Improving Advisor Placement in IQP Centers via 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:

Kayla Fabry

ADVISOR:

Professor Andrew C. Trapp, Ph.D.

SPONSOR:

Global Experience Office (GEO)

Abstract

The goal of this MQP is to improve the off-campus Interactive Qualifying Project (IQP) advisor placement process for the Global Experience Office (GEO), formerly the Interdisciplinary and Global Studies Division (IGSD), of Worcester Polytechnic Institute (WPI). The GEO is part of WPI's Global School, and manages and operates many project centers throughout the world. One of the critical aspects of running project centers successfully is to ensure that they are staffed with appropriate advisors. This creates a challenge for those in the GEO responsible for matching IQP advisors to project centers each year because of the complexities of the process. Additionally, the current manual process is laborious and takes longer than desired for them to complete. They presently need to match advisors to approximately 100 spots at over 50 site-term combinations, while taking many different aspects into consideration including site preferences, experience levels, and language abilities. We designed a semi-automated solution for this challenge by considering the matching of advisors to project centers according to the preferences and characteristics of each advisor and the attributes of each project center. In our solution, we develop a novel integer optimization model and embed it within Visual Basic for Applications (VBA) in Excel to create a tool that will output recommended IQP advisor matchings while allowing for interactivity. Thus, the GEO faculty using our tool can adjust the output based on their deeper understanding of the matching process. We expect this tool to have a noteworthy impact on WPI's global project program by significantly reducing the time commitment without compromising their ability to control the outcomes of this process.

Acknowledgments

I would like to thank my sponsor, the GEO. I especially would like to acknowledge Dean Kent Rissmiller and the following GEO staff: Dawn Farmer, Deborah Fusaro, and Anne Oglivie for their insights and collaboration throughout this project.

I would also like to thank Professor Andrew C. Trapp, my project advisor, for all of the guidance, encouragement, and support he has given me throughout the time I have known him.

Table of Contents

Table of Figures

1. Introduction

“True optimization is the revolutionary contribution of modern research to decision processes” - George Dantzig, an American mathematician who made fundamental contributions to industrial engineering, operations research, computer science, economics, and statistics.

The Major Qualifying Project (MQP) at Worcester Polytechnic Institute (WPI) is an amazing opportunity for students to demonstrate what they have learned through their undergraduate education. For this MQP, to complete my degree in Mathematical Sciences, we developed an optimization model and system to help people in the Global Experience Office (GEO) at WPI make scheduling decisions. The GEO staff must decide which staff they send to advise students on each Interactive Qualifying Project (IQP) offered at WPI.

The IQP is an opportunity for WPI students to apply their skills to a real world project, either on campus or at one of the 44[1] project sites around the globe. WPI’s Global Projects Program, with a goal of having more than 90% of students participate in one or more off-campus projects, is well known and has won various awards. For example, it earned “the top spot in the 2017 Princeton Review rankings for study abroad programs” (Quinn-Szcesuil, 2017). Students who participate in the Global Projects program say that the experience was extremely meaningful, and that the “connections made between their work and the larger world are impressive” and impactful (Quinn-Szcesuil, 2017). It has also won the Gordon Prize for Innovation in Engineering and Technology Education from the National Academy of Engineering in 2016. This award is intended to recognize innovation in education methods that

[1] As of the 2020-2021 academic year.

grow and develop effective engineering leaders, and is only awarded once each year ("Bernard M. Gordon Prize"). In the term prior to their project, students must take two preparation courses: ID 2050 *Social Science Research for the IQP*, which teaches them how to properly conduct and analyze social science research, and PQP *Pre-Qualifying Project*, in which students begin research and planning of the specific project they will be completing at their project site.

The global IQPs require one or more faculty advisors to accompany students at project sites and provide them guidance throughout their project. These advisors must also be available to teach the PQP course in the term before the project's term. Advisors are currently chosen from three categories: GEO faculty, full-time faculty, and retired WPI faculty who are hired on a part-time basis. The GEO has also had WPI staff involved in co-advising projects with faculty, but the staff are not able to travel off-campus to project centers because they have full-time jobs on campus. GEO faculty are required to, as part of their job, either teach at least one of the IQP preparation courses or advise an IQP each term, so GEO faculty typically advise one or two projects each academic year. Any full-time faculty who will be working for WPI for the next academic year and would like to advise an IQP is eligible to apply to do so. These faculty would not be required to teach ID 2050 since, in most cases, they are not qualified to do so, but they would be responsible for teaching the PQP course.

The matching of advisors to sites is currently being done manually by Dean Kent Rissmiller, assisted by Dawn Farmer and Deborah Fusaro. He collects applications through a paper form and inputs the data into a decision matrix using an Excel spreadsheet. Then he matches them all by hand, which is a long and difficult task when considering he is typically matching about 95 potential advisors with about 50 site-term combinations. Unfortunately, the

GEO faculty and the full-time faculty from other departments who apply to advise IQPs are not numerous enough and not interested in enough sites, given the current matching system, to advise all of the IQPs. There are also a few sites that are considered less desirable and restrictions on how often applicants can advise IQPs that limit feasibility. For this reason, WPI has needed to consider alternate options to find a feasible solution, including bringing back a number of retired faculty who had previously advised an IQP to advise the current year's IQPs.

The sponsor for this MQP is the GEO at WPI. The GEO not only decides how to allocate IQP advisors, but they also are responsible for serving as a gateway for students and faculty to travel to communities around the world for all of the global projects that WPI offers. This includes all of WPI's IQPs and WPI's Global Projects Program, in which students can participate in a study abroad program to take courses at another university, complete their humanities requirement or their MQP, or participate in another global program. The GEO faculty are responsible for supporting and advising students participating in these projects at one of their 44 global project centers.

Our goal is to help the GEO match advisors to IQP sites in an optimal manner by maximizing the number of sites that have the correct number of advisors assigned to them. If multiple solutions exist with nearly the same number of unfilled positions, we want to prioritize solutions that also maximize utility. This means that we want to choose the matchings that allow the most advisors to advise in locations where they really want to advise, as opposed to sites where they would just be willing to advise. If we discover a solution where sacrificing a single placement leads to much greater utility, we will recommend that solution instead. We will do this by creating a tool that the GEO stakeholders will be able to use that will collect advisor applications and site information and output pairings of advisors with IQP sites. It will be built using an iterative process, in consultation with Dean Kent Rissmiller and Director of Team Learning Anne Ogilvie, key stakeholders in the GEO. The tool will also have to take into account the skills of specific advisors, such as foreign languages and whether the advisor has experience with advising an IQP. The tool will also allow for GEO stakeholders to observe the suggested matches and ask it to recalculate while preventing certain outcomes.

As an overview of the rest of this report: Chapter 2 will be dedicated to background research, in which we will discuss related projects, information about optimization and matching problems, and how the GEO is currently matching advisors to IQP sites. Chapter 3 will discuss methodology, in which we will describe the needs of the GEO, the form we will be using to collect data, our proposed optimization model, and an analysis of this model. Chapter 4 will describe the results and analysis of this MQP. Lastly, Chapter 5 will conclude this MQP with a discussion of contributions, lessons learned, limitations, and the potential for future research.

2. Background

This chapter will discuss projects related to this MQP, an introduction to optimization and matching problems, and the way in which the GEO is currently matching advisors to IQP sites.

2.1 Past Projects

By researching past projects completed at WPI, we discovered three MQPs that have significant similarities to this one. The titles were, in order of completion: “Improving the Scheduling of Operating Rooms at UMass Memorial Medical Center” (2014), “Finding Improving Solutions that Control Disruption to Binary Optimization Problems” (2014), and “Improving Student Placement in IQP Centers via Preference Matching” (2017).

The first MQP, “Improving the Scheduling of Operating Rooms at UMass Memorial Medical Center”, relates to this MQP through its use of scheduling to optimize utilization of resources. The students who completed this MQP wanted specifically to maximize the utilization of operating rooms (ORs) at UMass Memorial Medical Center (UMMMC) because they were operating at about 65% utilization, while the benchmark was set somewhere between 70-80%. They also had a similar goal to ours in that they wanted to produce multiple recommended schedules so the staff could determine which worked best for their purposes.

The next MQP, “Finding Improving Solutions that Control Disruption to Binary Optimization Problems”, overlaps with ours in its goal of simultaneously optimizing the objective function while trying not to dramatically change the proposed solution from the current conditions. This MQP, like ours, also utilizes Microsoft Excel's ability to solve optimization problems.

The third MQP, "Improving Student Placement in IQP Centers via Preference Matching", is by far the most similar to this MQP. The students who worked on this project were tasked with creating a decision tool to assist the GEO in matching students at WPI with IQP sites based on student preferences. This is very similar to our goal of creating a decision tool to assist the GEO in matching advisors at WPI with IQP sites based on advisor preferences. Additionally, the students behind this MQP also used Microsoft Excel in the creation of their decision support tool.

2.2 Optimization

The goal of optimization is to find the most effective way in which to complete a task. Examples of optimization can be found everywhere. Optimization is found in business, where organizations want to continue to produce the same item or service, but with a minimized cost, or in a way that will maximize their profits. Engineers use optimization when designing products to maximize the performance of their product while minimizing the costs of producing it (Yang, 2008). Optimization is also utilized in industrial design, scheduling problems, resource allocation, humanitarian aid, supply chains, and in many more applications.

One branch of optimization that is often found in our daily lives are routing problems. For example, UPS and other delivery services use optimization techniques to tell their drivers how to complete their routes in the minimum amount of time. Another type of optimization is matching problems. For example, Uber uses a matching system to ensure that all of their customers who are in need of rides are matched with drivers who are able to give rides.

No matter the application, the goal of optimization is always to obtain the best possible results such that the given constraints are satisfied. Optimization problems are typically represented with mathematical models. These models contain decision variables, which represent the decisions to make that influence the objective function and constraint equations, an objective function, a function of decision variables that dictates what is specifically trying to be maximized or minimized, and constraint equations, which restrict the decisions that are able to be made.

2.3 Matching Problems

Matching problems are a type of optimization problem, where the goal is to match up items in two different categories. Common examples include: matching customers to people who can service them, such as callers to a customer service line connecting with customer service representatives, matching employees with different time slots to create a schedule, matching students to courses they want to take, matching instructors to courses they are able to teach, and many more. Another example of a matching problem is this MQP, where the goal is to match advisors with sites they are able to advise.

Another very interesting matching problem comes from Alvin Roth, the Craig and Susan McCaw Professor of Economics at Stanford University, who utilized optimization in his proposed solution of the kidney exchange problem. He saw that many Americans were on the waiting list for a donor kidney, and that the list was so long that patients would often die while waiting for a transplant. He looked into the ways that someone could get a kidney, aside from the waiting list, and found that the only other source of transplants was from close friends and family of the patient. The issue with this is that, although patients might have multiple loved ones willing to donate to the patient, it was rare that they would be able to because in many cases loved ones are not a match. However, at some point a surgeon realized that, if a loved one was a match for another patient in need of a kidney, and if that patient's loved one was a match for the other patient, the loved ones could each donate a kidney to a stranger in exchange for a stranger donating a kidney to their loved one. Shortly after this idea began to be implemented, it was expanded upon to allow for more patient donor pairs to participate in an exchange.

However, this became more complicated and led to some concerns. For example, say matchings were made between 3 patients and three loved ones who were willing to donate. Then, in order for the exchange to take place, a transplant center would require 6 operating rooms to be available at the same time. It had to be done this way because, otherwise, a patient might receive a kidney, and then their loved one might change their mind about donating. Then, a person who donated their kidney would be left without a kidney for their loved one, and they would no longer have a kidney to donate in a new kidney exchange. So, the solution was to complete the exchanges all at the same time.

Unfortunately, there was another flaw with this system. In some cases, patients wouldn't have any loved ones who were willing to donate a kidney for them, and they wouldn't be able to participate in these kidney exchanges. This is where Professor Roth had the idea to allow for non-directed donors (Roth, 2018) to participate in this exchange. In his model, a good Samaritan would begin the process by volunteering to donate a kidney, with no particular patient in mind. Then, that person's kidney would be donated to a patient in a patient-donor pair waiting for a kidney exchange. Then, once the patient received the kidney, their loved one would donate a kidney to another patient in a patient-donor pair, and so on in a long chain. One benefit of this method is that exchanges don't all have to happen simultaneously, so the size of the chain is not restricted by hospital capacities. Additionally, the chain can keep going on and on until it reaches a patient who doesn't have a loved one to donate a kidney, which allows these patients to participate in the transplant chain instead of needing to wait for a donor kidney.

2.4 Decision Support Systems

A Decision Support System (DSS) is a tool that helps people make better decisions based on quantitative data. It will typically have a front end for the user, which will prompt them to provide answers to questions that will help the tool make decisions. Once the user has provided the input data, the DSS will take this information and find the best solution to the problem based on what it was given. Lastly, the DSS will then output the solution in a way that is clear and simple for the user (Power and Sharda, 2007). Decision Support Systems are useful because they take much less time to solve problems than a person usually does.

Decision support systems have been researched and developed for over 40 years, with results proving useful in the support of both semi-structured and unstructured problems. Despite this, and the enhanced decision making ability they provide to users, the increase in the breadth of DSS usage has seen limited adoption. They are most commonly found in corporate functions, with limited use within non-corporate areas (Shibl et. al., 2013).

These systems can become more challenging to create but also more useful to users when they are based on multiple criteria. Researchers in this field "are finding novel means to make

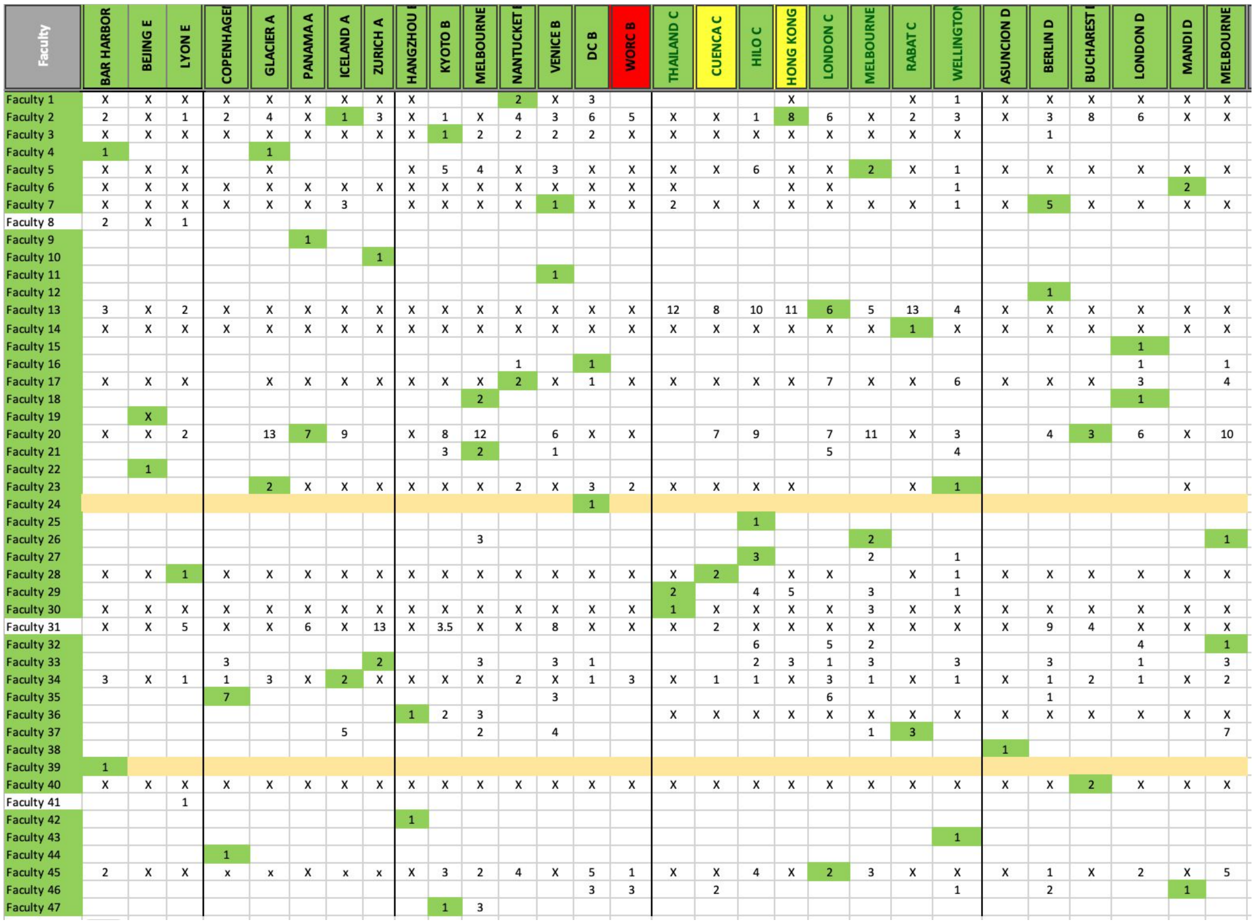
[them] by using multiple criteria in integration with machine learning and artificial intelligence" (Yang et. al., 2020). Decision support systems based on multiple criteria have the potential to support complex decision-making previously only undertaken by humans.

We will develop a DSS to help the GEO match advisor applicants to sites they can advise. The data will be collected through the application form, which will ask applicants about their site preferences as well as other topics that will help the DSS make a recommendation, such as language abilities and whether they have advised an IQP before. The DSS will then propose matchings that will minimize the number of sites that don't have enough advisors. It will also need to take into account how applicants rank each site so it can provide a solution that prioritizes advisors going to a site where they really want to go. So if the DSS determines that there are multiple solutions with the same number of unfilled positions, which will be zero if all positions are filled, it will examine each solution and determine which solution creates more overall site placement satisfaction by counting which solution has the most advisors placed at sites in which they are very interested. This data will be outputted in a way that is simple and easy to read for the GEO stakeholders. Additionally, our tool will allow the GEO stakeholders to be able to view the proposed solution and, if they see something they do not like, to prevent that outcome and re-run the system. We also include the ability for the GEO stakeholders to keep an outcome if they see something they do like.

2.5 Current Matching Process

Before we can begin to create a matching tool, we need to take a close look at how the GEO is currently making these matches. As of right now, the matches are being made using a decision matrix on an Excel spreadsheet. The initial matching process typically takes 4-6 weeks, and is followed by a much longer process of finding more advisors to fill the unfilled advisor positions. Attached below is a photograph of their current spreadsheet, with faculty names down the first column and site names across the first row.[2] Faculty names that are highlighted in green indicate that they were assigned to advise one or more IQP sites, and faculty names that are left white were not assigned to advise anywhere (e.g. Faculty 8). Site locations highlighted in green indicate that the location has the correct number of assigned advisors, which is usually two but can be one for smaller sites such as Lyon in E term. Sites highlighted in yellow means only one advisor was assigned to that location but the site requires two advisors, such as Cuenca and Hong Kong in C term. Sites highlighted in red indicate that no advisors have been assigned to that site, such as Worcester in B term. Looking inside the decision matrix, numbers or x's highlighted in green indicate that the faculty member whose row the green mark is in was assigned to the site at the top of that column. In a couple of instances, an entire row is highlighted in beige except for one highlighted green box with a 1 in it (e.g. Faculty 24). This indicates that the faculty member had retired from WPI, but was asked to come back to advise an IQP. The GEO reports that they would like to avoid having to bring back retired faculty whenever possible.

[2] Note that faculty names have been replaced by "Faculty number" to preserve anonymity, and that some faculty members and site locations have been removed for simplification purposes.



