusetts C18	○	○	○
Rabat, Morocco C18	○	○	○
San Jose, Costa Rica C18	○	○	○

Wellington, New Zealand C18	○	○	○
Worcester Center for Community Empowerment and Environmental Responsibility C18	○	○	○
Asuncion, Paraguay D18	○	○	○
Copenhagen, Denmark D18	○	○	○
London, England D18	○	○	○
Mandi, India D18	○	○	○
Massachusetts Water Resource Outreach Center, Worcester, MA D18	○	○	○
Melbourne, Australia D18	○	○	○
Pioneer Valley, Massachusetts D18	○	○	○
Thessaloniki, Greece D18	○	○	○
Windhoek, Namibia D18	○	○	○
Worcester Center for Community Empowerment and Environmental Responsibility D18	○	○	○
Worcester, England D18	○	○	○

NEXT

IQP Student Survey

If you volunteer, please describe your volunteer experience. What organization do you volunteer for? How did you become involved?

Your answer

Do you have any PLA/TA/Tutoring experience? If so, how much?

○ 2-3 years

○ 6-12 months

○ 3-6 months

○ 1-3 months

○ None

○ Other:

Do you have any work experience? If so, how much?

○ 2-3 years

○ 6-12 months

○ 3-6 months

○ 1-3 months

○ None

○ Other:

75

How much time have you spent outside of your home country or culture?

- ○ None
- ○ Days
- ○ Weeks
- ○ Months
- ○ Years
- ○ Other: _____

Do you speak more than one language? If so, which one(s)?

- ☐ Spanish
- ☐ Portuguese
- ☐ French
- ☐ German
- ☐ Mandarin
- ☐ Cantonese
- ☐ Russian
- ☐ Other: _____

What extracurriculars are you involved in? Please describe the extent and length of your involvement.

Your answer

Please briefly describe the work experience you are most proud of.

IQP Student Survey

What is your overall GPA?

○ 3.75-4.00

○ 3.50-3.75

○ 3.25-3.50

○ 3.00-3.25

○ 2.50-3.00

○ Below 2.50

Have you received any NRs? If yes, how many have you received?(Do not include NRs on Physical Education)

○ 0

○ 1

○ 2

○ 3

○ More than 3

Have you ever overloaded? If yes, how many terms?

○ 0

○ 1

○ 2

○ 3

What has been your average grade in your humanities classes?

○ All A's

○ Mix of A's and B's

○ All B's

○ Mix of B's and C's

○ All C's

○ N/A

What has been your average grade in your major classes?

○ All A's

○ Mix of A's and B's

○ All B's

○ Mix of B's and C's

○ All C's

○ N/A

How have your grades changed over time?

○ Improved a lot

○ Improved a little

○ Stayed consistent

○ Worsened a little

○ Worsened a lot

How many classes at the 3000-level or above have you taken so far?

○ None

○ One

○ Two

○ Three

○ Four

○ Five or more

Appendix B: Director's Survey

IQP Project Center Director Survey

This is a survey to help determine the aspects of student applications that are of interest to project center directors, and what criteria directors use to place students. This is part of an IGSD-supported project to consider an automated system that uses student and project center director preferences in order to make informed, improved matches. (This will take you approximately 5 minutes)

*Required

Email address *

Your email address

What is your name?

Your answer

What project center(s) do you direct?

Your answer

Please rate what categories of strengths are most important to you in considering student applications. *

	Very Important	Somewhat Important	Neither Important nor Unimportant	Somewhat Unimportant	Entirely Unimportant
Academic excellence	○	○	○	○	○
Teamwork / Cooperation	○	○	○	○	○
Communication / Writing	○	○	○	○	○
Initiative / Motivation	○	○	○	○	○
Openness to new experience	○	○	○	○	○

Openness to new experience	○	○	○	○	○
Good Time Management / Responsibility	○	○	○	○	○
Site-specific Language Requirement	○	○	○	○	○

Please rate how important each of these considerations are when evaluating a student's application. *

	Very important	Somewhat Important	Neither Important nor Unimportant	Somewhat Unimportant	Entirely Unimportant
Overall GPA	○	○	○	○	○
Consistency in passing classes	○	○	○	○	○
Humanities grade average	○	○	○	○	○
Major grade average	○	○	○	○	○
Number of challenging classes taken	○	○	○	○	○
Grade in GPS	○	○	○	○	○
Grade improvement over time	○	○	○	○	○
Extracurricular Involvement	○	○	○	○	○
Leadership Positions	○	○	○	○	○
Quality of Essay Writing	○	○	○	○	○
Content of Personal Essay	○	○	○	○	○
Presence/Confidence of Student	○	○	○	○	○
PLA/TA/Tutoring Work	○	○	○	○	○

Volunteering	○	○	○	○	○
Length of Work Experience	○	○	○	○	○
Quality of Work Experience	○	○	○	○	○
International Exposure/Experience	○	○	○	○	○
Site-Specific Language Requirement	○	○	○	○	○

Are there certain site specific requirements you have for your project center? (e.g., if language proficiency, what language)

Your answer

Do you have additional requirements or considerations not mentioned above?

Your answer

Appendix C: Matching Tool Instructions

Instructions for the Student-Director IQP Matching Tool

Step 1: The tool is based on OpenSolver add-in, but first you need to enable solver, which is a Microsoft Office Excel add-in program that is available when you install Microsoft Office. To use the Solver you need to do the following steps:

1. Click the **File** tab, and then click **Options** below the **Excel** tab.
2. In the **Excel Options** dialog box, click **Add-Ins**.
3. In the **Manage** drop-down box, select **Excel Add-ins**, and then click **Go**.
4. In the **Add-Ins** dialog box, select **Solver Add-in**, and then click OK.

After you have enabled the Solver add-In, Excel will auto-install the Add-in if it is not already installed, and the **Solver** command will be added to the **Analysis** group on the **Data** tab in the ribbon.