Faculty	BAR HARBOR	BEJING E	LYON E	COPENHAGE	GLACIER A	PANAMA A	ICELAND A	ZURICH A	HANGZHOU	KYOTO B	MELBOURNE	NANTUCKET	VENICE B	DC B	WORC B	THAILAND C	CUENCA C	HILO C	HONG KONG	LONDON C	MELBOURNE	RABAT C	WELLINGTON	ASUNCION D	BERLIN D	BUCHAREST	LONDON D	MANDI D	MELBOURNE
Faculty 1	X	X	X	X	X	X	X	X	X			2	X	3					X			X	1	X	X	X	X	X	X
Faculty 2	2	X	1	2	4	X	1	3	X	1	X	4	3	6	5	X	X	1	8	6	X	2	3	X	3	8	6	X	X
Faculty 3	X	X	X	X	X	X	X	X	X	1	2	2	2	2	X	X	X	X	X	X	X	X	X		1				
Faculty 4	1				1																								
Faculty 5	X	X	X		X				X	5	4	X	3	X	X	X	X	6	X	X	2	X	1	X	X	X	X	X	X
Faculty 6	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X			X	X			1					2	
Faculty 7	X	X	X	X	X	X	3		X	X	X	X	1	X	X	2	X	X	X	X	X	X	1	X	5	X	X	X	X
Faculty 8	2	X	1																										
Faculty 9						1																							
Faculty 10								1																					
Faculty 11													1																
Faculty 12																									1				
Faculty 13	3	X	2	X	X	X	X	X	X	X	X	X	X	X	X	12	8	10	11	6	5	13	4	X	X	X	X	X	X
Faculty 14	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X	X	X	X	X	X	X
Faculty 15																											1		
Faculty 16												1		1													1		1
Faculty 17	X	X	X		X	X	X	X	X	X	X	2	X	1	X	X	X	X	X	7	X	X	6	X	X	X	3		4
Faculty 18											2																1		
Faculty 19		X																											
Faculty 20	X	X	2		13	7	9		X	8	12		6	X	X		7	9		7	11	X	3		4	3	6	X	10
Faculty 21										3	2		1							5			4						
Faculty 22		1																											
Faculty 23					2	X	X	X	X	X	X	2	X	3	2	X	X	X	X			X	1					X	
Faculty 24														1															
Faculty 25																		1											
Faculty 26											3										2								1
Faculty 27																		3			2		1						
Faculty 28	X	X	1	X	X	X	X	X	X	X	X	X	X	X	X	X	2		X	X		X	1	X	X	X	X	X	X
Faculty 29																2		4	5		3		1						
Faculty 30	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	1	X	X	X	X	3	X	X	X	X	X	X	X	X
Faculty 31	X	X	5	X	X	6	X	13	X	3.5	X	X	8	X	X	X	2	X	X	X	X	X	X	X	9	4	X	X	X
Faculty 32																		6		5	2						4		1
Faculty 33				3				2			3		3	1				2	3	1	3		3		3		1		3
Faculty 34	3	X	1	1	3	X	2	X	X	X	X	2	X	1	3	X	1	1	X	3	1	X	1	X	1	2	1	X	2
Faculty 35				7									3							6					1				
Faculty 36									1	2	3					X	X	X	X	X	X	X	X	X	X	X	X	X	X
Faculty 37							5				2		4								1	3							7
Faculty 38																								1					
Faculty 39	1																												
Faculty 40	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	X	2	X	X	X
Faculty 41			1																										
Faculty 42									1																				
Faculty 43																							1						
Faculty 44				1																									
Faculty 45	2	X	X	x	x	X	x	x	X	3	2	4	X	5	1	X	X	4	X	2	3	X	X	X	1	X	2	X	5
Faculty 46														3	3		2						1		2			1	
Faculty 47										1	3																		

Figure 1: Previous Spreadsheet used by GEO Stakeholders

The data in this spreadsheet comes from the applications faculty members fill out, which is currently conducted through a paper form. The form prompts the faculty to rank their choices in preference order (1, 2, 3, etc.) and to put an X next to any sites they would not consider. Unfortunately, this method of data collection results in a variety of answer types. For example, some faculty leave everything blank and just check one or more sites they want to advise (e.g. Faculty 4 and 9). Other faculty rank their site preferences and then put an x for the rest of the sites (e.g. Faculty 13 and 14), which is how the form is meant to be filled out. However, many of the faculty do a combination of these, where some sites are ranked, some have an x, and the rest are blank (e.g. Faculty 1, 20, and 36). Also, the way in which the faculty rank their preferences varies. The faculty members generally rank a site with a 1 if the site is their first choice, a 2 if the site is their second choice, etc. However, there are a couple of unusual rankings which stray from this pattern. For example, looking at Faculty 2's rankings, we see that there are three 1's, three 2's, four 3's, two 4's, one 5, three 6's, etc. Or, looking at Faculty 31, we see that they ranked one site with a 3.5. These variances in answers makes them a lot more difficult to interpret and use to make matches on such a large scale.

This current method of matching advisors to IQP sites has many noteworthy benefits. It provides faculty with the opportunity to specifically state their preferences for which sites they would like to advise. However, when solving a matching problem like this, it is generally more helpful if rankings are less specific and more consistent. Another reason this spreadsheet is useful is it allows the GEO stakeholders to see all of the components of the problem at once. On the other hand, this might turn out to be a little overwhelming with the quantity of potential faculty advisors and global sites. This is where opportunities for improvement with operations analytics and optimization arise.

3. Methodology

This chapter will discuss methodology, in which we will describe the needs of the Global Experience Office (GEO), the form we will be using to collect data, and our proposed optimization model.

3.1 Understanding the Needs of the GEO

To improve our understanding of the needs of the GEO, we began by meeting with GEO stakeholders to discuss their present process, what was working well for them, what wasn't, and what could be achieved through this MQP. From this initial meeting we first wanted to understand how they were currently making these advisor-site matches. We learned that the current application is done through a printed out paper form. The biggest drawback of this system, according to the GEO stakeholders, is that applicants tend to fill out the form in a variety of different ways. Therefore, our application form should prevent different response types (e.g. by using an electronic form that allows applicants to drag and drop various site-term combinations into ranking categories). Additionally, it was requested that the sites be listed as "site, term" in this form, as some sites are offered during multiple terms and applicants need to be able to indicate for which terms they are able to travel. We also learned that the GEO faculty usually advise one to two times per year, so the form could potentially allow for GEO faculty to indicate if they prefer to advise one or two times. This might be done through an open answer question, because an applicant might be willing to go away twice only if certain other preferences are observed (e.g. someone might be willing to travel twice if at least one of the sites was domestic).

The GEO stakeholders also conveyed to us some of the general preferences they take into account when making advisor-site matchings. One important note is that advisors cannot advise during back-to-back terms because they need to be on campus to advise the PQP course in the term before they go away. Additionally, when assigning two advisors to a site, it is preferred that at least one of them has advised an IQP before so that there are not two novices together at the same site. The GEO stakeholders informed us that they also frequently need to bring back retired faculty who have advised an IQP before so there are no sites that are short on advisors. They expressed that, whenever possible, they would like to avoid bringing back these retired faculty. This would be taken into account in our model by having it solve for all of the current faculty first, and then using the retired faculty to fill in the gaps. We also learned that certain advisors have known restrictions, so we will need to give advisors a space to describe any unique preferences. We might also need a space for the GEO stakeholders to prevent certain outcomes based on these preferences.

Next we discussed specifications for various sites. From this we learned that, although the majority of sites require two advisors, a small minority need only one advisor per site. The number of advisors required for each site is determined before the advisor-site matchings take place, so ideally the tool will be able to let the GEO stakeholders input this data. Also, each project center has one or two directors who are WPI faculty members. These project center directors have priority in going to their own center if they would like to, so the tool must match directors with their own sites if requested before making any other matchings. Another consideration of specific sites is whether there are language restrictions. We learned that, if possible, the GEO wants at least one advisor who speaks the site's language to travel to that site because it makes communication easier, but for most sites it is not required that either advisor has this ability. The last site specific consideration we discussed was whether we should add incentives for advisors of less desirable project centers. This could help resolve the lack of applicants willing to advise at these locations and reduce the number of sites that do not have enough advisors. These incentives would be listed in the application form, so applicants can take them into account when listing their preferences. The "undesirable" sites change year to year, so the form would need to allow the GEO stakeholders to update this list of sites and their incentives.

We also asked about what other features the GEO stakeholders were hoping to see in this tool. They expressed that they wanted it to be interactive, to the extent possible, so they could see the tool's output, choose to prevent certain matchings and include others by typing a "Y" to force a matching or an "N" to prevent a matching in the cell corresponding to that advisor-site match, and ask it to re-solve. They conveyed that they might also like the option to see multiple recommended matchings (e.g. be able to view the top five solutions). They would also like these matchings to be displayed in an easy to read manner. We learned that we cannot require applicants to identify a minimum number of sites they would be willing to travel to because there are some applicants who are only willing to advise if they go one place. Instead they suggested that we include a note using persuasive language at the start of the form to encourage them to list more sites, perhaps by providing quotes from past participants. We could also suggest that they list multiple sites from a single term for which they are able to go away. The main message of this note at the top of the form should be to explain that the more options they provide the greater the benefit for the whole system.

3.2 Collection of Data

We use an online application form to collect data from potential advisors. This form is built in Qualtrics. From this survey we learn the applicant's name, email, gender identity, employment status, department affiliation, director status, experience level, term availability, language abilities, and specific site-term preferences. Images of this Qualtrics survey can be found in Appendix A: Advisor's Survey.

Once the applications were complete, we used Visual Basic for Applications (VBA) to automate the matching process. VBA is a programming language in Excel that allows users to create "dynamic applications which can receive user input for the model base component" (Hanna, 2004) of a spreadsheet-based optimization model. It also has several important features including "recording macros and working with variables, procedures, programming structures, and arrays" (Hanna, 2004). VBA can also be used to create a user interface for a clear and professional output.

3.3 Optimization Model

We will now present our mathematical model, leading with the sets, parameters, and variables used. This model takes in the advisor survey's output as input data. It then mathematically finds the solution that will maximize the objective function's value, subject to the provided constraints.

Set Definitions:

- L: set of all languages, indexed by ℓ
- A: set of all advisors, indexed by i
  - Subset A_{FT} are full-time faculty members
  - Subset A_{GEO} are GEO faculty members
  - Subset A_E are experienced advisors
  - Subset A_ℓ are advisors who speak language ℓ
  - Subset A_{DP} are advisors who are scheduled to advise D term of the previous year
  - Subset A_{DY} are directors who want to advise at their site
- S: set of all site-terms, indexed by j
  - Subsets S_ℓ are site-terms that have language restrictions, needing language ℓ
  - Subsets S_E, S_A, S_B, S_C, and S_D contain all site-terms for terms E, A, B, C, and D
  - Subsets $S_{DY,i}$ are site-terms where advisor i is a director for that site
  - Subsets $S_{NI,i}$ are site-terms where advisor i is not interested in advising at that site

Parameter Definitions:

- $\lambda \in [0, 1]$, weight of advisor preferences (default $\lambda = 0.05$)
  - λ closer to 0 will give preference to the first component of the objective function

- $a_{ij} \in \{0,1,2\}$, advisor i's preference of going to site j
  - 0 if not interested, 1 if interested, and 2 if very interested
- $\mu \in [0,1]$, weight of experienced advisor constraint (default $\mu = 0.05$)
  - μ closer to 0 will give preference to the first component of the objective function
- $c_j \in \{1,2\}$, advisor capacity of site j

Variable Definitions:

- $x_{ij} \in \{0,1\}$, 1 if advisor i is matched to site-term j, 0 if they are not
- $z_j \in \{0,1\}$, 0 if there is at least one experienced advisor at site j, 1 if there is not

Mathematical Model:

$$\begin{aligned}
\text{maximize} \quad & \sum_{i \in A_{FT}} \sum_{j \in S} x_{ij} + \lambda * (\sum_{j \in S} \sum_{i \in A} a_{ij} x_{ij}) - \mu * (\sum_{j \in S} z_j) && (1) \\
\text{subject to} \quad & \sum_{i \in A} x_{ij} \leq c_j, \ j \in S && (2) \\
& \sum_{j \in S} x_{ij} \geq 1, \ i \in A_{GEO} && (3) \\
& \sum_{i \in A_E} x_{ij} \geq 1 - z_j, \ j \in S && (4) \\
& \sum_{i \in A_\ell} x_{ij} \geq 1, \ j \in S_\ell && (5) \\
& \sum_{j \in S_A \cup S_B} x_{ij} \leq 1, \ i \in A && (6) \\
& \sum_{j \in S_B \cup S_C} x_{ij} \leq 1, \ i \in A && (7)
\end{aligned}$$

$$\sum_{j \in S_C \cup S_D} x_{ij} \leq 1, \ i \in A \tag{8}$$

$$\sum_{j \in S_E} x_{ij} \leq 1, \ i \notin A_{DP} \tag{9}$$

$$\sum_{j \in S_A} x_{ij} \leq 1, \ i \notin A_{DP} \tag{10}$$

$$\sum_{j \in S_E \cup S_A} x_{ij} = 0, \ i \in A_{DP} \tag{11}$$

$$x_{ij} = 1, \ i \in A_{DY}, \ j \in S_{DY,i} \tag{12}$$

$$\sum_{j \in S_{NI,i}} x_{ij} = 0, \ i \in A \tag{13}$$

$$x_{ij} \in \{0,1\}, \ i \in A, \ j \in S \tag{14}$$

$$z_j \in \{0,1\}, \ j \in S \tag{15}$$

Objective function (1) maximizes the number of matchings made while giving preference to matchings that increase the number of advisors matched to sites in which they are very interested and penalizing the number of site-terms without at least one experienced advisor. Constraint set (2) ensures that there are no more advisors per site-term than the capacity of that site-term. Constraint set (3) ensures that GEO faculty are assigned to advise for at least one site-term each year. Constraint set (4) works in conjunction with the z variables, which indicate whether the constraint is violated, to prioritize having at least one experienced advisor matched to each site-term. Constraint set (5) indicates a requirement that a minimum number of advisors speak language ℓ at the sites in subsets S_ℓ. Constraint sets (6) - (11) prevent an advisor from being assigned more than once in the same term and in back to back terms. Constraint set (12) ensures that project center directors who want to advise at their site-term are automatically matched there. Constraint set (13) prevents advisors from being matched to site-terms they ranked as not interested. Constraint sets (14) and (15) ensure that x_{ij} and z_j are binary variables.

The following section outlines how the abstract model (1) - (15) can be implemented in our system.

3.4 User Interaction Procedure

In this section we outline how we envision the user interacting with the optimization model and its output. In viewing this section, note that $\widehat{x}$ is the ($A * S$ dimensional) optimal solution that is generated by the solver and recommended to the user, $\widehat{x}^0$ is the initial optimal solution, $\widehat{x}^k$ is the optimal solution corresponding to re-solve k, and $\widehat{x}^*$ is the final match recommendation. We define $\widehat{x}^k_{(i,j)}$ as the match between advisor i and site-term j in iteration k.

Algorithm 1 User Interaction Procedure

```
input:      Mathematical model (1) - (15) $M$ with survey responses $S$
output:     Final matrix of match recommendation $\widehat{x}^{*}$ such that $M$ (1) is maximized
counter k = 0
$\widehat{x}^{k} \Leftarrow$ solve model (1) - (15)
while $\widehat{x}^{k}$ is not optimal for the user do
        for each pair $(i, j)$ do
                if $\widehat{x}^{k}_{(i,j)} = 1$ and it is not a desirable match
                        enter “N” in spreadsheet in cell corresponding to $(i, j)$
                if $\widehat{x}^{k}_{(i,j)} = 0$ and it is an unmatched desirable match
                        enter “Y” in spreadsheet in cell corresponding to $(i, j)$
        $k = k + 1$
        $\widehat{x}^{k} \Leftarrow$ re-solve model (1) - (15) with additional constraints
        $\widehat{x}^{*} \Leftarrow \widehat{x}^{k}$
return $\widehat{x}^{*}$
```

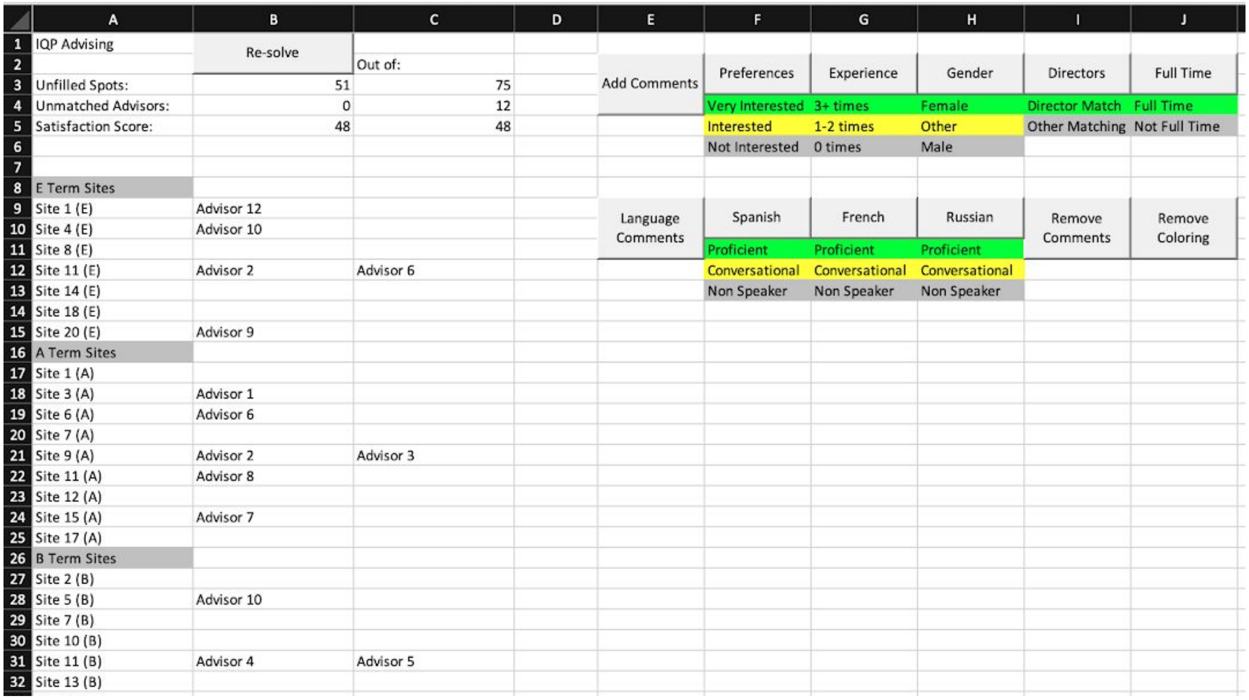
As outlined in this procedure, the user will begin by running the tool to solve the model. Then, if the user is not satisfied with the output, they can enter an “N” into any cell corresponding to a match they would like to prevent and enter a “Y” into any cell corresponding to a match they would like to see in the output. This is taken into the mathematical model by setting $\widehat{x}^{k}_{(i,j)} = 0$ or $\widehat{x}^{k}_{(i,j)} = 1$ for the corresponding cell (i, j) containing an “N” or a “Y”, respectively. Next, the user can re-solve the model and evaluate the new output. The user can repeat this process and re-solve the model as many times as they would like until they are satisfied with the output or the model becomes infeasible.

4. Results and Analysis

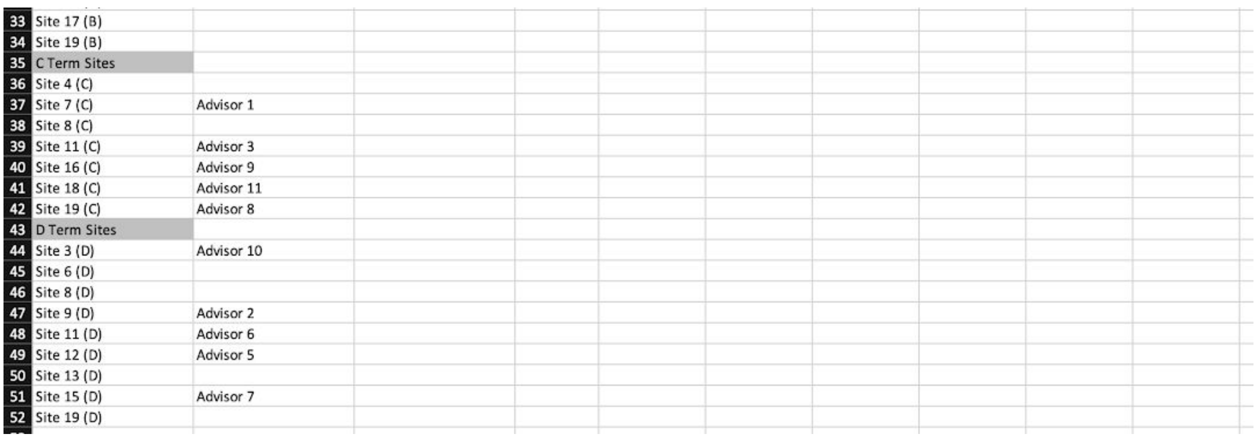
This section will discuss the results and analysis of our matching tool, in which we will describe the results of using sample data and the benefits of our tool.

4.1 Results of Sample Data

To test the functionality of our matching tool, we first created 20 representative sites, labeled Site 1, Site 2, …, Site 19, Site 20, and decided upon which terms each site would be offered as well as the desired number of advisors for each site-term combination. We then generated synthetic advisor data by taking the survey 12 times, labeling the advisors as Advisor 1, Advisor 2, …, Advisor 11, Advisor 12, and varying the responses each time. The results from running our tool with this data are shown below.



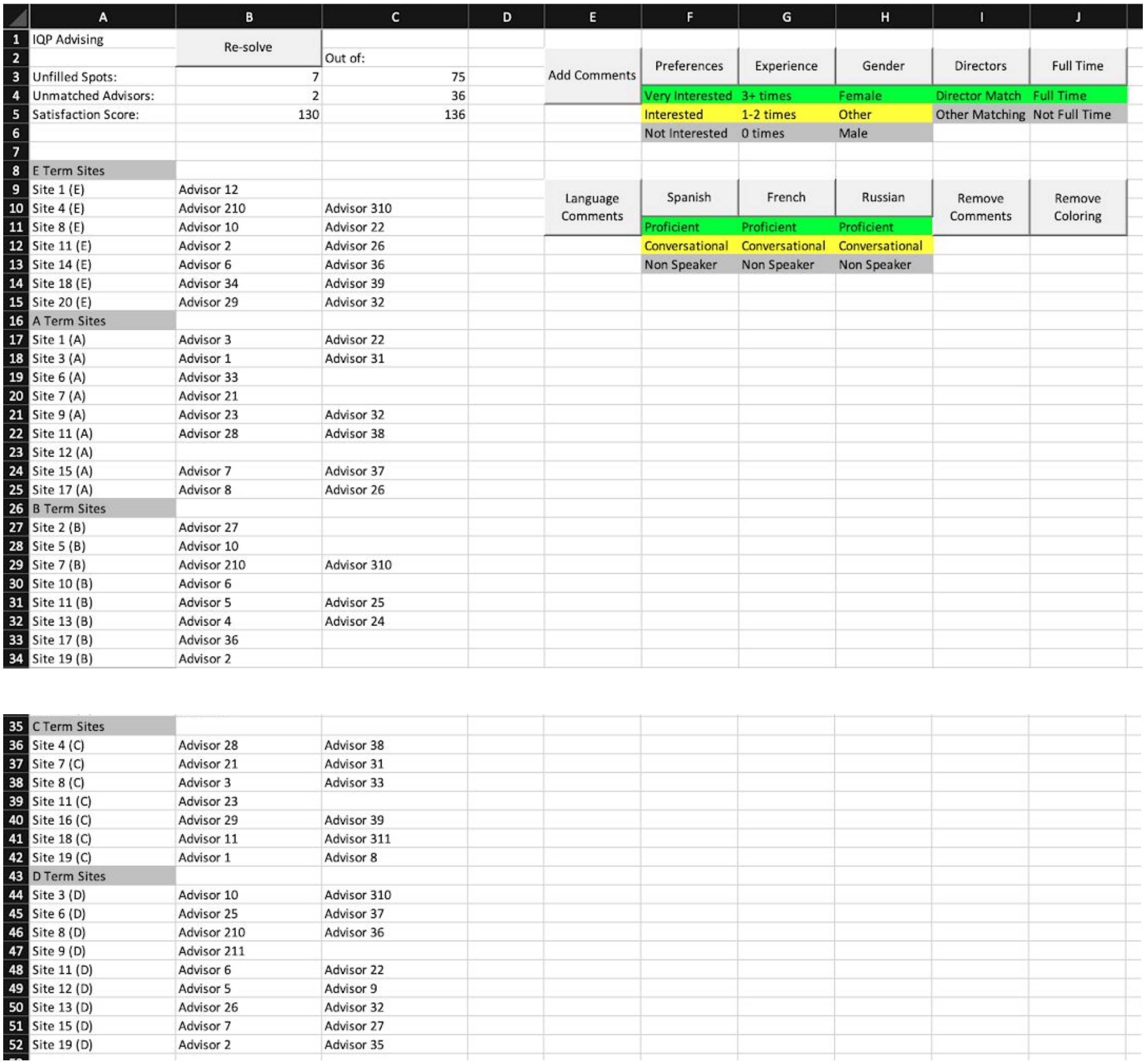
	A	B	C	D	E	F	G	H	I	J
1	IQP Advising	Re-solve								
2			Out of:		Add Comments	Preferences	Experience	Gender	Directors	Full Time
3	Unfilled Spots:	51	75							
4	Unmatched Advisors:	0	12			Very Interested	3+ times	Female	Director Match	Full Time
5	Satisfaction Score:	48	48			Interested	1-2 times	Other	Other Matching	Not Full Time
6						Not Interested	0 times	Male		
7										
8	E Term Sites									
9	Site 1 (E)	Advisor 12			Language Comments	Spanish	French	Russian	Remove Comments	Remove Coloring
10	Site 4 (E)	Advisor 10								
11	Site 8 (E)					Proficient	Proficient	Proficient		
12	Site 11 (E)	Advisor 2	Advisor 6			Conversational	Conversational	Conversational		
13	Site 14 (E)					Non Speaker	Non Speaker	Non Speaker		
14	Site 18 (E)									
15	Site 20 (E)	Advisor 9								
16	A Term Sites									
17	Site 1 (A)									
18	Site 3 (A)	Advisor 1								
19	Site 6 (A)	Advisor 6								
20	Site 7 (A)									
21	Site 9 (A)	Advisor 2	Advisor 3							
22	Site 11 (A)	Advisor 8								
23	Site 12 (A)									
24	Site 15 (A)	Advisor 7								
25	Site 17 (A)									
26	B Term Sites									
27	Site 2 (B)									
28	Site 5 (B)	Advisor 10								
29	Site 7 (B)									
30	Site 10 (B)									
31	Site 11 (B)	Advisor 4	Advisor 5							
32	Site 13 (B)									



33	Site 17 (B)	
34	Site 19 (B)	
35	C Term Sites	
36	Site 4 (C)	
37	Site 7 (C)	Advisor 1
38	Site 8 (C)	
39	Site 11 (C)	Advisor 3
40	Site 16 (C)	Advisor 9
41	Site 18 (C)	Advisor 11
42	Site 19 (C)	Advisor 8
43	D Term Sites	
44	Site 3 (D)	Advisor 10
45	Site 6 (D)	
46	Site 8 (D)	
47	Site 9 (D)	Advisor 2
48	Site 11 (D)	Advisor 6
49	Site 12 (D)	Advisor 5
50	Site 13 (D)	
51	Site 15 (D)	Advisor 7
52	Site 19 (D)	

Figure 2: Output from Sample Data with 12 Advisors

However, we noticed that these results were unsatisfying because there were so few advisors. Ideally we would have been able to simply take the survey over and over again to generate more data, but since the survey was fairly time consuming to complete when trying to create random and realistic responses we decided to duplicate the data we already had two times below our original data, and rename the advisors with a 2 or 3 in front of their original number (e.g. Advisor 6 in the original data becomes Advisor 26 in the first duplicate and Advisor 36 in the second duplicate). We also realized we needed to make sure none of the duplicate advisors were labeled as directors because then we would have three advisors trying to be matched to the same one site. The results of running the tool with this data with 36 advisors are shown below.



	A	B	C	D	E	F	G	H	I	J
1	IQP Advising	Re-solve								
2			Out of:			Preferences	Experience	Gender	Directors	Full Time
3	Unfilled Spots:	7	75		Add Comments					
4	Unmatched Advisors:	2	36			Very Interested	3+ times	Female	Director Match	Full Time
5	Satisfaction Score:	130	136			Interested	1-2 times	Other	Other Matching	Not Full Time
6						Not Interested	0 times	Male		
7										
8	E Term Sites									
9	Site 1 (E)	Advisor 12								
10	Site 4 (E)	Advisor 210	Advisor 310		Language Comments	Spanish	French	Russian	Remove Comments	Remove Coloring
11	Site 8 (E)	Advisor 10	Advisor 22			Proficient	Proficient	Proficient		
12	Site 11 (E)	Advisor 2	Advisor 26			Conversational	Conversational	Conversational		
13	Site 14 (E)	Advisor 6	Advisor 36			Non Speaker	Non Speaker	Non Speaker		
14	Site 18 (E)	Advisor 34	Advisor 39							
15	Site 20 (E)	Advisor 29	Advisor 32							
16	A Term Sites									
17	Site 1 (A)	Advisor 3	Advisor 22							
18	Site 3 (A)	Advisor 1	Advisor 31							
19	Site 6 (A)	Advisor 33								
20	Site 7 (A)	Advisor 21								
21	Site 9 (A)	Advisor 23	Advisor 32							
22	Site 11 (A)	Advisor 28	Advisor 38							
23	Site 12 (A)									
24	Site 15 (A)	Advisor 7	Advisor 37							
25	Site 17 (A)	Advisor 8	Advisor 26							
26	B Term Sites									
27	Site 2 (B)	Advisor 27								
28	Site 5 (B)	Advisor 10								
29	Site 7 (B)	Advisor 210	Advisor 310							
30	Site 10 (B)	Advisor 6								
31	Site 11 (B)	Advisor 5	Advisor 25							
32	Site 13 (B)	Advisor 4	Advisor 24							
33	Site 17 (B)	Advisor 36								
34	Site 19 (B)	Advisor 2								
35	C Term Sites									
36	Site 4 (C)	Advisor 28	Advisor 38							
37	Site 7 (C)	Advisor 21	Advisor 31							
38	Site 8 (C)	Advisor 3	Advisor 33							
39	Site 11 (C)	Advisor 23								
40	Site 16 (C)	Advisor 29	Advisor 39							
41	Site 18 (C)	Advisor 11	Advisor 311							
42	Site 19 (C)	Advisor 1	Advisor 8							
43	D Term Sites									
44	Site 3 (D)	Advisor 10	Advisor 310							
45	Site 6 (D)	Advisor 25	Advisor 37							
46	Site 8 (D)	Advisor 210	Advisor 36							
47	Site 9 (D)	Advisor 211								
48	Site 11 (D)	Advisor 6	Advisor 22							
49	Site 12 (D)	Advisor 5	Advisor 9							
50	Site 13 (D)	Advisor 26	Advisor 32							
51	Site 15 (D)	Advisor 7	Advisor 27							
52	Site 19 (D)	Advisor 2	Advisor 35							

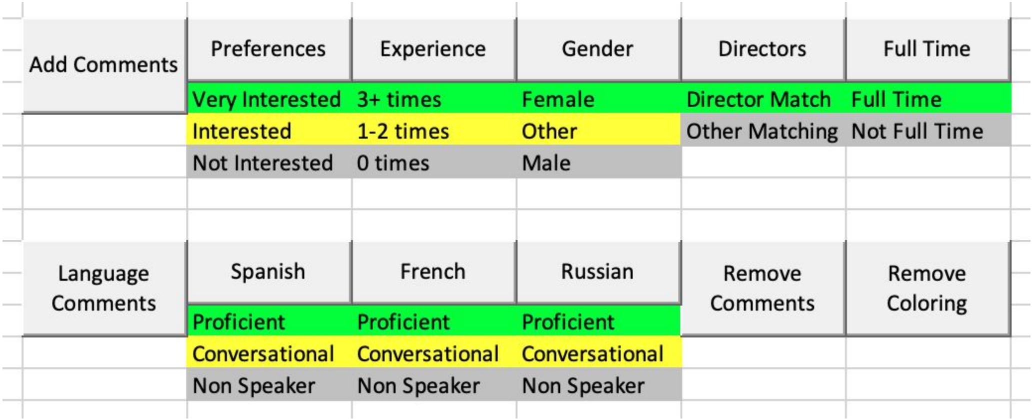
Figure 3: Output from Sample Data with 36 Advisors

4.2 Benefits of the Tool

Perhaps the largest benefit of the tool for the GEO stakeholders is the potential time savings. We found that with our synthetic data containing 40 site-term combinations, 75 advisor positions (roughly 2 per site-term combination), and 36 advisors, our tool was able to find an optimal solution satisfying all constraints in less than one and a half minutes. The solution generated by our tool was able to fill 68 of the 75 advisor positions, with only two advisors (Advisor 212 and Advisor 312) remaining unmatched. This was due to the fact that they were both duplicates of Advisor 12, who was only willing to advise at Site 1 in E term, where there is only one advisor position available. This means that Advisor 12 was matched to Site 1 in E term, and then since Advisor 212 and Advisor 313 were only willing to be matched there and there was no need for more advisors at that site, they were not able to be matched anywhere.

This solution also had a satisfaction score of 130 out of 136. The satisfaction score is calculated by finding the sum of the advisor-site rankings over all matches. Because 68 positions were filled, if all matched advisors went to a site in which they were very interested, each match would have a ranking of 2 resulting in a satisfaction score of 68 * 2 = 136. Our satisfaction score of 130 means that six advisors were matched to sites they were interested in, but not very interested in (recall that advisors cannot be matched to sites in which they are not interested).

We wanted to allow the users of our tool to not only see the recommended matchings, but to be able to interpret and interact with the generated solution. To facilitate the interpretation of the tool's output, we included buttons on the Recommended Matchings sheet that would add comments to the cells containing advisors names which appear when hovering over the cell, as well as buttons that color these cells based on various traits of the advisors. We incorporated two buttons that add comments, one to display and additional comments an applicant included in their survey, and one to display all of the foreign languages from the survey with the advisor's corresponding self-reported proficiency level. We also included eight buttons, shown below, that would color advisor cells based on the advisor's preference for being matched to the site they were matched to, experience level, gender, director status, employment status, and language abilities for the three most important languages as indicated by the GEO stakeholders. We also added a button that would remove the comments and a button that would remove the coloring of advisor cells so the users could review the output without those emphases.

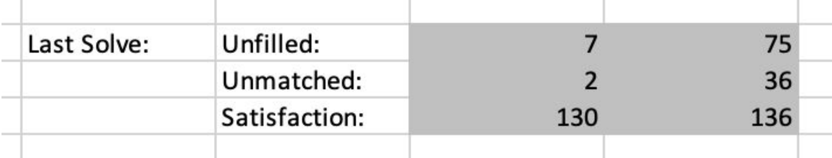


Add Comments	Preferences	Experience	Gender	Directors	Full Time
	Very Interested	3+ times	Female	Director Match	Full Time
	Interested	1-2 times	Other	Other Matching	Not Full Time
	Not Interested	0 times	Male		

Language Comments	Spanish	French	Russian	Remove Comments	Remove Coloring
	Proficient	Proficient	Proficient		
	Conversational	Conversational	Conversational		
	Non Speaker	Non Speaker	Non Speaker		

Figure 4: Available Buttons to Facilitate Interpretation of the Tool's Output

To allow the users to interact with the generated solution, we created another button that would run the tool again with the same baseline constraints and with additional conditions expressed by the users. The GEO stakeholders will be able to view many aspects of the recommended matchings and if they find a matching they would like to prevent, they can simply go to the position of that matching on the Site Matchings sheet and type an "N" into the cell representing the matching they want to disallow. Alternatively, if they discover a matching that the tool did not make but that they would like to see in the final recommendation, they can type a "Y" into the corresponding cell on the Site Matchings sheet to force the matching. This will allow the GEO staff to override an advisor's preference of "0" for a site, as they may have more information regarding the match. These additional conditions are then interpreted as constraints by the tool. The GEO stakeholders are able to make as many of these changes as they would like, and run the tool to re-solve the model repeatedly until they are satisfied with the final result.



Last Solve:	Unfilled:	7	75
	Unmatched:	2	36
	Satisfaction:	130	136

Figure 5: Display of Results from Last Solve

To help guide the GEO staff in making their final decision, we include a display of the results of the previous solution, a sample of which is shown above. We included this so they can compare the overall quality of the previous and current solution.

4.3 Reflection

After the survey had been sent out and completed by IQP advisor applicants, we asked the GEO stakeholders whether they had any feedback to give on the new application or whether they had received any feedback from applicants. They told us they were excited that this new format eliminated the need to go through each application to collect data manually. They also told us that they hadn’t received too much feedback from the applicants, which the GEO stakeholders were surprised about since they said the applicant group can be vocal when they have opinions. The only concerns they were presented with were from applicants who wanted to go back into the survey to change their answers, and a general concern from one or two people that the GEO stakeholders would be too reliant on the tool’s output. Luckily, we know this will not be the case as the GEO stakeholders have the ability to fully customize the tool’s output.

Department heads also indicated that they will need to be made aware of which of their professors will be advising an IQP, which will be easy to do after the matchings are made.

We also asked the GEO stakeholders if they had any feedback for us on the matching tool itself. They were all really pleased with what was developed, and Dean Rissmiller commented that it has many more features than he expected it to. The two graphs below illustrate the estimated time savings brought to the GEO stakeholders and the increased value of the new matches. Unfortunately, because of the complications of passing this tool between different GEO stakeholders while working remotely during the pandemic, they did not use the re-solve feature in their first year with this tool and instead altered the initial solution themselves. However, they were still able to see a significant reduction in the time this process takes and in the optimality of the final matches.

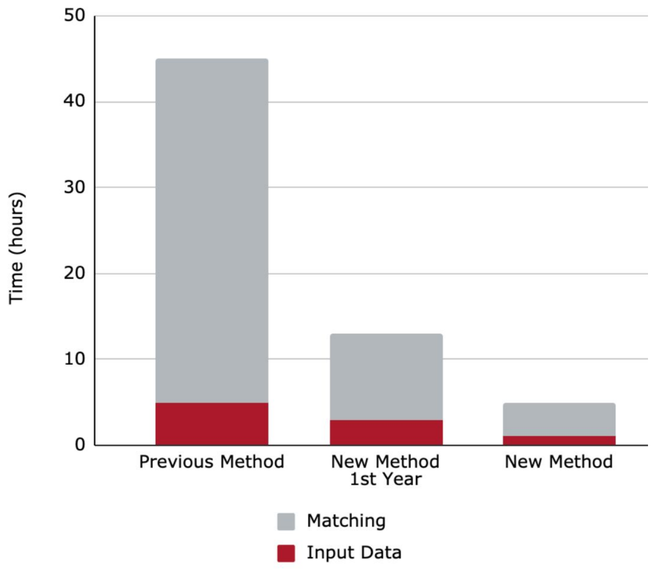


Figure 6: Estimated Time Savings for GEO Stakeholders

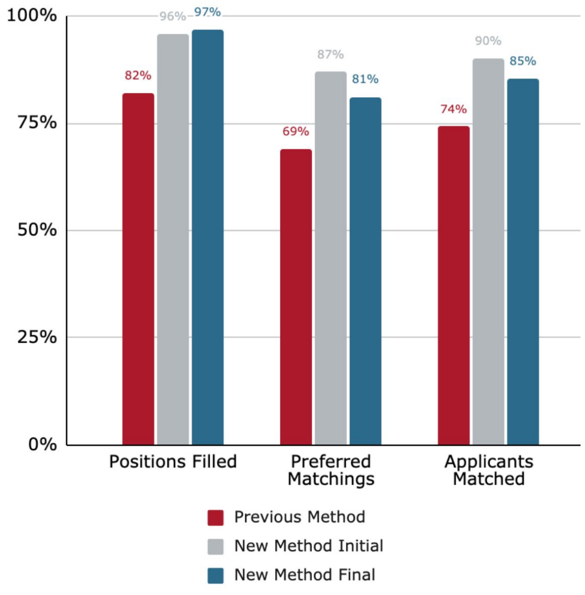


Figure 7: Increased Optimality of New Matches

In previous years, it was estimated that the stakeholders spent about 5 hours inputting advisor applicant data, and then 40+ hours on the matching process. In their first year of using the tool, including the time it took everyone to learn how to use it, they spent 3 hours inputting data and only 10 hours re-adjusting the matches after the tool's initial solve. The stakeholders anticipate that in future years, the entire process will take just 5 hours in total, which equates to at least an 89% reduction in time.

Looking at last year's data, we calculated that the GEO stakeholders were able to fill 82% of their advisor positions, with 69% of those matchings being to sites preferred by the applicants and only 74% of applicants being matched to any one or more sites. By using our tool this year, they were able to fill 97% of their positions, 15% more than when it was done manually. This is incredibly important to the stakeholders as they need to fill all remaining spots. The process to do so is long and difficult because the GEO stakeholders need to reach out to a series of potential advisors individually until all of the spots are filled. Additionally, 81% of the matches were to sites the advisors preferred to travel to, and 85% of those who applied were able to be assigned to at least one site.