Then you need to install OpenSolver from https://opensolver.org/.
1. Download the OpenSolver Linear zip file
2. Extract all the files from the zip file to create your OpenSolver folder containing all the OpenSolver files. (Please do not place the OpenSolver folder on your desktop, as it seems to cause problems for some users in Window 7. Instead, place it in your Documents or Program Files folder.)
3. Double click on the OpenSolver.xlam file.
4. OpenSolver will appear in the Data tab.

After installing the OpenSolver, you need to add Developer tab into Excel.
1. Click the File tab.
2. Click Options.
3. Click Customize Ribbon.
4. Under Customize the Ribbon and under Main Tabs, select the Developer check box.

Next open the Visual Basic Editor, under the Developer Tab to establish a reference to the Solver add-in. In the Visual Basic Editor, with a module active, click **References** on the **Tools** menu, and then check **Solver** and **OpenSolver** under **Available References**.

For more information please go to: https://msdn.microsoft.com/en-us/library/office/ff196600.aspx

Note: The Solvers only need to be enabled once when using the same computer.

Step 2: Define categories and criteria in the "Parameters" sheet.
A. You can enter by typing the new categories and/or criteria in the table as shown below.
B. You can remove the categories and/or criteria by erasing the cell and resizing the table.

Categories ▼
Academic excellence
Teamwork / Cooperation
Communication / Writing
Initiative / Motivation

Criteria ▼
Overall GPA
Consistency in passing classes
Humanities grade average
Major grade average
Number of challenging classes taken
Grade improvement over time
Extracurricular/Volunteer
Leadership Positions
Quality of Essay Writing
Content of Personal Essay
PLA/TA/Tutoring Work
Length of Work Experience
Quality of Work Experience
International Exposure/Experience

C. Click the "Erase Table" button to delete the previous data.
D. Click the "Regenerate Table" button to generate new data.

Controls

Regenerate Table

Erase Table
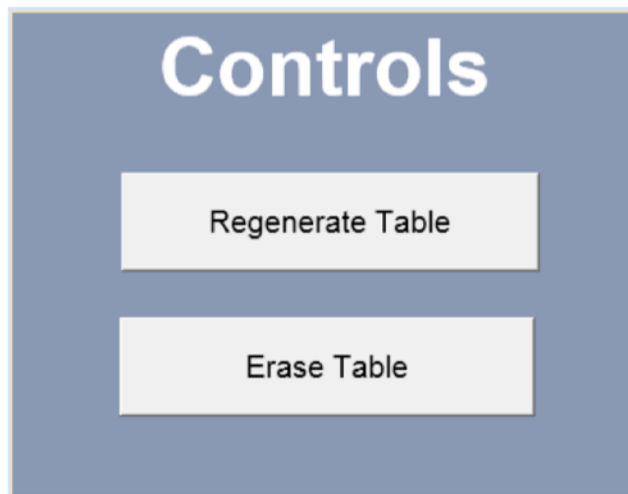
E. In "Category x Criteria Table", remove the criteria you think is unrelated to that category.

Category x Criteria Table Below

Academic excellence	Teamwork / Cooperation	Communication / Writing
Overall GPA		
Consistency in passing classes		
Humanities grade average		Humanities grade average
Major grade average		
Number of challenging classes taken		
Grade improvement over time		
	Extracurricular/Volunteer	
	Leadership Positions	
		Quality of Essay Writing
		Content of Personal Essay
	PLA/TA/Tutoring Work	PLA/TA/Tutoring Work
	Length of Work Experience	

The final table should look like this. Once the table is initialized the way you want, you can leave it as is. It does not need to be redefined each time the model is run.

Step 3: Enter Student and Director Information in "Student Info" and "Director Info" sheets.
1. Paste in the student information with right format.
    a. Criteria scores should be from 1 to 5.
    b. The headings of the table should be ID, followed by criteria names and lastly project center names.
    c. Student preferences for project centers should be 0, 0.5 and 1 (0: Do not want to go, 0.5: Would like to go and 1: Most want to go).
    d. Make sure all the criteria names in the "Student Info" sheet are the same and in the same as the ones in the "Parameters" sheet.
2. Paste in the project center information with right format.
    a. Category and criteria scores should be from 1 to 5.
    b. The headings should be project centers, category names, followed by criteria names.
    c. The capacity information should be located in the last column of the table.
    d. Make sure all the category and criteria names in the "Director Info" sheet is the same as the ones in the "Parameters" sheet.
    e. Make sure the project centers are in the same order as the ones in the "Student Info".

Note: You will need to ensure that the number and order of categories and criteria are consistent throughout the model. The same criteria and categories defined on the parameters sheet should be the column names in the student and director sheets.

Step 4: Run model and see results.

1. Follow the instructions in "Welcome" sheet.
2. Click "Run Model" button.
3. Look at the "IGSD Summary.xlsx" for a summary report including number of primary and secondary ratings, average student match scores, number of student selected and recommended students, etc.
4. Look at individual project center report for detailed student information.

Project Centers	Academic excellence	Teamwork / Cooperation	Communication / Writing	Initiative / Motivation		Total
Original Category Ratings	4	5	4	5	= 4 + 5 + 4 +5	18
Normalized Category Weights	= 4 / 18 22%	28%	22%	28%		

	Academic excellence	Teamwork / Cooperation	Communication / Writing	Initiative / Motivation
Overall GPA	1	0	0	1
Consistency in passing classes	1	0	0	1
Humanities grade average	1	0	1	0
Major grade average	1	0	0	0
Number of challenging classes taken	1	0	0	1
Grade improvement over time	1	0	0	1
Extracurricular/Volunteer	0	1	0	0
Leadership Positions	0	1	0	1
Quality of Essay Writing	0	0	1	0
Content of Personal Essay	0	0	1	0
PLA/TA/Tutoring Work	0	1	1	1
Length of Work Experience	0	1	0	0
Quality of Work Experience	0	0	0	1
International Exposure/Experience	0	0	0	0