5. Conclusions

We conclude this MQP with a summary and a discussion of contributions, lessons learned, limitations, and the potential for future research.

5.1 Summary

This MQP has improved upon the IQP advisor placement process for the GEO at WPI. More specifically, we helped those in the GEO responsible for making matches between IQP advisor applicants and IQP sites around the world. The GEO is responsible for filling about 100 advisor positions in over 50 IQP site-term combinations. Before this tool was developed, the GEO made these matches by hand. This manual process was laborious and took about 45 hours total, over the course of four to eight weeks, making it necessary for the GEO stakeholders to work over weekends and taking time away from their other obligations. They need to do this while taking into consideration many aspects such as advisor site preferences, experience levels, and foreign language abilities. To help them with this process, we designed a decision support system using VBA in Excel to create a semi-automated solution that takes into consideration the preferences and characteristics of advisors. The matching tool's outputted solution allows the GEO stakeholders to interact with the proposed matches to better fit their more nuanced preferences that cannot be easily built into the model. Each solve of the tool takes just a couple of minutes, and the GEO stakeholders have estimated that the entire process will take them about five hours. This tool is expected to have a significant impact on WPI's global project program by greatly reducing the amount of time the GEO stakeholders need to spend making these matches without decreasing their ability to control the results.

5.2 Contributions

By creating this matching tool for the GEO stakeholders, we addressed the challenge of matching advisors to project centers. This manual process was previously cumbersome and took at least 5 hours to collect data from all of the advisor applications, and then at least 40 hours to make the advisor matchings. However, given that the GEO stakeholders have many other responsibilities, this was not all done at once. The matching process typically began in mid-November and would conclude anytime from mid-December to mid-January, meaning matchings would not be determined until four to eight weeks after they were started. Instead, we have automated some of the more challenging parts of this process, allowing the GEO stakeholders to download application responses from Qualtrics and put them directly into the matching tool. Then, they can simply run the tool we created to determine initial matches in a couple of minutes.

Not only does our tool output a solution to the matching problem, but optimization methods provide the “best outcome” with respect to the given constraints, as determined mathematically by OpenSolver. Additionally, the presented solution is not just displayed, but is able to be interpreted using the coloring and comments buttons, as well as interacted with using the Y/N option as described in section 4.2. Given that the re-solve requires less computations than the initial solve, this takes just under two minutes. The GEO stakeholders estimate that this new matching process including inputting application responses, making initial matchings, and re-solving until they are satisfied with the results will take them about 5 hours in total, which is far less time than the 45+ hours it has taken in previous years.

Overall, this tool will provide the GEO stakeholders with a better and less time-intensive matching process for their IQP advisors. With this time saved, Dean Rissmiller will be able to put more of his focus into his other obligations and responsibilities. He also hopes to gain some free time during his weekends to not spend on work. This will also contribute to WPI by further improving their already prestigious Global Projects Program. In making matches faster, the tool will help department heads in their course planning by allowing them to know sooner which terms their instructors will be travelling and therefore unavailable to teach courses. Additionally, it will benefit individual faculty by enabling them to plan ahead for traveling to the site they were matched to much sooner, and alleviating the concern of handling travel logistics for unmatched sites.

More broadly, this matching tool also provides contributions to operations research. Through our use of a multi-tiered objective function, which takes into consideration multiple aspects on which to optimize, we are able to categorize our model as a multi-objective programming problem. This is valuable because there remains many things to explore in this subset of optimization models. Therefore, creating a tool that combines multi-objective programming with interactive outputs in a real-world application benefits operations research as a whole by developing depth and further applications within the field.

5.3 Lessons Learned

The process of designing and creating this matching tool was an invaluable learning experience. We were able to utilize VBA to develop our knowledge of the programming language as well as binary integer optimization techniques applied to a real world problem. This process allowed us to expand upon our critical thinking and decision making skills. As we grew to understand more about the needs of the GEO stakeholders, we developed an understanding of how to tailor designs towards a specific goal as well as the importance of an interactive interface to assist in the decision making process.

Conducting this project during the coronavirus pandemic forced us to also adapt to the challenges of working entirely virtually. While challenging, we managed to have all of our meetings with the GEO stakeholders in a virtual format throughout the course of working with them. Our communication was done entirely through email and video calls, with our meetings being held over Zoom. These meetings presented their own challenges, as we needed to show the tool solely through screen sharing, which lessened our ability to observe the reactions of our stakeholders. Additionally, discussions were hindered by the delay between when someone began talking and when the members of the call began to hear them, as multiple people would begin talking at once fairly frequently. Since all interactions in the WPI community were over Zoom or other virtual platforms at that time, we were all able to adjust quickly and still have successful meetings.

5.4 Limitations

One of the largest limitations in this project was working around external time constraints. The GEO stakeholders had limited availability which, when combined with the challenge of coordinating the schedules of the six attendees, presented a lot of difficulty in finding times to hold these meetings. This was a product of the really challenging times as well as the fact that many of us were trying to juggle lots of commitments in pressured times. I also faced challenges during the pandemic, so I understand that these difficulties are often unavoidable. The GEO stakeholders also had strict deadlines of when they needed particular aspects of the survey and the matching tool to be completed by, in order to use them in the coming year's IQP advisor matching process. Additionally, this project work was done in conjunction with classes, so there was difficulty in balancing project work with coursework.

Working in a one-person group also limited our capacity to discuss key concepts and ideas for further development of the project. Thankfully, this was significantly lessened by the ability to collaborate with the project's advisor, Professor Trapp, who provided an educated and insightful perspective on all aspects of the project.

5.5 Future Research Directions

In one of our meetings with the GEO stakeholders, they told us that if we have extra time it would be nice for the tool to make or partially make ID 2050 instructor assignments at the same time. They told us that this is not a priority, but it would be helpful to them if we could easily work it into our tool. This would require the tool to either be able to read a list of faculty qualified to teach the course, or to prompt applicants for their home department when they are filling out the form. If the tool collected the departments of applicants, it would then only assign the GEO faculty to teach ID 2050, as only the GEO faculty and a select few other people are able to properly instruct the course. As this did not happen in this MQP, this could be one area in which future research could improve upon our tool.

Another aspect that could be explored in the future is asking the tool to match two specific advisors together. As we were wrapping up this MQP, this idea was brought to us as an idea of a new constraint that would bring value to the GEO stakeholders. This is an additional request some applicants have, to be able to advise an IQP with a particular advisor but not necessarily at one particular site. It would be very feasible for someone to work this preference into the tool in the future, if further work is done on it.

The matching tool we designed should be explored and developed with respect to its potential application to solving other similar problems. Many universities have advisor placement problems. For example, schools with travel abroad programs that need to assign advisors to each site they send students to could utilize this tool with some minor adjustments to make their advisor matchings. With some more significant modifications to this tool, it could also be used to assign Professors to teach different courses, or to assign different courses to different classrooms and lecture halls. The goal and constraints in these applications would be different, but our model is powerful because its basic structure can be applied to a broad range of uses.

Outside of the academic setting, our tool could also become useful for companies and organizations. The focus of our model is finding the best placement of assets subject to limited resources, which is a common problem in industry. Of course, in order to apply our tool in other settings with different constraints and objectives, it will need to be adjusted significantly to fit the new problem.

Bibliography

"Bernard M. Gordon Prize for Innovation in Engineering and Technology Education." *National Academy of Engineering*, www.nae.edu/20685/GordonPrize#tabs.

Carroll, Courtney, Meredith Juers, and Sonja Kent. "Improving the Scheduling of Operating Rooms at UMass Memorial Medical Center." *Worcester Polytechnic Institute*, 2014, https://web.wpi.edu/Pubs/E-project/Available/E-project-030614-144337/unrestricted/Finl_OR_Paper.pdf.

Dias, Camila, Lin Jiang, and Elizabeth Karpinski. "Improving Student Placement in IQP Centers via Preference Matching." *Worcester Polytechnic Institute*, 2017, https://web.wpi.edu/Pubs/E-project/Available/E-project-032417-112059/unrestricted/Imroving_Student_Placement_in_IQP_Centers_via_Preference_Matching.pdf.

Hanna, Michelle M. "Principles of Designing and Developing Spreadsheet-Based Decision Support Systems." *University of Florida*, 2004.

"Interdisciplinary & Global Studies." *Worcester Polytechnic Institute*, www.wpi.edu/academics/departments/interdisciplinary-global-studies.

Nguyen, Uyen Ngoc Thao. "Finding Improving Solutions that Control Disruption to Binary Optimization Problems." *Worcester Polytechnic Institute*, 2014, https://web.wpi.edu/Pubs/E-project/Available/E-project-031114-153750/unrestricted/Uyn_Nguyen_MQP_Final_Report.pdf.

Power, Daniel J., and Ramesh Sharda. "Model-Driven Decision Support Systems: Concepts and Research Directions." *Decision Support Systems*, vol. 43, no. 3, 2007, pp. 1044–1061.

Science Direct, doi:10.1016/j.dss.2005.05.030.

Quinn-Szcesuil, Julia. “Global Projects for All.” *Worcester Polytechnic Institute*, 21 Aug. 2017, www.wpi.edu/news/global-projects-all.

Roth, Alvin E. “Marketplaces, Markets, and Market Design.” *American Economic Review*, vol. 108, no. 7, 2018, pp. 1609–1658., doi:10.1257/aer.108.7.1609.

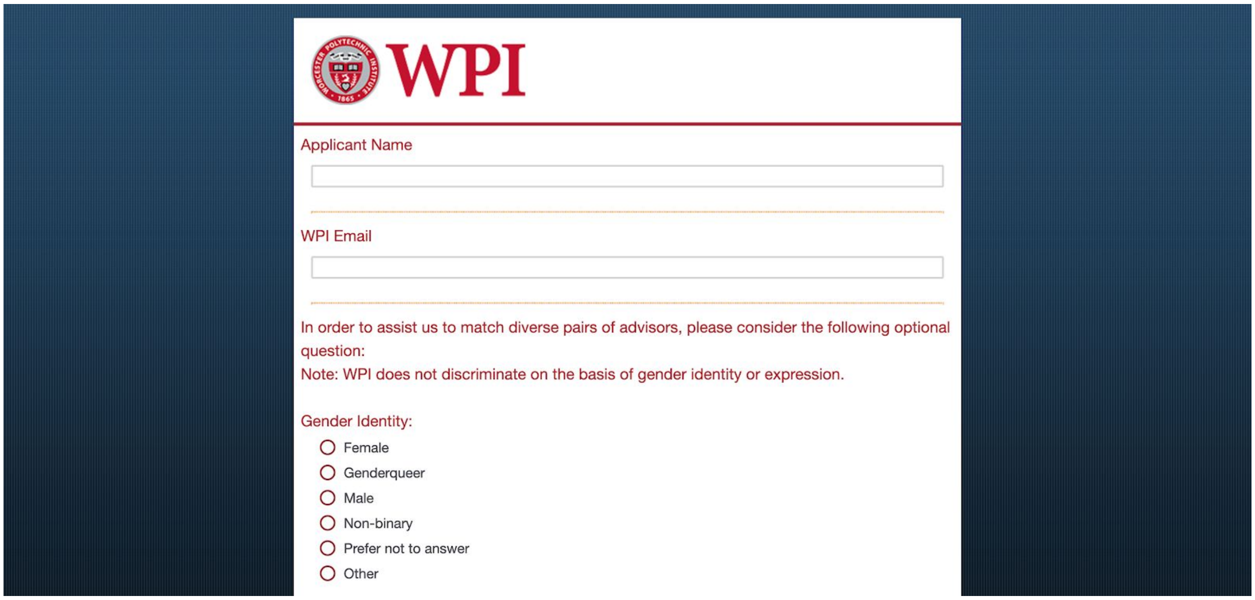
Shibl, Rania, Meredith Lawley, and Justin Debuse. "Factors influencing decision support system acceptance." *Decision Support Systems*, vol. 54, no. 2, 2013, pp. 953-961.

Sunde-Brown, Alexander, and Ethan M. Moon. “Pre-Mission Flight Plan Optimization.” *Worcester Polytechnic Institute*, 2015, https://web.wpi.edu/Pubs/E-project/Available/E-project-032715-144626/unrestricted/ECDR.pdf.

Yang, Mei, Shah Nazir, Qingshan Xu, and Shaukat Ali. "Deep Learning Algorithms and Multicriteria Decision-Making Used in Big Data: A Systematic Literature Review." *Complexity*, vol. 2020, 2020, https://doi.org/10.1155/2020/2836064.

Yang, Xin-She. *Introduction to Mathematical Optimization: from Linear Programming to Metaheuristics*. Cambridge International Science Publishing, 2008.

Appendix A: Advisor’s Survey



WPI

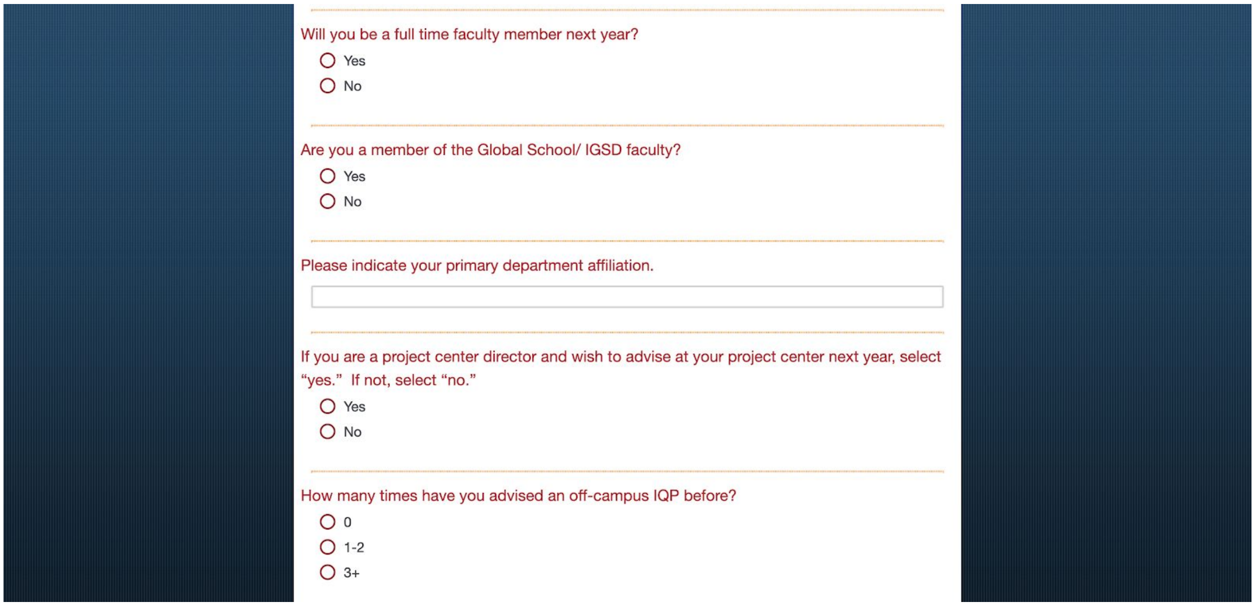
Applicant Name

WPI Email

In order to assist us to match diverse pairs of advisors, please consider the following optional question:
Note: WPI does not discriminate on the basis of gender identity or expression.

Gender Identity:

○ Female
○ Genderqueer
○ Male
○ Non-binary
○ Prefer not to answer
○ Other



Will you be a full time faculty member next year?

○ Yes
○ No

Are you a member of the Global School/ IGSD faculty?

○ Yes
○ No

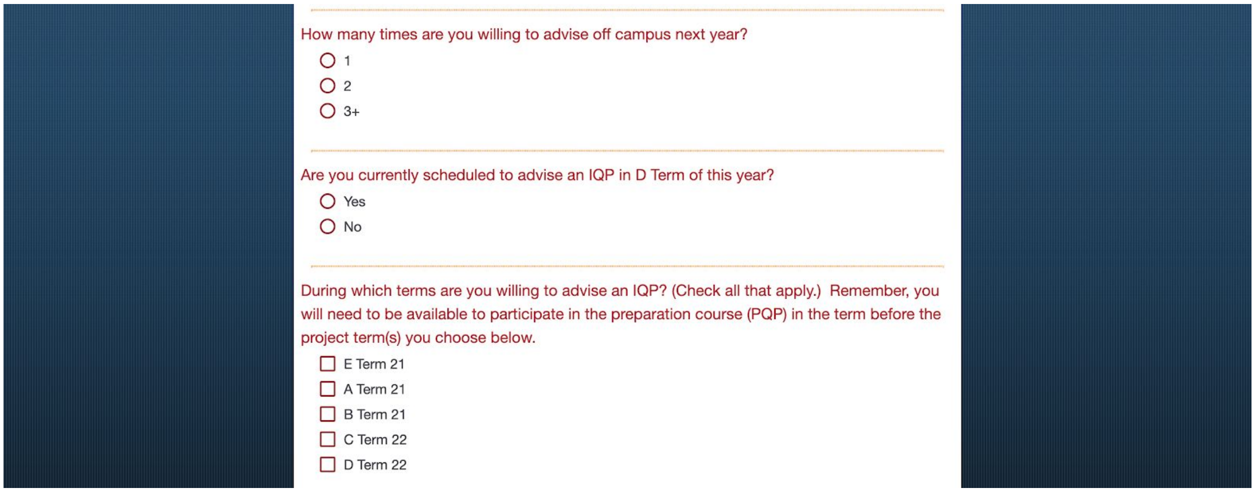
Please indicate your primary department affiliation.

If you are a project center director and wish to advise at your project center next year, select “yes.” If not, select “no.”

○ Yes
○ No

How many times have you advised an off-campus IQP before?

○ 0
○ 1-2
○ 3+



How many times are you willing to advise off campus next year?

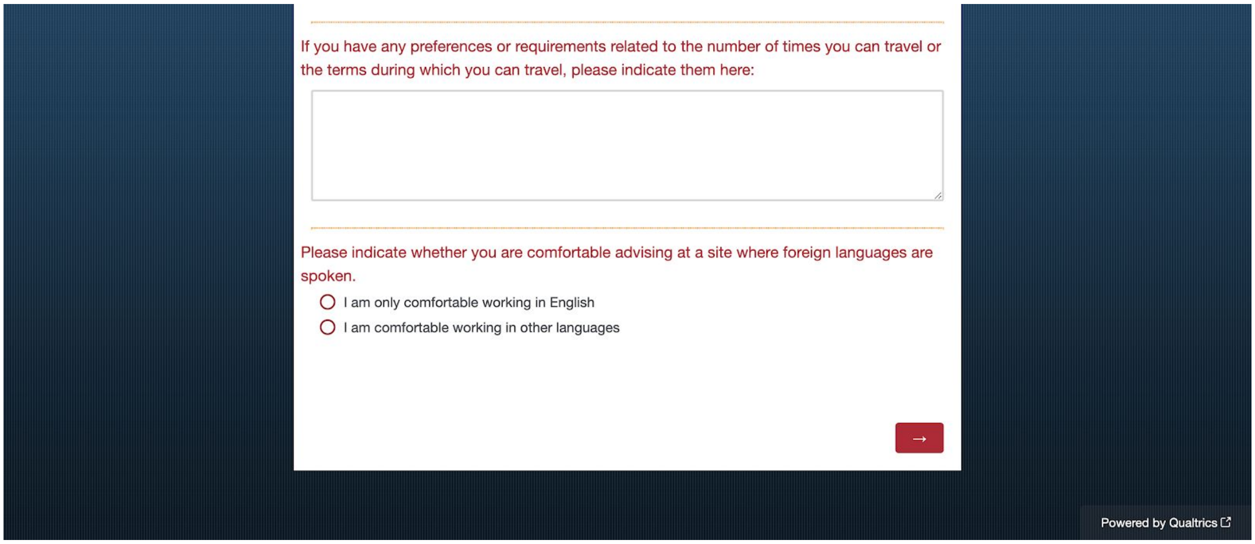
○ 1
○ 2
○ 3+

Are you currently scheduled to advise an IQP in D Term of this year?

○ Yes
○ No

During which terms are you willing to advise an IQP? (Check all that apply.) Remember, you will need to be available to participate in the preparation course (PQP) in the term before the project term(s) you choose below.

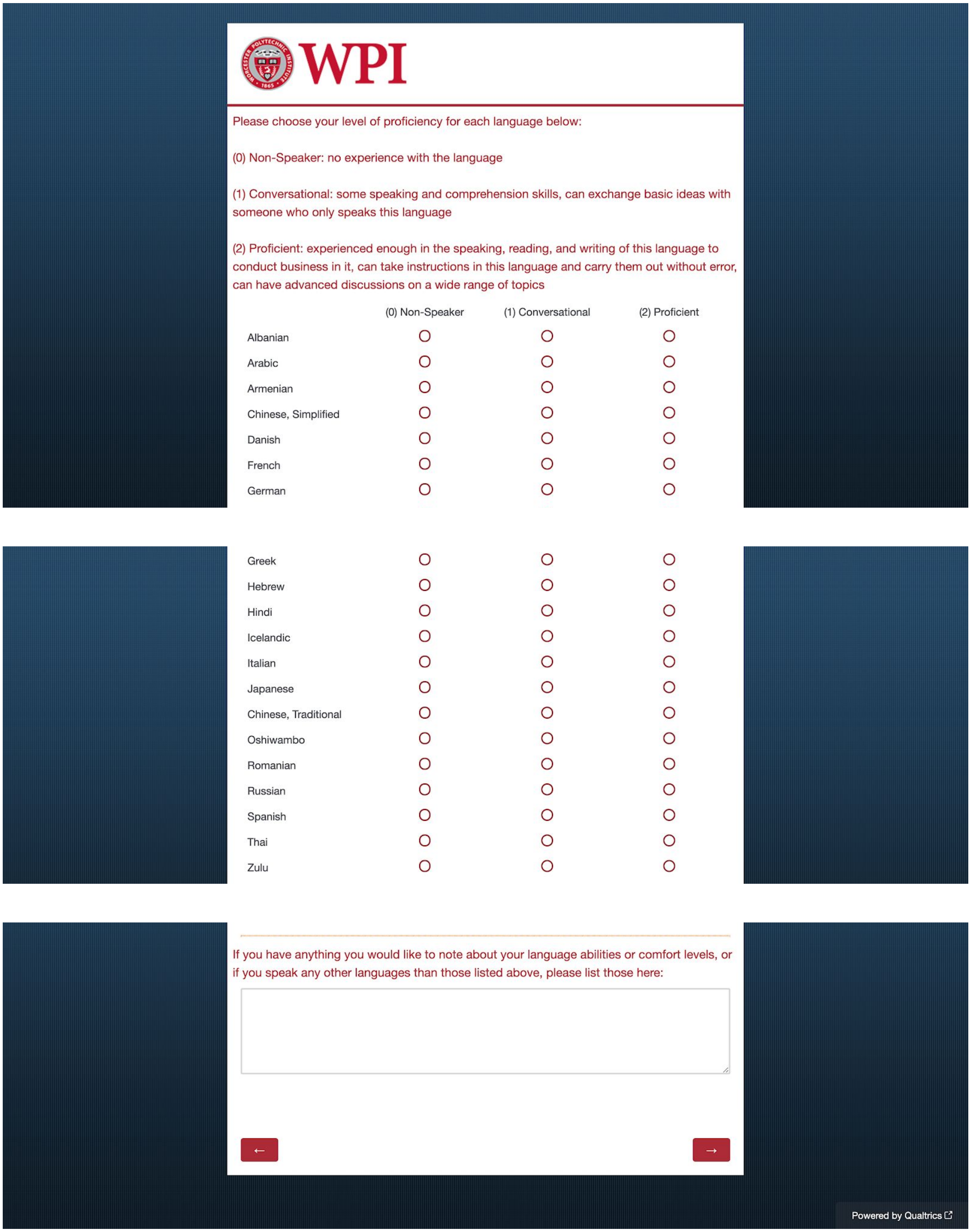
□ E Term 21
□ A Term 21
□ B Term 21
□ C Term 22
□ D Term 22



If you have any preferences or requirements related to the number of times you can travel or the terms during which you can travel, please indicate them here:

Please indicate whether you are comfortable advising at a site where foreign languages are spoken.

○ I am only comfortable working in English
○ I am comfortable working in other languages



WPI

Please choose your level of proficiency for each language below:

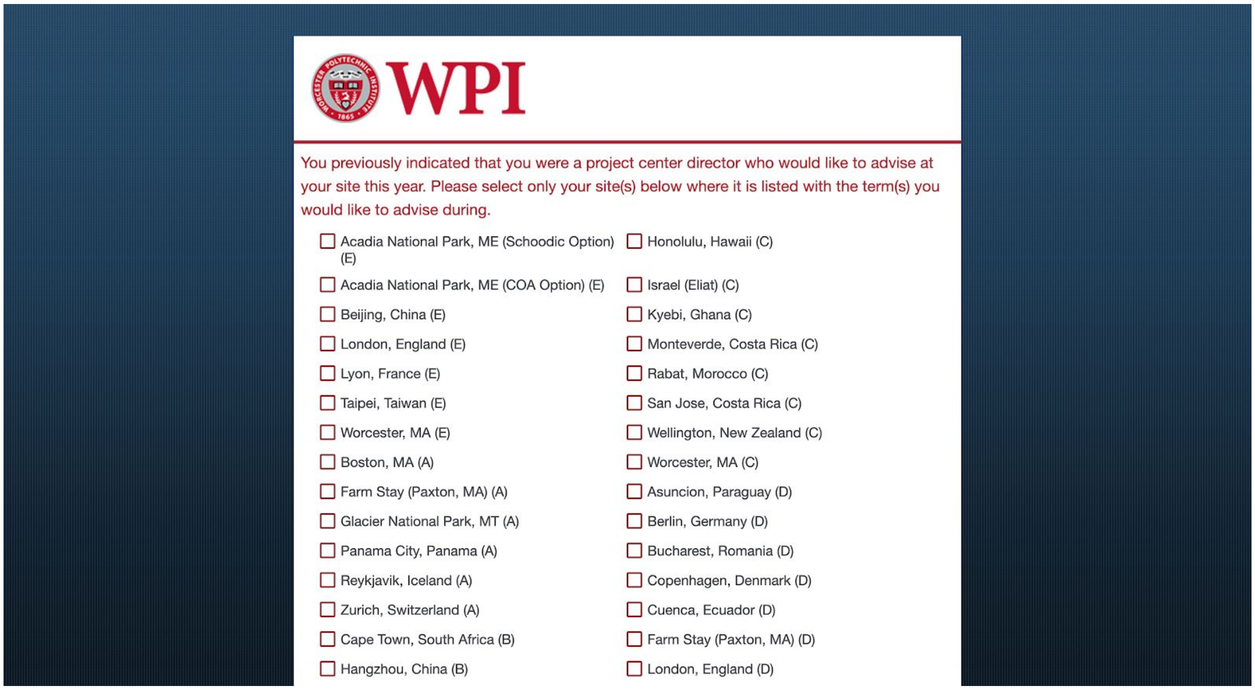
(0) Non-Speaker: no experience with the language

(1) Conversational: some speaking and comprehension skills, can exchange basic ideas with someone who only speaks this language

(2) Proficient: experienced enough in the speaking, reading, and writing of this language to conduct business in it, can take instructions in this language and carry them out without error, can have advanced discussions on a wide range of topics

	(0) Non-Speaker	(1) Conversational	(2) Proficient
Albanian	○	○	○
Arabic	○	○	○
Armenian	○	○	○
Chinese, Simplified	○	○	○
Danish	○	○	○
French	○	○	○
German	○	○	○
Greek	○	○	○
Hebrew	○	○	○
Hindi	○	○	○
Icelandic	○	○	○
Italian	○	○	○
Japanese	○	○	○
Chinese, Traditional	○	○	○
Oshiwambo	○	○	○
Romanian	○	○	○
Russian	○	○	○
Spanish	○	○	○
Thai	○	○	○
Zulu	○	○	○

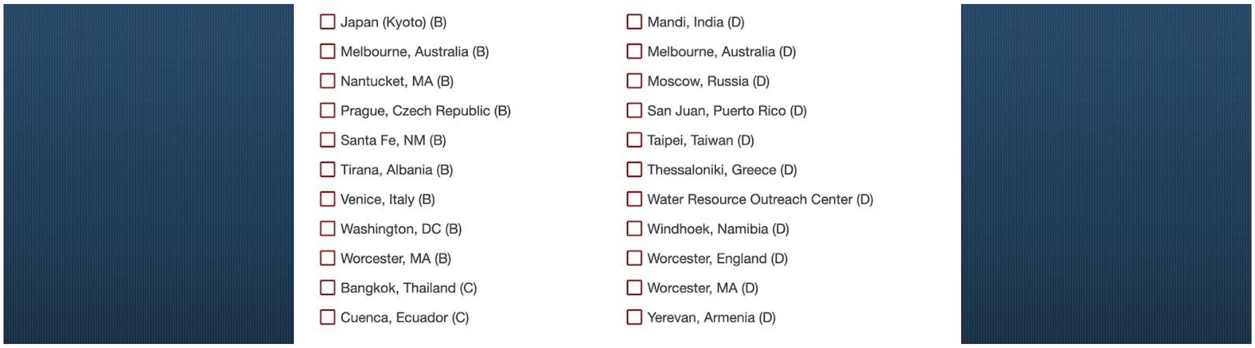
If you have anything you would like to note about your language abilities or comfort levels, or if you speak any other languages than those listed above, please list those here:



You previously indicated that you were a project center director who would like to advise at your site this year. Please select only your site(s) below where it is listed with the term(s) you would like to advise during.

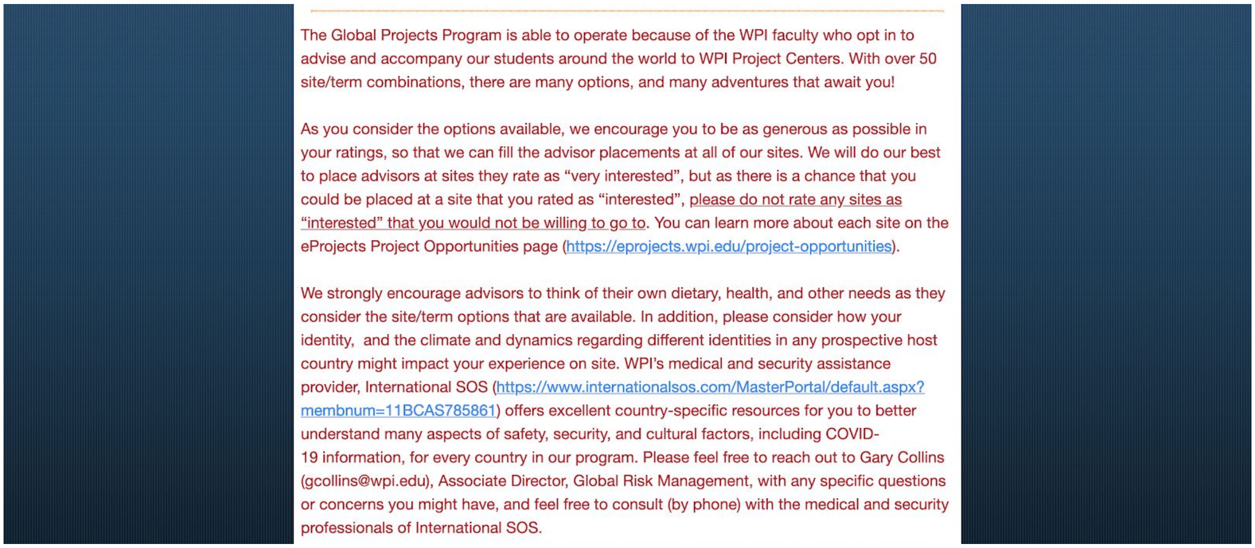
☐ Acadia National Park, ME (Schoodic Option) (E)
☐ Acadia National Park, ME (COA Option) (E)
☐ Beijing, China (E)
☐ London, England (E)
☐ Lyon, France (E)
☐ Taipei, Taiwan (E)
☐ Worcester, MA (E)
☐ Boston, MA (A)
☐ Farm Stay (Paxton, MA) (A)
☐ Glacier National Park, MT (A)
☐ Panama City, Panama (A)
☐ Reykjavik, Iceland (A)
☐ Zurich, Switzerland (A)
☐ Cape Town, South Africa (B)
☐ Hangzhou, China (B)

☐ Honolulu, Hawaii (C)
☐ Israel (Eliat) (C)
☐ Kyebi, Ghana (C)
☐ Monteverde, Costa Rica (C)
☐ Rabat, Morocco (C)
☐ San Jose, Costa Rica (C)
☐ Wellington, New Zealand (C)
☐ Worcester, MA (C)
☐ Asuncion, Paraguay (D)
☐ Berlin, Germany (D)
☐ Bucharest, Romania (D)
☐ Copenhagen, Denmark (D)
☐ Cuenca, Ecuador (D)
☐ Farm Stay (Paxton, MA) (D)
☐ London, England (D)



☐ Japan (Kyoto) (B)
☐ Melbourne, Australia (B)
☐ Nantucket, MA (B)
☐ Prague, Czech Republic (B)
☐ Santa Fe, NM (B)
☐ Tirana, Albania (B)
☐ Venice, Italy (B)
☐ Washington, DC (B)
☐ Worcester, MA (B)
☐ Bangkok, Thailand (C)
☐ Cuenca, Ecuador (C)

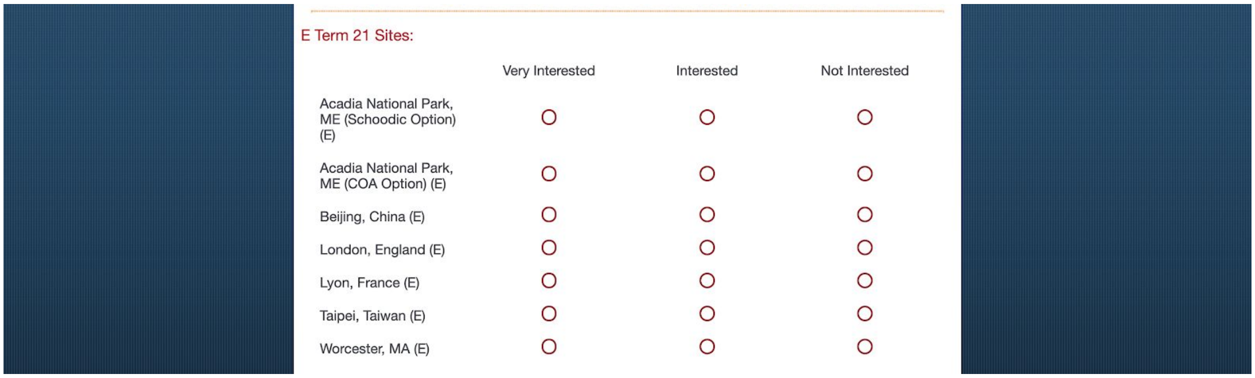
☐ Mandi, India (D)
☐ Melbourne, Australia (D)
☐ Moscow, Russia (D)
☐ San Juan, Puerto Rico (D)
☐ Taipei, Taiwan (D)
☐ Thessaloniki, Greece (D)
☐ Water Resource Outreach Center (D)
☐ Windhoek, Namibia (D)
☐ Worcester, England (D)
☐ Worcester, MA (D)
☐ Yerevan, Armenia (D)



The Global Projects Program is able to operate because of the WPI faculty who opt in to advise and accompany our students around the world to WPI Project Centers. With over 50 site/term combinations, there are many options, and many adventures that await you!

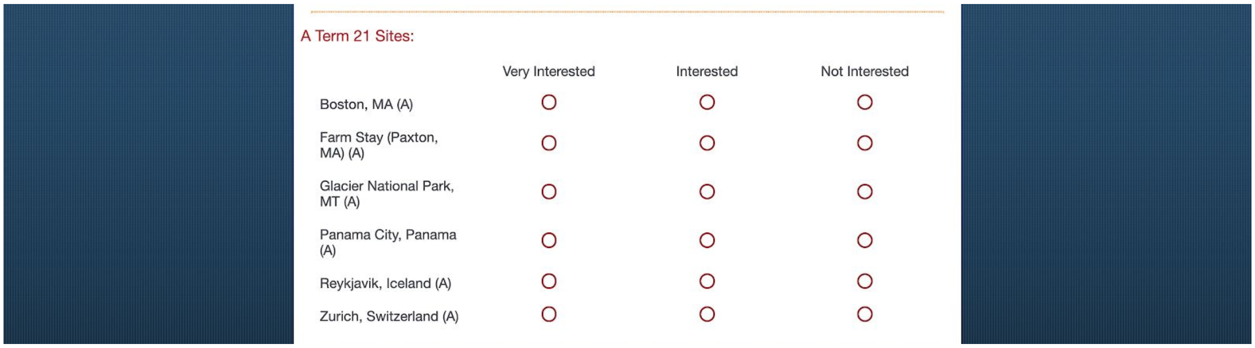
As you consider the options available, we encourage you to be as generous as possible in your ratings, so that we can fill the advisor placements at all of our sites. We will do our best to place advisors at sites they rate as "very interested", but as there is a chance that you could be placed at a site that you rated as "interested", please do not rate any sites as "interested" that you would not be willing to go to. You can learn more about each site on the eProjects Project Opportunities page (https://eprojects.wpi.edu/project-opportunities).

We strongly encourage advisors to think of their own dietary, health, and other needs as they consider the site/term options that are available. In addition, please consider how your identity, and the climate and dynamics regarding different identities in any prospective host country might impact your experience on site. WPI's medical and security assistance provider, International SOS (https://www.internationalsos.com/MasterPortal/default.aspx?membnum=11BCAS785861) offers excellent country-specific resources for you to better understand many aspects of safety, security, and cultural factors, including COVID-19 information, for every country in our program. Please feel free to reach out to Gary Collins (gcollins@wpi.edu), Associate Director, Global Risk Management, with any specific questions or concerns you might have, and feel free to consult (by phone) with the medical and security professionals of International SOS.