	Academic excellence	Teamwork / Cooperation	Communication / Writing	Initiative / Motivation
Overall GPA	= 1 * 0.22 0.22	-	-	0.28
Consistency in passing classes	0.22	-	-	0.28
Humanities grade average	0.22	-	0.22	-
Major grade average	0.22	-	-	-
Number of challenging classes taken	0.22	-	-	0.28
Grade improvement over time	0.22	-	-	0.28
Extracurricular/Volunteer	= 0 * 0.22 -	0.28	-	-
Leadership Positions	-	0.28	-	0.28
Quality of Essay Writing	-	-	0.22	-
Content of Personal Essay	-	-	0.22	-
PLA/TA/Tutoring Work	-	0.28	0.22	0.28
Length of Work Experience	-	0.28	-	-
Quality of Work Experience	-	-	-	0.28
International Exposure/Experience	-	-	-	-

	Criteria Ratings	Academic excellence	Teamwork / Cooperation	Communication / Writing	Initiative / Motivation	Final Scores	Final Normalized Weights
Overall GPA	5.00	= 5 * 0.22 1.11	-	-	1.39	= 1.11 + 0 + 0 + 1.39 2.50	= 2.50 / 20.81 12%
Consistency in passing classes	4.00	0.89	-	-	1.11	2.00	10%
Humanities grade average	4.00	0.89	-	0.89	-	1.78	9%
Major grade average	4.00	0.89	-	-	-	0.89	4%
Number of challenging classes taken	5.00	1.11	-	-	1.39	2.50	12%
Grade improvement over time	4.50	1.00	-	-	1.25	2.25	11%
Extracurricular/Volunteer	3.00	-	0.83	-	-	0.83	4%
Leadership Positions	3.00	-	0.83	-	0.83	1.67	8%
Quality of Essay Writing	3.00	-	-	0.67	-	0.67	3%
Content of Personal Essay	3.00	-	-	0.67	-	0.67	3%
PLA/TA/Tutoring Work	4.00	-	1.11	0.89	1.11	3.11	15%
Length of Work Experience	4.00	-	1.11	-	-	1.11	5%
Quality of Work Experience	3.00	-	-	-	0.83	0.83	4%
SUM						= 2.50 + 2.00 +… +0.83 20.81	

Appendix E: Matching Tool VBA Code

```
Public Function CheckNames(SummaryName As String) As String
'Check to see if a workbook with the same name is open
For Each b In Workbooks
   If b.Name = SummaryName + ".xlsx" Then
      SummaryName = SummaryName + "(1)"
      CheckNames (SummaryName)
   End If
Next b
CheckNames = SummaryName
End Function
Sub Model()

'Define names for each worksheet in excel workbook
Dim StdSource As Worksheet
Dim Destination As Worksheet
Dim DrtSource As Worksheet
Dim OpModel As Worksheet
Dim Consider As Worksheet
Set StdSource = Sheets("Student Info")
Set DrtSource = Sheets("Director Info")
Set Matching = Sheets("Director-Student Matching")
Set Directors = Sheets("Director Preferences")
Set OpModel = Sheets("Optimization Model")
Set Consider = Sheets("Consideration Matrix")

'Define s to be number of students and d to be number of directors
Dim s As Long
Dim d As Long

'Bring all the calculations forward
Application.ScreenUpdating = False
Matching.Visible = True
Directors.Visible = True
OpModel.Visible = True
Consider.Visible = True

Directors.Cells.ClearContents
Matching.Cells.ClearContents
OpModel.Cells.ClearContents
```

```vba
'count number of students and directors from the input sheet
s = StdSource.Cells(Rows.Count, "A").End(xlUp).Row 'number of students
d = DrtSource.Cells(Rows.Count, "A").End(xlUp).Row 'number of directors

'Count number of categories and criteria, flexible and can be changed by users
Category = Sheets("Parameters").Cells(2, 5).Value
Criteria = Sheets("Parameters").Cells(2, 6).Value

'ERROR CATCHING BEGINS

'Test for missing student information
StdSource.Select
For r = 1 To s
   For C = 1 To Criteria + d - 1
      If IsEmpty(Cells(r, C)) = True Then
         Countsm = Countsm + 1
      End If
   Next C
Next r
If Countsm > 0 Then
   MsgBox ("There are " & Countsm & " blank cells in student info table.")
   Exit Sub
End If


'Test for non-quantified student criteria scores
For r = 2 To s 'for each student
   For C = 2 To Criteria 'for each criteria
      If Not (Cells(r, C) >= 0 And Cells(r, C) <= 5) Then '
         CountsCriteria = CountsCriteria + 1
      End If
   Next C
Next r

If CountsCriteria > 0 Then
   MsgBox ("There are " & CountsCriteria & " cells not in scale 1-5 in student criteria scores.")
   Exit Sub
End If

'Test if wrong format student preferences
For r = 2 To s 'for each student
   For C = Criteria + 1 To Criteria + d - 1 'for column preferences
```

```
      If Cells(r, C) < 0 Or Cells(r, C) > 1 Then
         CountsPref = CountsPref + 1
      End If
   Next C
Next r
If CountsPref > 0 Then
   MsgBox ("There are " & CountsPref & " cells not in scale 0-1 in student preferences.")
   Exit Sub
End If


'Test whether we have the right columns for criteria
For C = 2 To Criteria
   If Cells(1, C) <> Sheets("Parameters").Cells(C, 3) Then
      MsgBox ("Student Info: The first set of columns need to match criteria names in Parameter
Sheet.")
      Exit Sub
   End If
Next C


'Test for the same project centers
For C = Criteria + 1 To Criteria + d - 1
   If Cells(1, C) <> DrtSource.Cells(2 + (C - (Criteria + 1)), 1) Then
      MsgBox ("Student Info: The project centers headings should come after the criteria, or
project center names don't match the ones in Parameter Sheet.")
      Exit Sub
   End If
Next C


'Error catching in director info
'Test if we miss director info
DrtSource.Select
For r = 1 To d
   For C = 1 To Category + Criteria
      If IsEmpty(Cells(r, C)) = True Then
         Countdm = Countdm + 1
      End If
   Next C
Next r
If Countdm > 0 Then
   MsgBox ("There are " & Countdm & " blank cells in director info table.")
   Exit Sub
End If
```

```
'test if wrong format director category and criteria scores
For r = 2 To d
   For C = 2 To Category + Criteria - 1
      If Cells(r, C) < 0 Or Cells(r, C) > 5 Then
         CountdCateCri = CountdCateCri + 1
      End If
   Next C
Next r

If CountdCateCri > 0 Then
   MsgBox ("There are " & CountdCateCri & " cells not in scale 1-5 in director scores.")
   Exit Sub
End If

'test if wrong format capacity
For r = 2 To d
   If Cells(r, Category + Criteria) < 0 Then
      Countdc = Countdc + 1
   End If
Next r

If Countdc > 0 Then
   MsgBox ("There are " & Countdc & " cells with negative capacity.")
   Exit Sub
End If

'test whether we have the right columns for category, criteria and capacity
For C = 2 To Category
   If Cells(1, C) <> Sheets("Parameters").Cells(C, 1) Then
      MsgBox ("Director Info: The first set of columns need to match category names in
Parameter Sheet.")
      Exit Sub
   End If
Next C

For C = Category + 1 To Criteria - 1
   If Cells(1, C) <> Sheets("Parameters").Cells(2 + (C - (Category + 1)), 3) Then
      MsgBox ("Director Info: The criteria headings need to come after the categories, or criteria
names don't match the ones in Parameter Sheet.")
      Exit Sub
   End If
```

Next C

'DATA PROCESSING BEGINS

'Paste in student attribute scores
StdSource.Select
Range(Cells(1, 1), Cells(s + 1, Criteria)).Copy
Matching.Select
Range("B1").PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=False,
Transpose:=True
StdHeaders = Range(Cells(1, 2), Cells(1, s + 2))

'Paste in director categories
DrtSource.Select
Range(Cells(1, 1), Cells(d + 1, Category)).Copy
Directors.Select
Range("A1").PasteSpecial Paste:=xlPasteAll, Operation:=xlNone, SkipBlanks:=False,
Transpose:=True

DrtHeaders = Range(Cells(1, 1), Cells(1, d + 1)) 'Director Names
CatHeaders = Range(Cells(1, 1), Cells(Category + 1, 1)) 'Category Headers
Directors.Range(Cells(Category + 4, 1), Cells(Category * 2 + 4, 1)) = CatHeaders

'Paste in director criteria
DrtSource.Select
Range(Cells(1, Category + 1), Cells(d + 1, Criteria + Category - 1)).Copy
Directors.Select
Range(Cells(Category * 2 + 7, 1), Cells(Category * 2 + 7, d + 1)) = DrtHeaders
Range("A" & Category * 2 + 8).PasteSpecial Paste:=xlPasteAll, Operation:=xlNone,
SkipBlanks:=False, Transpose:=True
Row = Category * 2 + 6
CritHeaders = Range(Cells(Row + 1, 1), Cells(Row + Criteria, 1))

Row = Row + Criteria + 1 '39 here
Range(Cells(Row + 3, 1), Cells(Row + 2 + Criteria, 1)) = CritHeaders 'Normalized matrix
headers
Range(Cells(Row + Criteria + 6, 1), Cells(Row + Criteria * 2 + 4, 1)) = CritHeaders 'Normalized
matrix headers

'add all the project site names
Range(Cells(Row + 3, 1), Cells(Row + 3, 1 + d)) = DrtHeaders

Range(Cells(Category + 4, 1), Cells(Category + 4, 1 + d)) = DrtHeaders
Range(Cells(Row + Criteria + 6, 1), Cells(Row + Criteria + 6, 1 + d)) = DrtHeaders

```
'For each director
For C = 2 To d 'these are columns for each director
   For r = 1 To Category 'these are rows in the category matrix
      For dr = 1 To Criteria - 1 'these are rows in the atribute matrix
      'Combined Consideration Matrix section
      Directors.Cells(Row + dr + 3, C).FormulaR1C1 = "=R[-" & Criteria + 3 &
"]C*SUMPRODUCT(R" & Category + 5 & "C:R" & Category * 2 + 4 & "C,'Consideration
Matrix'!R2C" & 2 + dr & ":R" & 1 + Category & "C" & 2 + dr & ")"
      'Normalized formula for directors categories and atributes
      Directors.Cells(r + Category + 4, C).FormulaR1C1 = "=R[-" & Category + 3 &
"]C/SUM(R2C:R" & Category + 1 & "C)" 'normalized formula for categories
      Directors.Cells(dr + Row + Criteria + 6, C).FormulaR1C1 = "=R[-" & Criteria + 3 &
"]C/SUM(R" & Row + 4 & "C:R" & Row + Criteria - 1 + 4 & "C)" 'normalized formula for
atributes
      Next dr
   Next r
Next C
```

'Location: the row that the vectors of director weights start on on the directors sheet
Location = (Criteria + 1) * 2 + (Category + 1) * 2 + 10