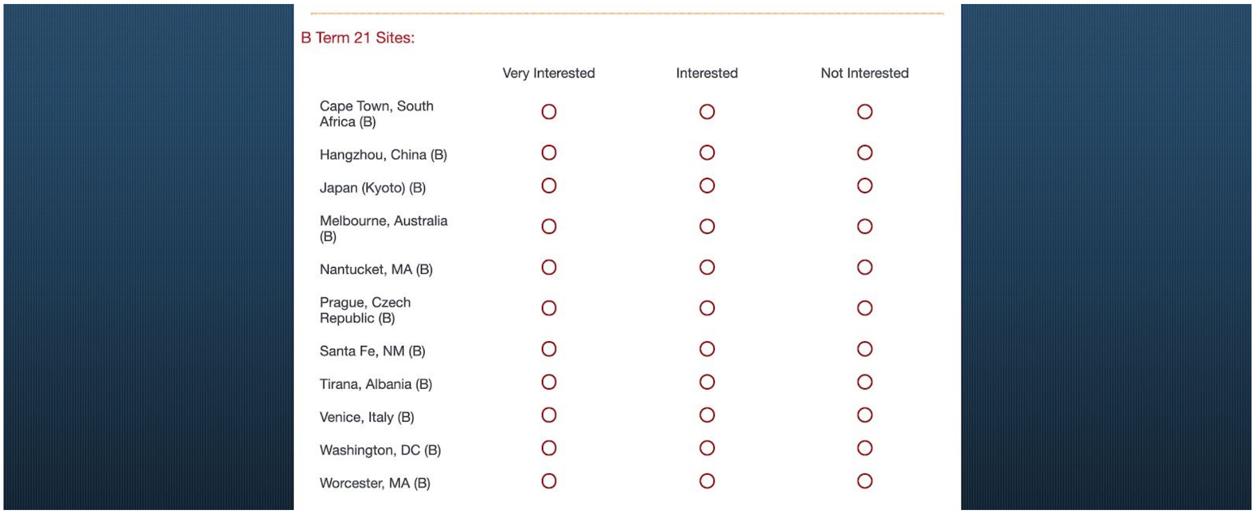
E Term 21 Sites:

	Very Interested	Interested	Not Interested
Acadia National Park, ME (Schoodic Option) (E)	○	○	○
Acadia National Park, ME (COA Option) (E)	○	○	○
Beijing, China (E)	○	○	○
London, England (E)	○	○	○
Lyon, France (E)	○	○	○
Taipei, Taiwan (E)	○	○	○
Worcester, MA (E)	○	○	○



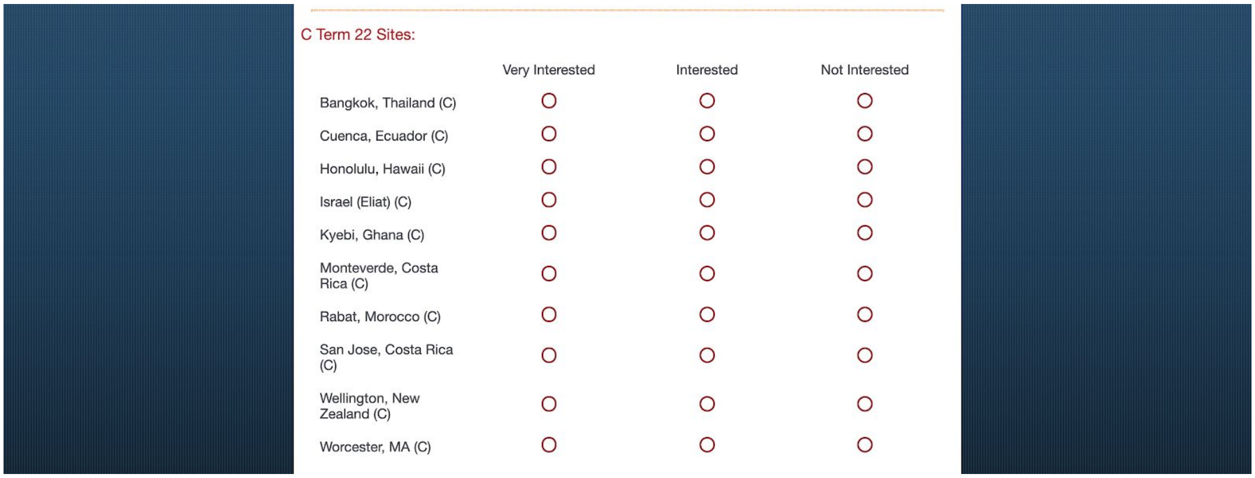
A Term 21 Sites:

	Very Interested	Interested	Not Interested
Boston, MA (A)	○	○	○
Farm Stay (Paxton, MA) (A)	○	○	○
Glacier National Park, MT (A)	○	○	○
Panama City, Panama (A)	○	○	○
Reykjavik, Iceland (A)	○	○	○
Zurich, Switzerland (A)	○	○	○



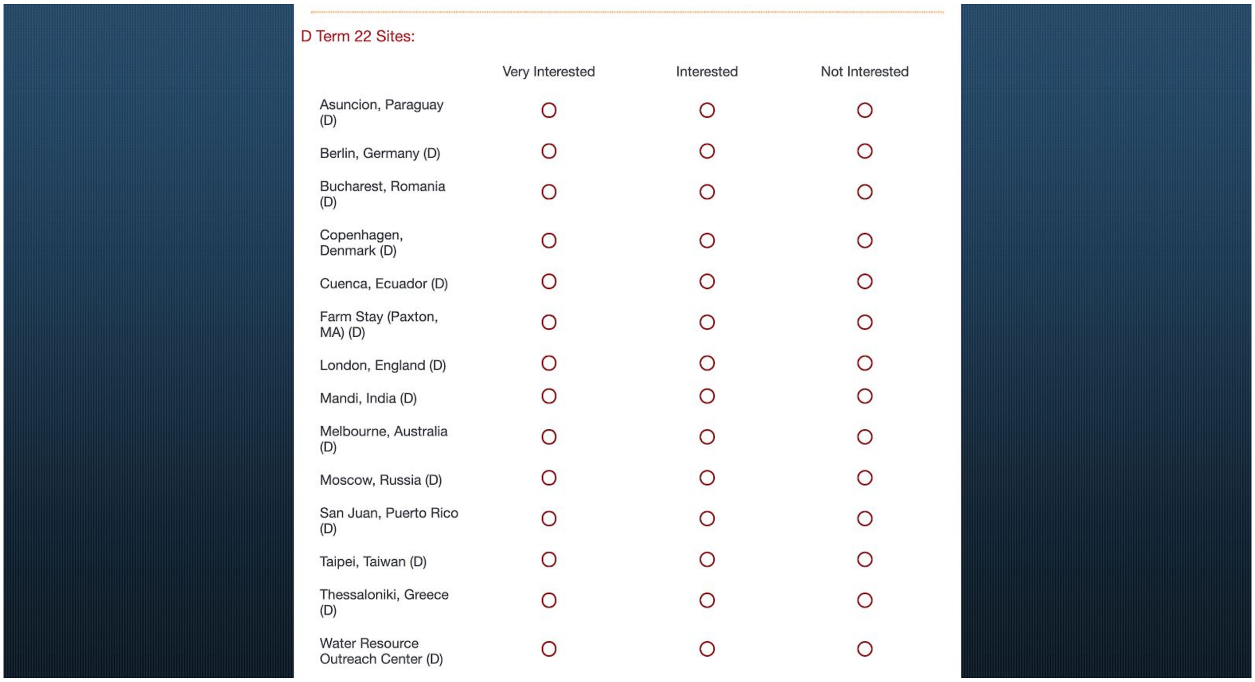
B Term 21 Sites:

	Very Interested	Interested	Not Interested
Cape Town, South Africa (B)	○	○	○
Hangzhou, China (B)	○	○	○
Japan (Kyoto) (B)	○	○	○
Melbourne, Australia (B)	○	○	○
Nantucket, MA (B)	○	○	○
Prague, Czech Republic (B)	○	○	○
Santa Fe, NM (B)	○	○	○
Tirana, Albania (B)	○	○	○
Venice, Italy (B)	○	○	○
Washington, DC (B)	○	○	○
Worcester, MA (B)	○	○	○



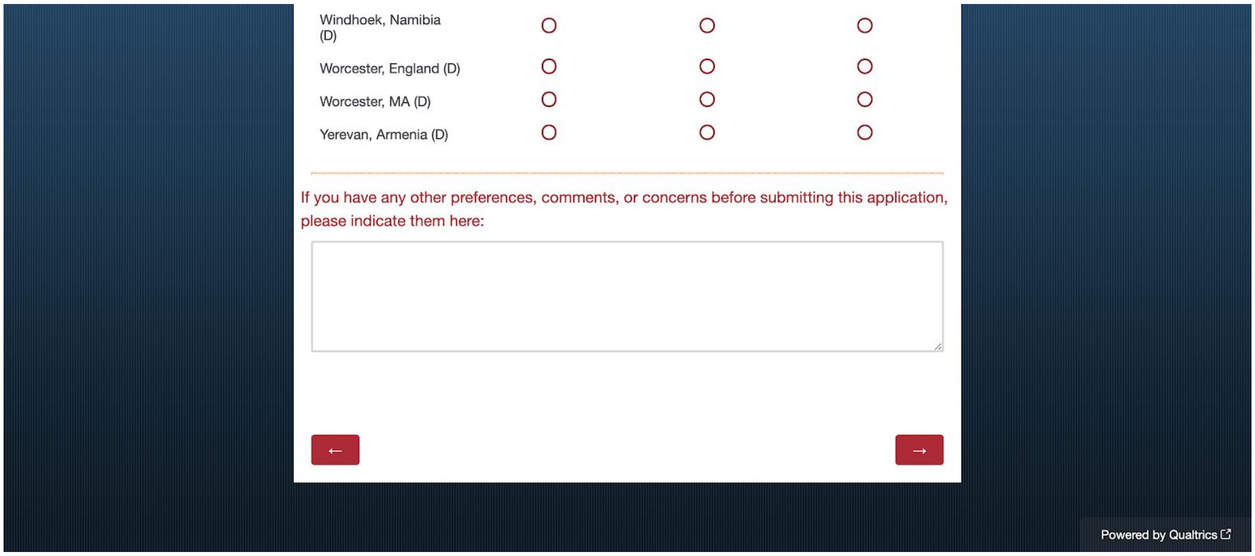
C Term 22 Sites:

	Very Interested	Interested	Not Interested
Bangkok, Thailand (C)	○	○	○
Cuenca, Ecuador (C)	○	○	○
Honolulu, Hawaii (C)	○	○	○
Israel (Eliat) (C)	○	○	○
Kyebi, Ghana (C)	○	○	○
Monteverde, Costa Rica (C)	○	○	○
Rabat, Morocco (C)	○	○	○
San Jose, Costa Rica (C)	○	○	○
Wellington, New Zealand (C)	○	○	○
Worcester, MA (C)	○	○	○



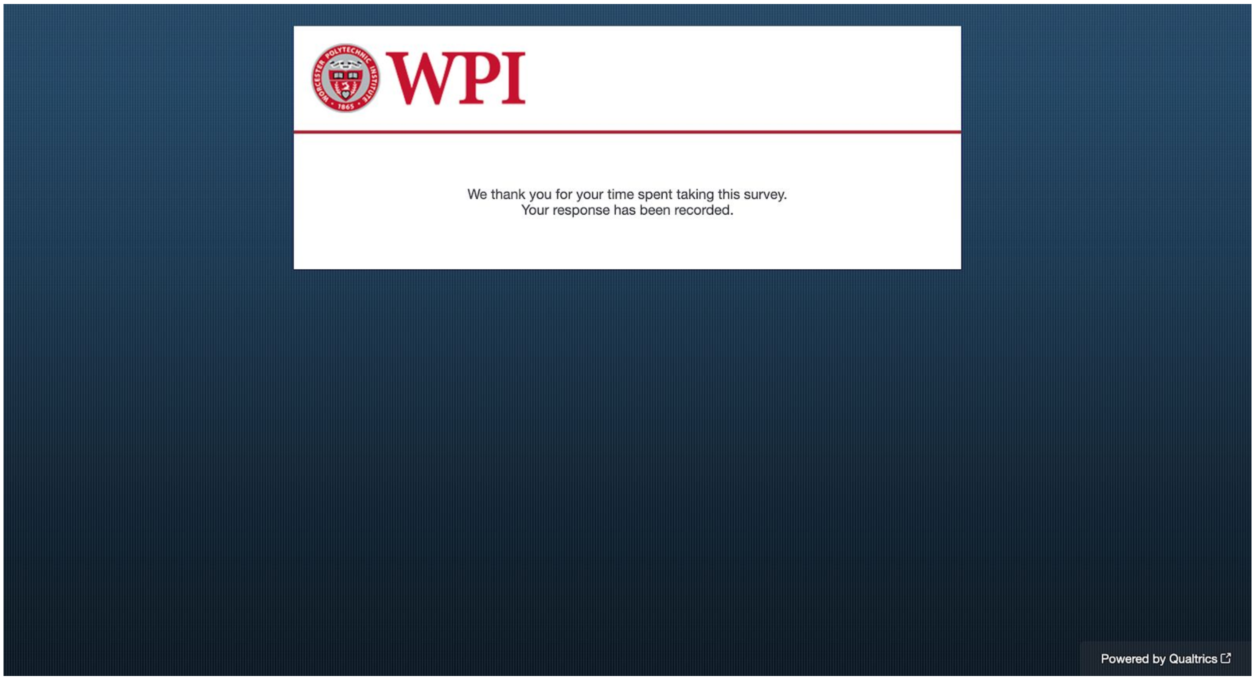
D Term 22 Sites:

	Very Interested	Interested	Not Interested
Asuncion, Paraguay (D)	○	○	○
Berlin, Germany (D)	○	○	○
Bucharest, Romania (D)	○	○	○
Copenhagen, Denmark (D)	○	○	○
Cuenca, Ecuador (D)	○	○	○
Farm Stay (Paxton, MA) (D)	○	○	○
London, England (D)	○	○	○
Mandi, India (D)	○	○	○
Melbourne, Australia (D)	○	○	○
Moscow, Russia (D)	○	○	○
San Juan, Puerto Rico (D)	○	○	○
Taipei, Taiwan (D)	○	○	○
Thessaloniki, Greece (D)	○	○	○
Water Resource Outreach Center (D)	○	○	○



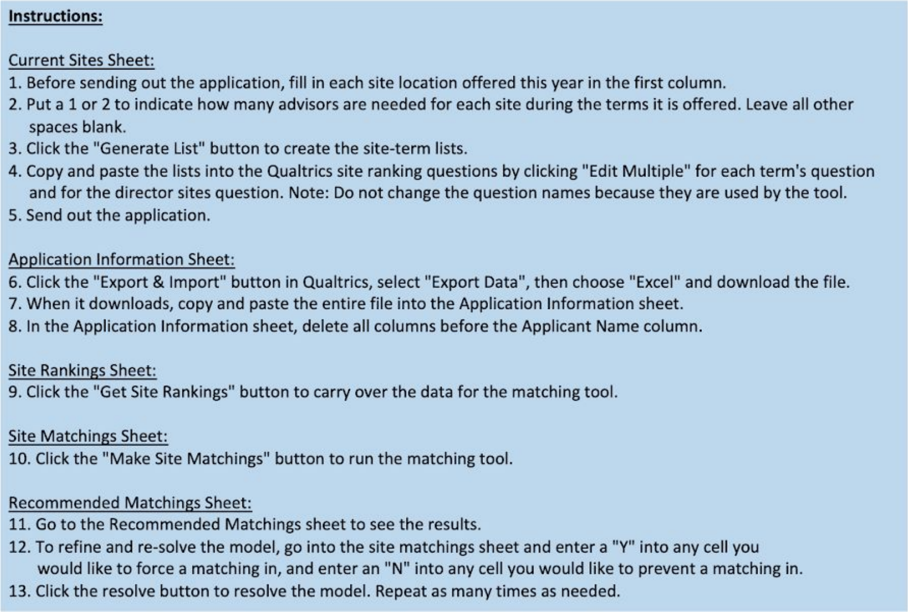
Windhoek, Namibia (D)	○	○	○
Worcester, England (D)	○	○	○
Worcester, MA (D)	○	○	○
Yerevan, Armenia (D)	○	○	○

If you have any other preferences, comments, or concerns before submitting this application, please indicate them here:



We thank you for your time spent taking this survey.
Your response has been recorded.

Appendix B: Matching Tool Instructions



Instructions:

Current Sites Sheet:

1. Before sending out the application, fill in each site location offered this year in the first column.
2. Put a 1 or 2 to indicate how many advisors are needed for each site during the terms it is offered. Leave all other spaces blank.
3. Click the "Generate List" button to create the site-term lists.
4. Copy and paste the lists into the Qualtrics site ranking questions by clicking "Edit Multiple" for each term's question and for the director sites question. Note: Do not change the question names because they are used by the tool.
5. Send out the application.

Application Information Sheet:

6. Click the "Export & Import" button in Qualtrics, select "Export Data", then choose "Excel" and download the file.
7. When it downloads, copy and paste the entire file into the Application Information sheet.
8. In the Application Information sheet, delete all columns before the Applicant Name column.

Site Rankings Sheet:

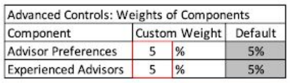
9. Click the "Get Site Rankings" button to carry over the data for the matching tool.

Site Matchings Sheet:

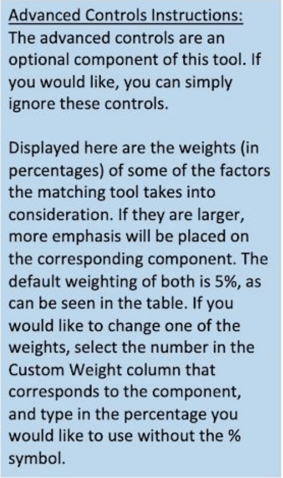
10. Click the "Make Site Matchings" button to run the matching tool.

Recommended Matchings Sheet:

11. Go to the Recommended Matchings sheet to see the results.
12. To refine and re-solve the model, go into the site matchings sheet and enter a "Y" into any cell you would like to force a matching in, and enter an "N" into any cell you would like to prevent a matching in.
13. Click the resolve button to resolve the model. Repeat as many times as needed.



Advanced Controls: Weights of Components			
Component	Custom Weight		Default
Advisor Preferences	5	%	5%
Experienced Advisors	5	%	5%



Advanced Controls Instructions:

The advanced controls are an optional component of this tool. If you would like, you can simply ignore these controls.

Displayed here are the weights (in percentages) of some of the factors the matching tool takes into consideration. If they are larger, more emphasis will be placed on the corresponding component. The default weighting of both is 5%, as can be seen in the table. If you would like to change one of the weights, select the number in the Custom Weight column that corresponds to the component, and type in the percentage you would like to use without the % symbol.

Appendix C: Matching Tool VBA Code

Module 1: Step1CurrentSites

```
Sub GenerateSiteTermList()

   'For Cells(j, k), j = row and k = column
   'Clear out old site term list
   For q = 9 To 13
      n = 12
      Do While Cells(n, q).Value <> Empty
         Cells(n, q).Value = ""
         n = n + 1
      Loop
   Next q

   'Iterate through list of sites and add "Site Name (Term)" to the list for each term a site is offered during
   p = 9
   For k = 2 To 6
      m = 12
      j = 2
      Do While Cells(j, 1).Value <> Empty
         If Cells(j, k).Value <> Empty Then
            Cells(m, p).Value = Cells(j, 1).Value & " (" & Cells(1, k).Value & ")"
            m = m + 1
         End If
         j = j + 1
      Loop
      p = p + 1
   Next k

End Sub
```

Module 2: Step2SiteRankings

```
Sub GetSiteRankings()

  'Turn off screen updating while performing tasks
  Application.ScreenUpdating = False

  'Clear out old data
  Sheets("Site Rankings").Range("A3:KA500").Select
  Selection.Delete
  Sheets("Site Rankings").Range("B1:KA2").Select
  Selection.Delete

  'Copy in advisor names
  Sheets("Application Information").Range("A2:A1000").Copy
  Sheets("Site Rankings").Range("A3").Select
  Sheets("Site Rankings").Paste

  'Copy in site names and site preference data
  Sheets("Application Information").Select
  n = 3
  Do While Cells(1, n).Value <> Empty
    If Cells(1, n).Value = "E Term Sites_1" Then
      Range(Cells(2, n), Cells(1000, n + 300)).Copy
    End If
    n = n + 1
  Loop
  Sheets("Site Rankings").Select
  Sheets("Site Rankings").Range("B3").Select
  Sheets("Site Rankings").Paste
  Range("A1:DA500").NumberFormat = "General"
  Sheets("Site Rankings").Range("A1").Select

  Dim j As Integer
  Dim k As Integer

  'Determine the index of the last row (advisor)
  Dim LastRow As Integer
  LastRow = 3
  Do While Cells(LastRow + 1, 1).Value <> Empty
    LastRow = LastRow + 1
  Loop

  'Determine the index of the last column (site-term)
  Dim LastCol As Integer
  LastCol = 1
  Do While Cells(3, LastCol + 1).Value <> Empty
    LastCol = LastCol + 1
  Loop
  Columns(LastCol).EntireColumn.Delete
  LastCol = LastCol - 1
  Columns(LastCol).EntireColumn.Delete
  LastCol = LastCol - 1

  'Simplify column headings to be just site names
  For k = 2 To LastCol
    Cells(3, k).Value = Replace(Cells(3, k).Value, "E Term ", "")
    Cells(3, k).Value = Replace(Cells(3, k).Value, "A Term ", "")
    Cells(3, k).Value = Replace(Cells(3, k).Value, "B Term ", "")
    Cells(3, k).Value = Replace(Cells(3, k).Value, "C Term ", "")
    Cells(3, k).Value = Replace(Cells(3, k).Value, "D Term ", "")
    Cells(3, k).Value = Replace(Cells(3, k).Value, " Sites: - ", "")
    Cells(3, k).Value = Replace(Cells(3, k).Value, "1", "")
    Cells(3, k).Value = Replace(Cells(3, k).Value, "2", "")
    Cells(3, k).Value = Replace(Cells(3, k).Value, "3", "")
    Cells(3, k).Value = Replace(Cells(3, k).Value, "4", "")
    Cells(3, k).Value = Replace(Cells(3, k).Value, "5", "")
    Cells(3, k).Value = Replace(Cells(3, k).Value, "6", "")
    Cells(3, k).Value = Replace(Cells(3, k).Value, "7", "")
    Cells(3, k).Value = Replace(Cells(3, k).Value, "8", "")
    Cells(3, k).Value = Replace(Cells(3, k).Value, "9", "")
        Cells(3, k).Value = Replace(Cells(3, k).Value, "If you have any other preferences, comments, or concerns before submitting this application, please indicate them here:", "Comments")
  Next k

  'Convert site interest data into a numerical format
  For k = 2 To LastCol
    For j = 4 To LastRow
      If Cells(j, k).Value = "Very Interested" Then
```

```
                Cells(j, k).Value = "2"
            End If
            If Cells(j, k).Value = "Not Interested" Then
                Cells(j, k).Value = "0"
            End If
            If Cells(j, k).Value = "Interested" Then
                Cells(j, k).Value = "1"
            End If
            If Cells(j, k).Value = "" Then
                Cells(j, k).Value = "0"
            End If
        Next j
    Next k

    'Add column sums
    Cells(1, 2).Value = "Site Interest:"
    For k = 2 To LastCol
        Cells(2, k).Value = "=Sum(" & Range(Cells(4, k), Cells(LastRow, k)).Address & ")"
    Next k

    'Turn on screen updating after performing tasks
    Application.ScreenUpdating = True

End Sub
```