'OPTIMIZATION MODEL BEGINS

```
'Student Scores Table
Dim ScoreStart As Integer
ScoreStart = 2 'start of scores table

'Copy sites and student names to optimization model sheet
Cells(ScoreStart, 1).Value = "Matches"
For r = 1 To d + 1
   Directors.Cells(1, r + 1).Copy 'Copy sites
   OpModel.Cells(ScoreStart + r, 1).PasteSpecial Paste:=xlValues, Operation:=xlNone,
SkipBlanks:=False, Transpose:=True
Next r
For C = 2 To s
   Matching.Cells(1, C + 1).Copy 'Copy students
   OpModel.Cells(ScoreStart, C).PasteSpecial xlPasteValues
Next C
```

```
'Write in matching scores
'Transpose student Info sheet
StdSource.Select
Range(Cells(1, 1), Cells(s, Criteria)).Copy
Matching.Select
Range(Cells(1, 1), Cells(Criteria, s)).PasteSpecial Paste:=xlPasteAll, Operation:=xlNone,
SkipBlanks:=False, Transpose:=True

OpModel.Select
For C = 1 To d - 1 'for each director
    'paste in matching scores
    For stu = 1 To s 'for each student
    'calculate matching
        Directors.Select
        dref = Range(Cells(Location, 1 + C), Cells(Location + Criteria - 2, 1 + C)).Address
        Matching.Select
        stuqual = Range(Cells(2, stu + 1), Cells(Criteria, stu + 1)).Address
        OpModel.Cells(ScoreStart + C, 1 + stu).Value = "=Sumproduct('Director Preferences'!" &
dref & ", 'Director-Student Matching'!" & stuqual & ")/5"
    Next stu
Next C

'Student Preferences Table
Dim PreferStart As Integer
PreferStart = ScoreStart + d + 2 '21

'Names for table, sites and students
OpModel.Select
Cells(PreferStart, 1).Value = "Student Preferences"

'Record student preferences
For C = 2 To s
    Cells(PreferStart, C).FormulaR1C1 = "=R[-" & d + 2 & "]C"
Next

'Make 0 preferences slightly negative to discourage placement
With Range(Cells(PreferStart + 1, 2), Cells(PreferStart + d, s))
    .Replace What:="0", Replacement:="-0.01", LookAt:=xlWhole, SearchOrder:=xlByRows,
MatchCase:=False
    .NumberFormat = "0.0"
End With
```

```
'Paste in student preferences
StdSource.Select
Range(Cells(1, Criteria + 1), Cells(s + 2, Criteria + d)).Copy
OpModel.Select
Range(Cells(PreferStart + 1, 1), Cells(PreferStart + d + 1, s + 2)).PasteSpecial Paste:=xlPasteAll,
Operation:=xlNone, SkipBlanks:=False, Transpose:=True

'Model Table
Dim ModelStart As Integer
ModelStart = PreferStart + d + 2 '78

'Names for table, sites and students
Cells(ModelStart, 1).Value = "Objective Function Coefficients"
For r = 1 To d - 1
   Cells(ModelStart + r, 1).FormulaR1C1 = "=R[-" & d + 2 & "]C"
Next r
For C = 2 To s
   Cells(ModelStart, C).FormulaR1C1 = "=R[-" & d + 2 & "]C"
Next

For r = 1 To d - 1
  For C = 2 To s
     Cells(r + ModelStart, C).FormulaR1C1 = "=R[-" & ModelStart - d - 4 & "]C * R[-" &
ModelStart - ScoreStart & "]C"
   Next C
Next r

'Solution Table
Dim SoluStart As Integer
SoluStart = ModelStart + d + 2 '114

'Names for table, sites and students
Cells(SoluStart + 1, 1).Value = "Model Assignments"

For r = 2 To d
   Cells(SoluStart + r, 1).FormulaR1C1 = "=R[-" & d + 3 & "]C"
Next r
For C = 2 To 1 + s
   Cells(SoluStart + 1, C).FormulaR1C1 = "=R[-" & d + 3 & "]C"
Next
```

```
'limits
For r = 1 To d
   Cells(SoluStart + 1, s + 1).Value = "TOTAL"
   Cells(SoluStart + r, s + 1).FormulaR1C1 = "=SUM(RC[-1]:RC[-" & s - 1 & "])"
   Cells(SoluStart + 1, s + 2).Value = "Capacity"
   Cells(SoluStart + r, s + 2).FormulaR1C1 = "='Director Info'!R" & r & "C" & Criteria +
Category
Next r

For C = 2 To s + 1
   Cells(SoluStart + d + 2, 1).Value = "TOTAL"
   Cells(SoluStart + d + 2, C).FormulaR1C1 = "=SUM(R[-" & d & "]C:R[-1]C)"
Next C

'objective function
Cells(SoluStart + d + 4, 1).Value = "Objective Function"
Cells(SoluStart + d + 3, 2).Value = "Model"
Cells(SoluStart + d + 4, 2).FormulaR1C1 = "=SUMPRODUCT(R[-" & d + 2 & "]C:R[-4]C" & s
& ",R[-" & 2 * d + 5 & "]C:R[-" & d + 7 & "]C" & s & ")"

'Define Ranges
Dim ObjectiveRange, VariableRange, CapacityLimitRange, CapacityRange, AssignLimitRange
As Range

Set ObjectiveRange = Range(Cells(SoluStart + d + 4, 2), Cells(SoluStart + d + 4, 2))
Set VariableRange = Range(Cells(SoluStart + 2, 2), Cells(SoluStart + d, s))
Set CapacityRange = Range(Cells(SoluStart + 2, s + 1), Cells(SoluStart + d, s + 1))
Set AssignRange = Range(Cells(SoluStart + d + 2, 2), Cells(SoluStart + d + 2, s))
Set CapacityLimitRange = Range(Cells(SoluStart + 2, s + 2), Cells(SoluStart + d, s + 2))

OpModel.Select
'
'Begin OpenSolver model
SolverReset
SolverOk SetCell:=Range(ObjectiveRange.Address), MaxMinVal:=1,
ByChange:=Range(VariableRange.Address), Engine:=1

'First Constraint: Variables are binary
SolverAdd CellRef:=Range(VariableRange.Address), Relation:=5

'Second Constraint: students cannot exceed site capacity
SolverAdd CellRef:=Range(CapacityRange.Address), Relation:=1,
```

```vba
FormulaText:=Range(CapacityLimitRange.Address)

'Third Constraint: each student can be sent to only X sites
X = Sheets("Welcome").Cells(70, 2).Value
SolverAdd CellRef:=Range(AssignRange.Address), Relation:=1, FormulaText:=X

'Run model
SolverOptions (IntTolerance = 0)
'SolverSolve
RunOpenSolver


'REPORTING BEGINS

'Define workbook names
Dim Calculations, Report As Workbook
'Start to generate result workbook for directors
Set Calculations = ActiveWorkbook
Dim Loc, Dest As String

Loc = ActiveWorkbook.Path
dt = "Recommendations_" + Format(CStr(Now), "mm-dd_hh.mm")
Dest = Loc + Application.PathSeparator + dt

MkDir Dest
ChDir (Dest)


Dim DirAssignRange, PreferRange, MatchRange, ab As Range
SoluStart = SoluStart + 1

Workbooks.Add
Set Summary = ActiveWorkbook
ActiveSheet.Cells(3, 1).Value = "Directors"
ActiveSheet.Cells(4, 1).Value = "# Primary Ratings"
ActiveSheet.Cells(5, 1).Value = "# Secondary Ratings"
ActiveSheet.Cells(6, 1).Value = "Average Student Match Score"
ActiveSheet.Cells(7, 1).Value = "Average Selected Student Match Score"
ActiveSheet.Cells(8, 1).Value = "# Students Selected"
ActiveSheet.Cells(9, 1).Value = "Original Capacity"
ActiveSheet.Cells(10, 1).Value = "Remaining Capacity"
ActiveSheet.Cells(12, 1).Value = "Recommended Students"
```

```vba
ActiveSheet.Name = "By Project Center"

'Student Outcome Sheet
ActiveWorkbook.Sheets.Add After:=Sheets("By Project Center")
ActiveSheet.Cells(1, 1).Value = "Student"
ActiveSheet.Name = "By Student Outcome"

MaxPlacements = 1

For C = 2 To s
    Place = 1 'First director to check in list
    Placements = 0 'Number of times they have been placed so far
    ActiveSheet.Cells(C, 1).Value = OpModel.Cells(2, C).Value 'Write student name
    For site = Place To d - 1 'check for accepted sites from those previously checked
    'If accepted at site, write results and increment placement
        If OpModel.Cells(SoluStart + site, C).Value = 1 Then
            ActiveSheet.Cells(C, 2 + 2 * Placements).Value = OpModel.Cells(SoluStart + site,
1).Value '

            'Write Site Preferences
            If OpModel.Cells(PreferStart + site, C).Value = 1 Then
                Pref = "Primary Choice"
            ElseIf OpModel.Cells(PreferStart + site, C).Value = 0.5 Then
                Pref = "Secondary Choice"
            End If
            ActiveSheet.Cells(C, 3 + 2 * Placements).Value = Pref

            'Increment
            Place = site
            Placements = Placements + 1
        End If
        If Placements > MaxPlacements Then
            MaxPlacements = Placements
        End If
    Next site
    If Placements = 0 Then
        ActiveSheet.Cells(C, 2).Value = "No Placement"
        ActiveSheet.Cells(C, 3).Value = "Waitlist"
    End If
Next C

Rows("1:1").Style = "Heading 4"
```

```
Columns("A:A").Style = "Heading 4"

ActiveSheet.Cells.Select
Selection.Columns.AutoFit

'Headers for the individual placements
For outcome = 1 To MaxPlacements
    ActiveSheet.Cells(1, 2 * outcome).Value = "Recommended Placement" & " " & Str(outcome)
    ActiveSheet.Cells(1, 1 + 2 * outcome).Value = "Preference for Site" & " " & Str(outcome)
Next outcome

Sheets("By Project Center").Select

Dim SelectedStudents

Dim Cap

For r = 0 To d - 2 'Loop for individual director reporting
Calculations.Activate
OpModel.Select

Dim Name As String

Name = Cells(SoluStart + r + 1, 1).Value 'name of director
Cap = Cells(SoluStart + r + 1, s + 2).Value 'Capacity of director
'find out what students are assigned to each center
Set DirAssignRange = Range(Cells(SoluStart + r + 1, 2), Cells(SoluStart + r + 1, s))
'find out what the student preferences are for each site
Set PreferRange = Range(Cells(PreferStart + r + 1, 2), Cells(PreferStart + r + 1, s))
'find out what the student match score is for each site
Set MatchRange = Range(Cells(3 + r, 2), Cells(3 + r, s))

'fill in information into new workbook
Workbooks.Add
Set Report = ActiveWorkbook
Report.Sheets("Sheet1").Select

'find out how many studentd rate this site as "most want to go"
ActiveSheet.Cells(1, 1).Value = Name
ActiveSheet.Cells(2, 1).Value = "Summary"
ActiveSheet.Cells(3, 1).Value = "Primary Preference for"
PrimPref = Application.CountIf(PreferRange, "=1")
```

```
ActiveSheet.Cells(3, 2).Value = PrimPref
ActiveSheet.Cells(3, 3).Value = "students"

'find out how many students rate this site as "would like to go"
ActiveSheet.Cells(4, 1).Value = "Secondary Preference For"
SecPref = Application.CountIf(PreferRange, ">0") - PrimPref
ActiveSheet.Cells(4, 2).Value = SecPref
ActiveSheet.Cells(4, 3).Value = "students"

'find out the average student matching score for the site
ActiveSheet.Cells(5, 1).Value = "Average Student-Director Matches"

ActiveSheet.Cells(6, 1).Value = "All Students"
AvgMatch = Application.Average(MatchRange)
ActiveSheet.Cells(6, 2).Value = AvgMatch

'average match score for those who rated the site as "most want to go"