Module 3: Step3SiteMatchings

```
Sub MakeSiteMatchings()

     'Turn off screen updating, set calculations to manual, and disable events while performing tasks
    Application.ScreenUpdating = False
    Application.Calculation = xlManual
    Application.EnableEvents = False

    'Clear out old data
    Sheets("Site Matchings").Range("A3:KA500").Select
    Selection.Delete
    Sheets("Site Matchings").Range("B1:KA2").Select
```

```
Selection.Delete

'Copy in advisor names and site names
Application.CopyObjectsWithCells = False
Sheets("Site Rankings").Range("A1:KA500").Copy
Sheets("Site Matchings").Range("A1").Select
Sheets("Site Matchings").Paste
Application.CopyObjectsWithCells = True
Range("B4:KA500").Select
Selection.Delete
Columns("B:C").Insert Shift:=xlToRight
Range("A1:KA500").NumberFormat = "General"
Columns("B:C").ColumnWidth = 10
Sheets("Site Matchings").Range("A1").Select

Dim j As Integer
Dim k As Integer

'Determine the index of the last row (advisor)
Dim LastRow As Integer
LastRow = 3
Do While Cells(LastRow + 1, 1).Value <> Empty
   LastRow = LastRow + 1
Loop

'Determine the index of the last column (site-term)
Dim LastCol As Integer
LastCol = 4
Do While Cells(3, LastCol + 1).Value <> Empty
   LastCol = LastCol + 1
Loop

'Determine the index of the "Full Time" column
Dim FullTimeCol As Integer
n = 1
Do While Sheets("Application Information").Cells(1, n).Value <> Empty
   If Sheets("Application Information").Cells(1, n).Value = "Full Time" Then
      FullTimeCol = n
   End If
```

```
        n = n + 1
    Loop

    'Begin creating optimization model
    SolverReset
    SolverOptions IntTolerance:=0
    Cells(LastRow + 1, 2).Value = Sheets("Instructions").Cells(6, 16).Value / 100
    Cells(LastRow + 2, 2).Value = Sheets("Instructions").Cells(7, 16).Value / 100
    Range(Cells(LastRow + 1, 2), Cells(LastRow + 2, 2)).Font.Color = vbWhite
    Cells(1, 1).Value = "=SumIf('Application Information'!" & Range(Cells(3, FullTimeCol), Cells(LastRow - 1, FullTimeCol)).Address & ", ""Yes"", " & "'Site Matchings'!" & Range(Cells(4, 2), Cells(LastRow, 2)).Address & ") + (Sum('Site Matchings'!" & Range(Cells(LastRow + 1, 2), Cells(LastRow + 1, 2)).Address & ") * SumProduct('Site Matchings'!" & Range(Cells(4, 4), Cells(LastRow, LastCol)).Address & ", 'Site Rankings'!" & Range(Cells(4, 2), Cells(LastRow, LastCol - 2)).Address & ")) - (Sum('Site Matchings'!" & Range(Cells(LastRow + 2, 2), Cells(LastRow + 2, 2)).Address & ") * Sum('Site Matchings'!" & Range(Cells(LastRow + 1, 4), Cells(LastRow + 1, LastCol)).Address & "))"
    SolverOK SetCell:=Cells(1, 1), MaxMinVal:=1, ByChange:=Range(Cells(4, 4), Cells(LastRow + 1, LastCol))

    'First constraint set: Declaring variables as binary
    SolverAdd cellRef:=Range(Cells(4, 4), Cells(LastRow + 1, LastCol)), relation:=5

    'Second constraint set: Prevent matchings for sites ranked with a 0
    For j = 4 To LastRow
        n = 1
        Do While Sheets("Application Information").Cells(1, n).Value <> Empty
            If Sheets("Application Information").Cells(1, n).Value = "Director" Then
                If Sheets("Application Information").Cells(j - 1, n).Value = "Yes" Then
                    p = 1
                    Do While Sheets("Application Information").Cells(1, p).Value <> Empty
                        If Sheets("Application Information").Cells(1, p).Value = "Director Sites" Then
                            If Sheets("Application Information").Cells(j - 1, p).Value <> Empty Then
                                Dim b() As String
                                b = Split(Sheets("Application Information").Cells(j - 1, p).Value, ",")
                                Sheets("Site Rankings").Select
                                For I = 0 To UBound(b, 1)
                                    Set c = Range("B3", Cells(3, LastCol)).Find(b(I))
                                    Cells(j, c.Column).Value = "2"
```

```
                     Next I
                  End If
                  Sheets("Site Matchings").Select
               End If
               p = p + 1
            Loop
         End If
      End If
      n = n + 1
   Loop
Next j

For j = 4 To LastRow
   For k = 4 To LastCol
      If Sheets("Site Rankings").Cells(j, k - 2).Value = "0" Then
         SolverAdd cellRef:=Cells(j, k), relation:=2, formulatext:=0
      End If
   Next k
Next j

'Third constraint set: Directors who want to advise at their site are automatically matched there
For j = 4 To LastRow
   n = 1
   Do While Sheets("Application Information").Cells(1, n).Value <> Empty
      If Sheets("Application Information").Cells(1, n).Value = "Director" Then
         If Sheets("Application Information").Cells(j - 1, n).Value = "Yes" Then
            p = 1
            Do While Sheets("Application Information").Cells(1, p).Value <> Empty
               If Sheets("Application Information").Cells(1, p).Value = "Director Sites" Then
                  If Sheets("Application Information").Cells(j - 1, p).Value <> Empty Then
                     b = Split(Sheets("Application Information").Cells(j - 1, p).Value, "),")
                     For I = 0 To UBound(b, 1)
                        Set c = Range("B3", Cells(3, LastCol)).Find(b(I))
                        SolverAdd cellRef:=Cells(j, c.Column), relation:=2, formulatext:=1
                     Next I
                  End If
               End If
               p = p + 1
            Loop
         End If
      End If
      n = n + 1
   Loop
Next j

'Fourth constraint set: There are no more advisors per site than the capacity of that site
Cells(1, 2).Value = "Assigned Advisors:"
Cells(2, 2).Value = "Maximum Advisors:"
k = 2
n = 4
Do While Sheets("Current Sites").Cells(1, k).Value <> Empty
   j = 2
   Do While Sheets("Current Sites").Cells(j, 1).Value <> Empty
      If Sheets("Current Sites").Cells(j, k).Value <> Empty Then
         Cells(2, n).Value = Sheets("Current Sites").Cells(j, k).Value
         n = n + 1
      End If
      j = j + 1
   Loop
   k = k + 1
Loop
For k = 4 To LastCol
   Cells(1, k).Value = "=Sum(" & Range(Cells(4, k), Cells(LastRow, k)).Address & ")"
   SolverAdd cellRef:=Cells(1, k), relation:=1, formulatext:=Cells(2, k)
Next k

'Fifth constraint set: Advisors are not assigned to more sites than they are willing to advise
Cells(3, 2).Value = "Assigned Terms:"
Cells(3, 3).Value = "Maximum Assignments:"
For j = 4 To LastRow
   Cells(j, 2).Value = "=Sum(" & Range(Cells(j, 4), Cells(j, LastCol)).Address & ")"
   n = 1
   Do While Sheets("Application Information").Cells(1, n).Value <> Empty
      If Sheets("Application Information").Cells(1, n).Value = "How many times" Then
         Cells(j, 3).Value = Sheets("Application Information").Cells(j - 1, n).Value
      End If
      n = n + 1
   Loop
```

```
        If Cells(j, 3).Value = "3+" Then
            Cells(j, 3).Value = "5"
        End If
        SolverAdd cellRef:=Cells(j, 2), relation:=1, formulatext:=Cells(j, 3)
    Next j

    'Sixth constraint set: IGSD faculty members need to advise at least once a year
    For j = 4 To LastRow
        n = 1
        Do While Sheets("Application Information").Cells(1, n).Value <> Empty
            If Sheets("Application Information").Cells(1, n).Value = "IGSD" Then
                If Sheets("Application Information").Cells(j - 1, n).Value = "Yes" Then
                    SolverAdd cellRef:=Cells(j, 2), relation:=3, formulatext:=1
                End If
            End If
            n = n + 1
        Loop
    Next j

    'Seventh constraint set: Prevent being assigned more than once in the same term and in back to back terms
    Dim FirstE As Integer
    FirstE = 4

    Dim LastE As Integer
    LastE = FirstE
    Do While InStr(1, Cells(3, LastE + 1).Value, "(E)", 1) > 2
        LastE = LastE + 1
    Loop

    Dim FirstA As Integer
    FirstA = LastE + 1

    Dim LastA As Integer
    LastA = FirstA
    Do While InStr(1, Cells(3, LastA + 1).Value, "(A)", 1) > 2
        LastA = LastA + 1
    Loop
    Dim FirstB As Integer
    FirstB = LastA + 1

    Dim LastB As Integer
    LastB = FirstB
    Do While InStr(1, Cells(3, LastB + 1).Value, "(B)", 1) > 2
        LastB = LastB + 1
    Loop

    Dim FirstC As Integer
    FirstC = LastB + 1

    Dim LastC As Integer
    LastC = FirstC
    Do While InStr(1, Cells(3, LastC + 1).Value, "(C)", 1) > 2
        LastC = LastC + 1
    Loop

    Dim FirstD As Integer
    FirstD = LastC + 1

    Dim LastD As Integer
    LastD = LastCol

    For j = 4 To LastRow
         Cells(j, LastCol + 2).Value = "=Sum(" & Range(Cells(j, FirstE), Cells(j, LastE)).Address &
")"
        SolverAdd cellRef:=Cells(j, LastCol + 2), relation:=1, formulatext:=1
        Cells(j, LastCol + 3).Value = "=Sum(" & Range(Cells(j, FirstA), Cells(j, LastB)).Address &
")"
        SolverAdd cellRef:=Cells(j, LastCol + 3), relation:=1, formulatext:=1
         Cells(j, LastCol + 4).Value = "=Sum(" & Range(Cells(j, FirstB), Cells(j, LastC)).Address &
")"
        SolverAdd cellRef:=Cells(j, LastCol + 4), relation:=1, formulatext:=1
        Cells(j, LastCol + 5).Value = "=Sum(" & Range(Cells(j, FirstC), Cells(j, LastD)).Address &
")"
        SolverAdd cellRef:=Cells(j, LastCol + 5), relation:=1, formulatext:=1
          Cells(j, LastCol + 6).Value = "=Sum(" & Range(Cells(j, FirstD), Cells(j, LastD)).Address
& ")"
```

```
        SolverAdd cellRef:=Cells(j, LastCol + 6), relation:=1, formulatext:=1
        If Sheets("Application Information").Cells(j - 1, 9).Value = "Yes" Then
            SolverAdd cellRef:=Cells(j, LastCol + 2), relation:=1, formulatext:=0
            SolverAdd cellRef:=Cells(j, LastCol + 3), relation:=1, formulatext:=0
        End If
    Next j
    Range(Cells(1, LastCol + 2), Cells(LastRow, LastCol + 6)).Font.Color = vbWhite

    'Eighth constraint set: At least one experienced advisor per site
    For k = 4 To LastCol
        p = LastRow + 2
        For j = 4 To LastRow
            n = 1
            Do While Sheets("Application Information").Cells(1, n).Value <> Empty
                If Sheets("Application Information").Cells(1, n).Value = "Experience" Then
                    If Sheets("Application Information").Cells(j - 1, n).Value <> "0" Then
                        If Cells(p, k).Value <> Empty Then
                            p = p + 1
                            Cells(p, k).Value = Cells(j, 1).Value
                        End If
                        If IsEmpty(Cells(p, k).Value) Then
                            Cells(p, k).Value = Cells(j, 1).Value
                        End If
                    End If
                End If
                n = n + 1
            Loop
        Next j
        S = p + 1
        For q = LastRow + 2 To p
            For r = 4 To LastRow
                If Cells(r, 1).Value = Cells(q, k).Value Then
                    Cells(p + 1, k).Value = "=Sum(" & Range(Cells(r, k), Cells(r, k)).Address & ")"
                    p = p + 1
                End If
            Next r
        Next q
        Cells(LastRow + LastRow + LastRow - 1, k).Value = "=Sum(" & Range(Cells(S, k), Cells(p, k)).Address & ")"
```

```
       Cells(LastRow + LastRow + LastRow - 2, k).Value = "=Sum(" & Range(Cells(LastRow +
1, k), Cells(LastRow + 1, k)).Address & ")"
       Cells(LastRow + LastRow + LastRow - 3, k).Value = "=Sum(" & Range(Cells(LastRow +
LastRow + LastRow - 2, k), Cells(LastRow + LastRow + LastRow - 1, k)).Address & ")"
              SolverAdd cellRef:=Cells(LastRow + LastRow + LastRow - 3, k), relation:=3,
formulatext:=1
   Next k

   'Ninth constraint set: Language requirements for Moscow and Paraguay
   For k = 4 To LastCol
      If InStr(1, Cells(3, k).Value, "Moscow", 1) > 0 Then
         p = LastRow + LastRow + LastRow + 2
         n = 1
         Do While Sheets("Application Information").Cells(2, n).Value <> Empty
            If InStr(1, Sheets("Application Information").Cells(2, n).Value, "Russian", 1) > 0 Then
               For m = 3 To LastRow - 1
                       If InStr(1, Sheets("Application Information").Cells(m, n).Value, "(2)", 1) > 0
Then
                     If Cells(p, k).Value <> Empty Then
                        p = p + 1
                        Cells(p, k).Value = Cells(m + 1, 1).Value
                     End If
                     If IsEmpty(Cells(p, k).Value) Then
                        Cells(p, k).Value = Cells(m + 1, 1).Value
                     End If
                  End If
               Next m
            End If
            n = n + 1
         Loop
         S = p + 1
         For q = LastRow + LastRow + LastRow + 2 To p
            For r = 4 To LastRow
               If InStr(1, Cells(r, 1).Value, Cells(q, k).Value, 1) > 0 Then
                  Cells(p + 1, k).Value = "=Sum(" & Range(Cells(r, k), Cells(r, k)).Address & ")"
                  p = p + 1
               End If
            Next r
         Next q
```

```
        Cells(LastRow + LastRow + LastRow + 1, k).Value = "=Sum(" & Range(Cells(S, k),
Cells(p, k)).Address & ")"
           SolverAdd cellRef:=Cells(LastRow + LastRow + LastRow + 1, k), relation:=3,
formulatext:=1
    End If
    If InStr(1, Cells(3, k).Value, "Paraguay", 1) > 0 Then
      p = LastRow + LastRow + LastRow + LastRow + LastRow + 2
      n = 1
      Do While Sheets("Application Information").Cells(2, n).Value <> Empty
        If InStr(1, Sheets("Application Information").Cells(2, n).Value, "Spanish", 1) > 0 Then
          For m = 3 To LastRow - 1
               If InStr(1, Sheets("Application Information").Cells(m, n).Value, "(2)", 1) > 0
Then
              If Cells(p, k).Value <> Empty Then
                p = p + 1
                Cells(p, k).Value = Cells(m + 1, 1).Value
              End If
              If IsEmpty(Cells(p, k).Value) Then
                Cells(p, k).Value = Cells(m + 1, 1).Value
              End If
            End If
          Next m
        End If
        n = n + 1
      Loop
      S = p + 1
      For q = LastRow + LastRow + LastRow + LastRow + LastRow + 2 To p
        For r = 4 To LastRow
          If InStr(1, Cells(r, 1).Value, Cells(q, k).Value, 1) > 0 Then
            Cells(p + 1, k).Value = "=Sum(" & Range(Cells(r, k), Cells(r, k)).Address & ")"
            p = p + 1
          End If
        Next r
      Next q
      Cells(LastRow + LastRow + LastRow + LastRow + LastRow + 1, k).Value = "=Sum(" &
Range(Cells(S, k), Cells(p, k)).Address & ")"
       SolverAdd cellRef:=Cells(LastRow + LastRow + LastRow + LastRow + LastRow + 1,
k), relation:=3, formulatext:=1
    End If
```

```
  Next k
   Range(Cells(LastRow + 1, 4), Cells(LastRow + LastRow + LastRow + LastRow + LastRow +
LastRow, LastCol)).Font.Color = vbWhite

  'Perform OpenSolver
  Dim Result As OpenSolverResult
  Result = RunOpenSolver(False, True) ' do not relax IP, do hide dialogs

  'Highlight all matches (cells containing a 1)
  For j = 4 To LastRow
    For k = 4 To LastCol
      If Cells(j, k).Value = 1 Then
        Cells(j, k).Interior.ColorIndex = 43
      End If
    Next k
  Next j


'______________________________________________________________________________
________

  'Clear out old data
  Sheets("Recommended Matchings").Select
  Range("A8:C500").Select
  Selection.Clear
  Range("E16:H18").Select
  Selection.Clear
  Range("A1").Select
  Range("F1", "K1").ClearFormats
  Range("F8", "K8").ClearFormats

  'Calculate number of unfilled spots, unmatched advisors, and satisfaction score
     Cells(1, 2).Value = "=Sum('Site Matchings'!" & Range(Cells(4, 4), Cells(LastRow,
LastCol)).Address & ")"
  Range(Cells(1, 2), Cells(2, 2)).Font.Color = vbWhite
   Cells(3, 4).Value = "=Sum('Site Matchings'!" & Range(Cells(2, 4), Cells(2, LastCol)).Address
& ")"
  Cells(3, 3).Value = Cells(3, 4).Value
  Cells(3, 2).Value = Cells(3, 3).Value - Cells(1, 2).Value
```

```
    Cells(4, 4).Value = "=CountIf('Site Matchings'!" & Range(Cells(4, 2), Cells(LastRow, 2)).Address & ", " & "0" & ")"
    Cells(4, 3).Value = LastRow - 3
    Cells(4, 2).Value = Cells(4, 4).Value
    Cells(5, 4).Value = "=SumProduct('Site Matchings'!" & Range(Cells(4, 4), Cells(LastRow, LastCol)).Address & ", 'Site Rankings'!" & Range(Cells(4, 2), Cells(LastRow, LastCol - 2)).Address & ")"
    Cells(5, 3).Value = (Cells(1, 2).Value) * 2
    Cells(5, 2).Value = Cells(5, 4).Value
    Range(Cells(3, 4), Cells(5, 4)).Font.Color = vbWhite

    'Add term and site labels and fill in names of assigned advisors
    j = 8
    Cells(j, 1).Value = "E Term Sites"
    Cells(j, 1).Interior.ColorIndex = 15
    j = j + 1
    For k = 4 To LastE
        Cells(j, 1).Value = Sheets("Site Matchings").Cells(3, k).Value
        For m = 4 To LastRow
            If Sheets("Site Matchings").Cells(m, k).Value <> 0 Then
                If Cells(j, 2).Value <> Empty Then
                    Cells(j, 3).Value = Sheets("Site Matchings").Cells(m, 1)
                End If
                If IsEmpty(Cells(j, 2).Value) Then
                    Cells(j, 2).Value = Sheets("Site Matchings").Cells(m, 1)
                End If
            End If
        Next m
        j = j + 1
    Next k
    Cells(j, 1).Value = "A Term Sites"
    Cells(j, 1).Interior.ColorIndex = 15
    j = j + 1
    For k = FirstA To LastA
        Cells(j, 1).Value = Sheets("Site Matchings").Cells(3, k).Value
        For m = 4 To LastRow
            If Sheets("Site Matchings").Cells(m, k).Value <> 0 Then
                If Cells(j, 2).Value <> Empty Then
                    Cells(j, 3).Value = Sheets("Site Matchings").Cells(m, 1)
```

```
         End If
         If IsEmpty(Cells(j, 2).Value) Then
            Cells(j, 2).Value = Sheets("Site Matchings").Cells(m, 1)
         End If
      End If
   Next m
   j = j + 1
Next k
Cells(j, 1).Value = "B Term Sites"
Cells(j, 1).Interior.ColorIndex = 15
j = j + 1
For k = FirstB To LastB
   Cells(j, 1).Value = Sheets("Site Matchings").Cells(3, k).Value
   For m = 4 To LastRow
      If Sheets("Site Matchings").Cells(m, k).Value <> 0 Then
         If Cells(j, 2).Value <> Empty Then
            Cells(j, 3).Value = Sheets("Site Matchings").Cells(m, 1)
         End If
         If IsEmpty(Cells(j, 2).Value) Then
            Cells(j, 2).Value = Sheets("Site Matchings").Cells(m, 1)
         End If
      End If
   Next m
   j = j + 1
Next k
Cells(j, 1).Value = "C Term Sites"
Cells(j, 1).Interior.ColorIndex = 15
j = j + 1
For k = FirstC To LastC
   Cells(j, 1).Value = Sheets("Site Matchings").Cells(3, k).Value
   For m = 4 To LastRow
      If Sheets("Site Matchings").Cells(m, k).Value <> 0 Then
         If Cells(j, 2).Value <> Empty Then
            Cells(j, 3).Value = Sheets("Site Matchings").Cells(m, 1)
         End If
         If IsEmpty(Cells(j, 2).Value) Then
            Cells(j, 2).Value = Sheets("Site Matchings").Cells(m, 1)
         End If
      End If
```

```
        Next m
        j = j + 1
    Next k
    Cells(j, 1).Value = "D Term Sites"
    Cells(j, 1).Interior.ColorIndex = 15
    j = j + 1
    For k = FirstD To LastD
        Cells(j, 1).Value = Sheets("Site Matchings").Cells(3, k).Value
        For m = 4 To LastRow
            If Sheets("Site Matchings").Cells(m, k).Value <> 0 Then
                If Cells(j, 2).Value <> Empty Then
                    Cells(j, 3).Value = Sheets("Site Matchings").Cells(m, 1)
                End If
                If IsEmpty(Cells(j, 2).Value) Then
                    Cells(j, 2).Value = Sheets("Site Matchings").Cells(m, 1)
                End If
            End If
        Next m
        j = j + 1
    Next k

    'Turn on screen updating, set calculations to automatic, and enable events after performing tasks
    Application.ScreenUpdating = True
    Application.Calculation = xlAutomatic
    Application.EnableEvents = True

End Sub
```