ActiveSheet.Cells(7, 1).Value = "Students Marked Primary Preference"

If IsError(Application.AverageIf(PreferRange, "=1", MatchRange)) Then
ActiveSheet.Cells(7, 2).Value = "N/A"
Else
ActiveSheet.Cells(7, 2).Value = Application.AverageIf(PreferRange, "=1", MatchRange)
End If

'average match score for those who rated the site as "would like to go"
ActiveSheet.Cells(8, 1).Value = "Students Marked Secondary Preference"
If IsError(Application.AverageIf(PreferRange, "=0.5", MatchRange)) Then
ActiveSheet.Cells(8, 2).Value = "N/A"
Else
ActiveSheet.Cells(8, 2).Value = Application.AverageIf(PreferRange, "=0.5", MatchRange)
End If

'average match score for those who are accepted to the site
ActiveSheet.Cells(9, 1).Value = "Selected Students"
ActiveSheet.Cells(9, 2).Value = "=IFERROR(AVERAGE(C:C),0)"
SelMatch = Application.SumProduct(DirAssignRange, MatchRange)

'ActiveSheet.Cells(9, 2).Copy
Summary.Sheets("By Project Center").Cells(7, r + 2).Value = ActiveSheet.Cells(9, 2).Value

'ActiveSheet.Cells(7, r + 2).PasteSpecial Paste:=xlValues, Operation:=xlNone,
```

```
SkipBlanks:=False, Transpose:=False'
'Report.Sheets("Sheet1").Select

Reportrow = 11
ActiveSheet.Cells(Reportrow, 1).Value = "Recommended Students"
ActiveSheet.Cells(Reportrow, 2).Value = "Student Preference for Site"
ActiveSheet.Cells(Reportrow, 3).Value = "Director Match for Student"
ActiveSheet.Cells(Reportrow, 4).Value = "Final Match Score"

'Add director information to summary sheet
Summary.Activate
TotalAssigned = Application.WorksheetFunction.Sum(DirAssignRange)
ActiveSheet.Cells(3, r + 2).Value = Name 'directorname
ActiveSheet.Cells(4, r + 2).Value = PrimPref
ActiveSheet.Cells(5, r + 2).Value = SecPref
ActiveSheet.Cells(6, r + 2).Value = AvgMatch
ActiveSheet.Cells(7, r + 2).FormulaR1C1 = "=Iferror(" & SelMatch & "/R[1]C,0)"
ActiveSheet.Cells(8, r + 2).Value = TotalAssigned
ActiveSheet.Cells(9, r + 2).Value = Cap

ActiveSheet.Cells(10, r + 2).Value = Cap - TotalAssigned
ActiveSheet.Cells(12, r + 2).Value = "Recommended Students for " + Name

'paste in criteria
Calculations.Activate
Sheets("Parameters").Select
Range(Cells(2, 3), Cells(Criteria, 3)).Select
Selection.Copy

Report.Activate
ActiveSheet.Cells(Reportrow, 5).PasteSpecial Transpose:=True
Reportrow = Reportrow + 1

For C = 2 To s     'for each student

If OpModel.Cells(SoluStart + r + 1, C).Value = 1 Then 'if student accepted, write info to sheet
'Information from Opt Model
    ActiveSheet.Cells(Reportrow, 1).Value = OpModel.Cells(SoluStart, C).Value 'Student Name
    ActiveSheet.Cells(Reportrow, 2).Value = OpModel.Cells(PreferStart + r + 1, C).Value
'Student Preference for Site
    ActiveSheet.Cells(Reportrow, 3).Value = OpModel.Cells(2 + r, C).Value 'Director Match for
Student
```

```vba
    ActiveSheet.Cells(Reportrow, 4).Value = OpModel.Cells(ModelStart + r + 1, C).Value ' Final
Match score

    Summary.Activate
    ActiveSheet.Cells(Reportrow + 1, r + 2).Value = OpModel.Cells(SoluStart, C).Value
    Report.Activate
'Information from student sheet
    For crit = 1 To Criteria 'for each criteria
        ActiveSheet.Cells(Reportrow, crit + 4).Value = Matching.Cells(1 + crit, C + 1).Value
    Next crit
    Reportrow = Reportrow + 1
End If
Next C


'Formatting
ActiveSheet.Cells.Select
Selection.Columns.AutoFit
Cells(1, 1).Select
Selection.Interior.ColorIndex = 2
Rows("1:9").Interior.ColorIndex = 15

With Rows("9:9").Borders(xlEdgeBottom)
    .LineStyle = xlContinuous
    .Weight = xlMedium
    .ColorIndex = xlAutomatic
End With

Rows("1:11").Style = "Heading 4"
Columns("A:A").Style = "Heading 4"
Columns("C:D").Style = "Percent"
Range(Cells(6, 2), Cells(9, 2)).Style = "Percent"

Application.DisplayAlerts = False
Dim ProjName As String

ProjName = CheckNames(Name)
ActiveWorkbook.SaveAs Dest + Application.PathSeparator + ProjName + ".xlsx"
ActiveWorkbook.Close
Next r

'hide all the calculation sheets from users
```

```vba
Matching.Visible = False
Directors.Visible = False
OpModel.Visible = False
Consider.Visible = False


Summary.Activate
ActiveSheet.Cells(1, 1).Value = "IGSD Summary Report"

ActiveSheet.Cells.Select
Selection.Columns.AutoFit
Rows("6:7").Style = "Percent"
Rows("3:3").Style = "Heading 4"
Rows("12:12").Style = "Heading 4"
Columns("A:A").Style = "Heading 4"

Rows("6:7").Copy
Rows("6:7").PasteSpecial Paste:=xlValues, Operation:=xlNone, SkipBlanks:=False, _
Transpose:=False

Dim SummaryName As String

SummaryName = "IGSD Summary"

SummaryName = CheckNames(SummaryName)

ActiveWorkbook.SaveAs Dest + Application.PathSeparator + SummaryName + ".xlsx"

answer = MsgBox("Do you want to see the results?", vbYesNo + vbQuestion, "Optimization
Complete")

Calculations.Activate

'Return to Cell A1 on the active workbook on all sheets
Dim ws As Worksheet

For Each ws In ActiveWorkbook.Worksheets
ws.Activate
ws.Cells(1, 1).Select
Next ws

Sheets("Welcome").Select
```

```vba
If answer = vbYes Then
   Summary.Activate
   Cells(1, 1).Select
Else
   Summary.Close
End If

Application.DisplayAlerts = True
Application.ScreenUpdating = True

End Sub

Sub RunOpenSolver()

'Ask the user to open OpenSolver if it is not currently open.
   On Error GoTo errHandler_NoOpenSolver
   Application.Run "OpenSolver.xlam!RunOpenSolver"
   On Error GoTo errHandler
   Exit Sub
errHandler:
   Err.Raise Err.Number, Err.Source, Err.Description, Err.HelpFile, Err.HelpContext
errHandler_NoOpenSolver:
   Err.Clear
   MsgBox "This workbook requires OpenSolver, a free Excel addin available at
http://opensolver.org" + vbCrLf + vbCrLf + "Please install & then open OpenSolver, and then
try again.", vbOKOnly, "OpenSolver"
   Exit Sub
End Sub

Sub ClearTable()

'Resize table to small
Sheets("Parameters").ListObjects("CategoryTable").Resize Range("$A$1:$A$2")
Sheets("Parameters").ListObjects("CriteriaTable").Resize Range("$C$1:$C$2")

'Hide arrows
ActiveSheet.ListObjects("CriteriaTable").Range.AutoFilter Field:=1, VisibleDropDown:=False
ActiveSheet.ListObjects("CategoryTable").Range.AutoFilter Field:=1, VisibleDropDown:=False

Sheets("Parameters").Select
Range(Cells(2, 1), Cells(2, 3)).ClearContents
```

```
With Rows("3:100")
  .ClearContents
  .ClearFormats
  .Interior.ColorIndex = 2
End With

'Reformat
ActiveSheet.Cells.Select
Selection.Columns.AutoFit
ActiveSheet.Shapes.Range(Array("Group 1")).Select
Selection.ShapeRange.Height = 194.4
Selection.ShapeRange.Width = 288
ActiveSheet.Cells(2, 1).Select

End Sub
Sub ConsiderMatrix()

Sheets("Parameters").Select
Application.ScreenUpdating = False
Sheets("Consideration Matrix").Visible = True

'count how many categoried definded by users
Category = 1 +
Sheets("Parameters").ListObjects("CategoryTable").DataBodyRange.Rows.Count
'count how many criteria defined by users
Criteria = 1 + Sheets("Parameters").ListObjects("CriteriaTable").DataBodyRange.Rows.Count

'resize table after user adjusting
Sheets("Parameters").ListObjects("CategoryTable").Resize Range("$A$1:$A$" & Category)
Sheets("Parameters").ListObjects("CriteriaTable").Resize Range("$C$1:$C$" & Criteria)

Startrow = Application.WorksheetFunction.Max(Category, Criteria, 10) + 4 '22

Rows(Startrow - 3 & ":100").Select
Selection.Delete Shift:=xlUp

If Category = 1 Then
MsgBox ("You can't list 0 Category!")
Exit Sub

ElseIf Criteria = 0 Then
```

```
MsgBox ("You can't list 0 Criteria!")
Exit Sub

Else
ActiveSheet.ListObjects.Add(xlSrcRange, Range(Cells(Startrow, 1), Cells(Startrow + Criteria -
2, Category - 1)), , xl).Name = _
"BaseConsider"


Rows(Startrow).Font.ColorIndex = 0

'add in all criteria under each category for a start
For Cat = 1 To Category - 1
   Cells(Startrow, Cat).Value = Cells(Cat + 1, 1).Value
   ActiveSheet.ListObjects("BaseConsider").Range.AutoFilter Field:=Cat,
VisibleDropDown:=False
   For crit = 1 To Criteria
      Cells(Startrow + crit, Cat).Value = Cells(crit + 1, 3).Value
   Next crit
Next Cat
ActiveSheet.Cells.Select
Selection.Columns.AutoFit

ActiveSheet.Shapes.Range(Array("Group 1")).Select
Selection.ShapeRange.Height = 194.4
Selection.ShapeRange.Width = 288

'clear everything
Sheets("Consideration Matrix").Cells.ClearContents

'construct the consideration matrix
Dim CategoryNames As Range
Dim CriteriaNames As Range

Sheets("Parameters").Range(Cells(2, 1), Cells(Category + 1, 1)).Select
Selection.Copy

'paste in the category rows
Sheets("Consideration Matrix").Range("A2").PasteSpecial Paste:=xlValues, Operation:=xlNone,
SkipBlanks:=False, Transpose:=False

'paste in the criteria columns
```

```
Sheets("Parameters").Range(Cells(2, 3), Cells(Criteria + 2, 3)).Select
Selection.Copy
Sheets("Consideration Matrix").Range("B1").PasteSpecial Paste:=xlValues, Operation:=xlNone,
SkipBlanks:=False, Transpose:=True


'Process Consideration Matrix Based on Criteria listed under Categories
For Cat = 1 To Category - 1
   Sheets("Parameters").Select
   TableCol = Range(Cells(Startrow, Cat), Cells(Startrow + Criteria, Cat)).Address
   Sheets("Consideration Matrix").Select
   For crit = 1 To Criteria - 1
      Cells(Cat + 1, crit + 1).FormulaR1C1 = "=IF(ISNUMBER(MATCH(R1C,Parameters!R" &
Startrow & "C" & Cat & ":R" & Startrow + Criteria & "C" & Cat & ",0)),1,0)"
   Next crit
Next Cat

Sheets("Consideration Matrix").Visible = False
Sheets("Parameters").Select
Cells(Startrow - 2, 1).Value = "Cross Reference Category x Criteria Table Below"
Cells(Startrow - 2, 1).Font.Bold = True
Cells(Startrow - 1, 1).Value = "Remove the criteria in each column that don't belong to that
category"
Cells(Startrow - 1, 1).Font.Bold = True

Rows(Startrow + Criteria & ":100").Interior.ColorIndex = 2
Range("A1" & ":Z100").Interior.ColorIndex = 2

Application.ScreenUpdating = True
Cells(1, 1).Select

End If

End Sub
```