Module 4: Step4CommentsAndColoring

```
Sub AddComments()

    'Determine the index of the last row (advisor)
    Dim LastRow As Integer
    LastRow = 8
    Do While Cells(LastRow + 1, 1).Value <> Empty
        LastRow = LastRow + 1
    Loop
```

```
  'Add department, travel, language, and final comments to advisor cells
  For j = 9 To LastRow
    For k = 2 To 3
      If Not Range(Cells(j, k), Cells(j, k)).comment Is Nothing Then
        Range(Cells(j, k), Cells(j, k)).comment.Delete
      End If
      If Cells(j, k).Value <> Empty Then
        m = 3
        Do While Sheets("Application Information").Cells(m, 1).Value <> Empty
          If Sheets("Application Information").Cells(m, 1).Value = Cells(j, k).Value Then
            n = 2
            Do While Sheets("Application Information").Cells(1, n).Value <> Empty
              If Sheets("Application Information").Cells(1, n).Value = "Department" Then
                           departmentComment = "Department: " & Sheets("Application
Information").Cells(m, n).Value & vbNewLine
              End If
                 If Sheets("Application Information").Cells(1, n).Value = "Travel comments"
Then
                  travelComment = "Travel: " & Sheets("Application Information").Cells(m,
n).Value & vbNewLine
              End If
                      If Sheets("Application Information").Cells(1, n).Value = "Language
comments" Then
                              languageComment = "Language: " & Sheets("Application
Information").Cells(m, n).Value & vbNewLine
              End If
                  If Sheets("Application Information").Cells(1, n).Value = "Final comments"
Then
                     finalComment = "Final: " & Sheets("Application Information").Cells(m,
n).Value
              End If
              n = n + 1
            Loop
          End If
          m = m + 1
        Loop
              comment = departmentComment & travelComment & languageComment &
finalComment
```

```
                Range(Cells(j, k), Cells(j, k)).AddComment.Text Text:=comment
                Range(Cells(j, k), Cells(j, k)).comment.Shape.Height = 200
                Range(Cells(j, k), Cells(j, k)).comment.Shape.Width = 200
            End If
        Next k
    Next j

End Sub
Sub PreferenceColoring()

    'Determine the index of the last row (advisor)
    Dim LastRow As Integer
    LastRow = 8
    Do While Cells(LastRow + 1, 1).Value <> Empty
        LastRow = LastRow + 1
    Loop

    'Color block above button green
    Range("F1", "K1").ClearFormats
    Range("F8", "K8").ClearFormats
    Cells(1, 6).Interior.ColorIndex = 4

    'Color advisor name green if very interested, yellow if interested, and grey if not interested
    For j = 9 To LastRow
        For k = 2 To 3
            If Cells(j, k).Value <> Empty Then
                m = 4
                Do While Sheets("Site Rankings").Cells(m, 1).Value <> Empty
                    If Sheets("Site Rankings").Cells(m, 1).Value = Cells(j, k).Value Then
                        n = 2
                        Do While Sheets("Site Rankings").Cells(3, n).Value <> Empty
                            If Sheets("Site Rankings").Cells(3, n).Value = Cells(j, 1).Value Then
                                If Sheets("Site Rankings").Cells(m, n).Value = "2" Then
                                    Cells(j, k).Interior.ColorIndex = 4
                                End If
                                If Sheets("Site Rankings").Cells(m, n).Value = "1" Then
                                    Cells(j, k).Interior.ColorIndex = 6
                                End If
                                If Sheets("Site Rankings").Cells(m, n).Value = "0" Then
```

```
                           Cells(j, k).Interior.ColorIndex = 15
                        End If
                     End If
                     n = n + 1
                  Loop
               End If
               m = m + 1
            Loop
         End If
      Next k
   Next j

End Sub
Sub ExperienceColoring()

   'Determine the index of the last row (advisor)
   Dim LastRow As Integer
   LastRow = 8
   Do While Cells(LastRow + 1, 1).Value <> Empty
      LastRow = LastRow + 1
   Loop

   'Color block above button green
   Range("F1", "K1").ClearFormats
   Range("F8", "K8").ClearFormats
   Cells(1, 7).Interior.ColorIndex = 4

   'Color advisor name green if experienced, yellow if somewhat experienced, and grey if not
   For j = 9 To LastRow
      For k = 2 To 3
         If Cells(j, k).Value <> Empty Then
            m = 3
            Do While Sheets("Application Information").Cells(m, 1).Value <> Empty
               If Sheets("Application Information").Cells(m, 1).Value = Cells(j, k).Value Then
                  n = 1
                  Do While Sheets("Application Information").Cells(1, n).Value <> Empty
                     If Sheets("Application Information").Cells(1, n).Value = "Experience" Then
                        If Sheets("Application Information").Cells(m, n).Value = "3+" Then
                           Cells(j, k).Interior.ColorIndex = 4
```

```
                                End If
                                If Sheets("Application Information").Cells(m, n).Value = "1-2" Then
                                    Cells(j, k).Interior.ColorIndex = 6
                                End If
                                If Sheets("Application Information").Cells(m, n).Value = "0" Then
                                    Cells(j, k).Interior.ColorIndex = 15
                                End If
                            End If
                            n = n + 1
                        Loop
                    End If
                    m = m + 1
                Loop
            End If
        Next k
    Next j

End Sub
Sub GenderColoring()

    'Determine the index of the last row (advisor)
    Dim LastRow As Integer
    LastRow = 8
    Do While Cells(LastRow + 1, 1).Value <> Empty
        LastRow = LastRow + 1
    Loop

    'Color block above button green
    Range("F1", "K1").ClearFormats
    Range("F8", "K8").ClearFormats
    Cells(1, 8).Interior.ColorIndex = 4

    'Color advisor name green if female, grey if male, and yellow if neither
    For j = 9 To LastRow
        For k = 2 To 3
            If Cells(j, k).Value <> Empty Then
                m = 3
                Do While Sheets("Application Information").Cells(m, 1).Value <> Empty
                    If Sheets("Application Information").Cells(m, 1).Value = Cells(j, k).Value Then
```

```
                        n = 1
                        Do While Sheets("Application Information").Cells(1, n).Value <> Empty
                            If Sheets("Application Information").Cells(1, n).Value = "Gender" Then
                                If Sheets("Application Information").Cells(m, n).Value = "Female" Then
                                    Cells(j, k).Interior.ColorIndex = 4
                                End If
                                If Sheets("Application Information").Cells(m, n).Value = "Male" Then
                                    Cells(j, k).Interior.ColorIndex = 15
                                End If
                                If Sheets("Application Information").Cells(m, n).Value <> "Male" Then
                                    If Sheets("Application Information").Cells(m, n).Value <> "Female" Then
                                        Cells(j, k).Interior.ColorIndex = 6
                                    End If
                                End If
                            End If
                            n = n + 1
                        Loop
                    End If
                    m = m + 1
                Loop
            End If
        Next k
    Next j

End Sub
Sub DirectorsColoring()

    'Determine the index of the last row (advisor)
    Dim LastRow As Integer
    LastRow = 8
    Do While Cells(LastRow + 1, 1).Value <> Empty
        LastRow = LastRow + 1
    Loop

    'Color block above button green
    Range("F1", "K1").ClearFormats
    Range("F8", "K8").ClearFormats
    Cells(1, 9).Interior.ColorIndex = 4
```

```
  'Color advisor name green if director matching and grey if not
  For j = 9 To LastRow
    For k = 2 To 3
      If Cells(j, k).Value <> Empty Then
        m = 3
        Do While Sheets("Application Information").Cells(m, 1).Value <> Empty
          If Sheets("Application Information").Cells(m, 1).Value = Cells(j, k).Value Then
            n = 1
            Do While Sheets("Application Information").Cells(1, n).Value <> Empty
              If Sheets("Application Information").Cells(1, n).Value = "Director" Then
                If Sheets("Application Information").Cells(m, n).Value = "Yes" Then
                  Cells(j, k).Interior.ColorIndex = 15
                  p = 1
                  Do While Sheets("Application Information").Cells(1, p).Value <> Empty
                         If Sheets("Application Information").Cells(1, p).Value = "Director
Sites" Then
                           If InStr(1, Sheets("Application Information").Cells(m, p).Value,
Cells(j, 1).Value, 1) > 0 Then
                        Cells(j, k).Interior.ColorIndex = 4
                      End If
                    End If
                    p = p + 1
                  Loop
                End If
                If Sheets("Application Information").Cells(m, n).Value <> "Yes" Then
                  Cells(j, k).Interior.ColorIndex = 15
                End If
              End If
              n = n + 1
            Loop
          End If
          m = m + 1
        Loop
      End If
    Next k
  Next j

End Sub
Sub RetiredColoring()
```

```
'Determine the index of the last row (advisor)
Dim LastRow As Integer
LastRow = 8
Do While Cells(LastRow + 1, 1).Value <> Empty
   LastRow = LastRow + 1
Loop

'Color block above button green
Range("F1", "K1").ClearFormats
Range("F8", "K8").ClearFormats
Cells(1, 10).Interior.ColorIndex = 4

'Color advisor name green if full time and grey if not
For j = 9 To LastRow
   For k = 2 To 3
      If Cells(j, k).Value <> Empty Then
         m = 3
         Do While Sheets("Application Information").Cells(m, 1).Value <> Empty
            If Sheets("Application Information").Cells(m, 1).Value = Cells(j, k).Value Then
               n = 1
               Do While Sheets("Application Information").Cells(1, n).Value <> Empty
                  If Sheets("Application Information").Cells(1, n).Value = "Full Time" Then
                     If Sheets("Application Information").Cells(m, n).Value = "Yes" Then
                        Cells(j, k).Interior.ColorIndex = 4
                     End If
                     If Sheets("Application Information").Cells(m, n).Value <> "Yes" Then
                        Cells(j, k).Interior.ColorIndex = 15
                     End If
                  End If
                  n = n + 1
               Loop
            End If
            m = m + 1
         Loop
      End If
   Next k
Next j
```

```
End Sub
Sub LanguageComments()

    'Determine the index of the last row (advisor)
    Dim LastRow As Integer
    LastRow = 8
    Do While Cells(LastRow + 1, 1).Value <> Empty
        LastRow = LastRow + 1
    Loop

    'Add all language comments to advisor cells
    For j = 9 To LastRow
        For k = 2 To 3
            If Not Range(Cells(j, k), Cells(j, k)).comment Is Nothing Then
                Range(Cells(j, k), Cells(j, k)).comment.Delete
            End If
            If Cells(j, k).Value <> Empty Then
                m = 3
                Do While Sheets("Application Information").Cells(m, 1).Value <> Empty
                    If Sheets("Application Information").Cells(m, 1).Value = Cells(j, k).Value Then
                        n = 2
                        Do While Sheets("Application Information").Cells(1, n).Value <> Empty
                            If Sheets("Application Information").Cells(1, n).Value = "Language_1" Then
                                AlbanianComment = "Albanian: " & Sheets("Application
Information").Cells(m, n).Value & vbNewLine
                                ArabicComment = "Arabic: " & Sheets("Application Information").Cells(m,
n + 1).Value & vbNewLine
                                ArmenianComment = "Armenian: " & Sheets("Application
Information").Cells(m, n + 2).Value & vbNewLine
                                ChineseSimplifiedComment = "Chinese, Simplified: " &
Sheets("Application Information").Cells(m, n + 3).Value & vbNewLine
                                DanishComment = "Danish: " & Sheets("Application Information").Cells(m,
n + 4).Value & vbNewLine
                                FrenchComment = "French: " & Sheets("Application Information").Cells(m,
n + 5).Value & vbNewLine
                                GermanComment = "German: " & Sheets("Application
Information").Cells(m, n + 6).Value & vbNewLine
                                GreekComment = "Greek: " & Sheets("Application Information").Cells(m, n
+ 7).Value & vbNewLine
```

```
                        HebrewComment = "Hebrew: " & Sheets("Application Information").Cells(m, n + 8).Value & vbNewLine
                        HindiComment = "Hindi: " & Sheets("Application Information").Cells(m, n + 9).Value & vbNewLine
                        IcelandicComment = "Icelandic: " & Sheets("Application Information").Cells(m, n + 10).Value & vbNewLine
                        ItalianComment = "Italian: " & Sheets("Application Information").Cells(m, n + 11).Value & vbNewLine
                        JapaneseComment = "Japanese: " & Sheets("Application Information").Cells(m, n + 12).Value & vbNewLine
                        ChineseTraditionalComment = "Chinese, Traditional: " & Sheets("Application Information").Cells(m, n + 13).Value & vbNewLine
                        OshiwamboComment = "Oshiwambo: " & Sheets("Application Information").Cells(m, n + 14).Value & vbNewLine
                        RomanianComment = "Romanian: " & Sheets("Application Information").Cells(m, n + 15).Value & vbNewLine
                        RussianComment = "Russian: " & Sheets("Application Information").Cells(m, n + 16).Value & vbNewLine
                        SpanishComment = "Spanish: " & Sheets("Application Information").Cells(m, n + 17).Value & vbNewLine
                        ThaiComment = "Thai: " & Sheets("Application Information").Cells(m, n + 18).Value & vbNewLine
                        ZuluComment = "Zulu: " & Sheets("Application Information").Cells(m, n + 19).Value & vbNewLine
                    End If
                        If Sheets("Application Information").Cells(1, n).Value = "Language comments" Then
                        languageComment = "Language Comments: " & Sheets("Application Information").Cells(m, n).Value
                    End If
                    n = n + 1
                Loop
            End If
            m = m + 1
        Loop
                    comment = AlbanianComment & ArabicComment & ArmenianComment & ChineseSimplifiedComment & DanishComment & FrenchComment & GermanComment & GreekComment & HebrewComment & HindiComment & IcelandicComment & ItalianComment & JapaneseComment & ChineseTraditionalComment & OshiwamboComment &
```

```
RomanianComment & RussianComment & SpanishComment & ThaiComment & ZuluComment
& languageComment
            Range(Cells(j, k), Cells(j, k)).AddComment.Text Text:=comment
            Range(Cells(j, k), Cells(j, k)).comment.Shape.Height = 300
            Range(Cells(j, k), Cells(j, k)).comment.Shape.Width = 200
         End If
      Next k
   Next j

End Sub
Sub SpanishColoring()

   'Determine the index of the last row (advisor)
   Dim LastRow As Integer
   LastRow = 8
   Do While Cells(LastRow + 1, 1).Value <> Empty
      LastRow = LastRow + 1
   Loop

   'Color block above button green
   Range("F1", "K1").ClearFormats
   Range("F8", "K8").ClearFormats
   Cells(8, 6).Interior.ColorIndex = 4

   'Color advisor name green if proficient, yellow if conversational, and grey if non-speaker
   For j = 9 To LastRow
      For k = 2 To 3
         If Cells(j, k).Value <> Empty Then
            m = 3
            Do While Sheets("Application Information").Cells(m, 1).Value <> Empty
               If Sheets("Application Information").Cells(m, 1).Value = Cells(j, k).Value Then
                  n = 2
                  Do While Sheets("Application Information").Cells(2, n).Value <> Empty
                     If InStr(1, Sheets("Application Information").Cells(2, n).Value, "Spanish", 1) >
0 Then
                           If Sheets("Application Information").Cells(m, n).Value = "(2) Proficient"
Then
                          Cells(j, k).Interior.ColorIndex = 4
                       End If
```

```
                                If Sheets("Application Information").Cells(m, n).Value = "(1)
Conversational" Then
                                    Cells(j, k).Interior.ColorIndex = 6
                                End If
                                If Sheets("Application Information").Cells(m, n).Value = "(0) Non-Speaker"
Then
                                    Cells(j, k).Interior.ColorIndex = 15
                                End If
                                If Sheets("Application Information").Cells(m, n).Value = "" Then
                                    Cells(j, k).Interior.ColorIndex = 15
                                End If
                            End If
                            n = n + 1
                        Loop
                    End If
                    m = m + 1
                Loop
            End If
        Next k
    Next j

End Sub
Sub FrenchColoring()

'Determine the index of the last row (advisor)
    Dim LastRow As Integer
    LastRow = 8
    Do While Cells(LastRow + 1, 1).Value <> Empty
        LastRow = LastRow + 1
    Loop

    'Color block above button green
    Range("F1", "K1").ClearFormats
    Range("F8", "K8").ClearFormats
    Cells(8, 7).Interior.ColorIndex = 4

    'Color advisor name green if proficient, yellow if conversational, and grey if non-speaker
    For j = 9 To LastRow
        For k = 2 To 3
```

```
         If Cells(j, k).Value <> Empty Then
            m = 3
            Do While Sheets("Application Information").Cells(m, 1).Value <> Empty
               If Sheets("Application Information").Cells(m, 1).Value = Cells(j, k).Value Then
                  n = 2
                  Do While Sheets("Application Information").Cells(2, n).Value <> Empty
                       If InStr(1, Sheets("Application Information").Cells(2, n).Value, "French", 1) >
0 Then
                             If Sheets("Application Information").Cells(m, n).Value = "(2) Proficient"
Then
                           Cells(j, k).Interior.ColorIndex = 4
                        End If
                                         If Sheets("Application  Information").Cells(m,  n).Value  =  "(1)
Conversational" Then
                           Cells(j, k).Interior.ColorIndex = 6
                        End If
                         If Sheets("Application Information").Cells(m, n).Value = "(0) Non-Speaker"
Then
                           Cells(j, k).Interior.ColorIndex = 15
                        End If
                        If Sheets("Application Information").Cells(m, n).Value = "" Then
                           Cells(j, k).Interior.ColorIndex = 15
                        End If
                     End If
                     n = n + 1
                  Loop
               End If
               m = m + 1
            Loop
         End If
      Next k
   Next j

End Sub
Sub RussianColoring()

'Determine the index of the last row (advisor)
   Dim LastRow As Integer
   LastRow = 8
```

```
  Do While Cells(LastRow + 1, 1).Value <> Empty
    LastRow = LastRow + 1
  Loop

  'Color block above button green
  Range("F1", "K1").ClearFormats
  Range("F8", "K8").ClearFormats
  Cells(8, 8).Interior.ColorIndex = 4

  'Color advisor name green if proficient, yellow if conversational, and grey if non-speaker
  For j = 9 To LastRow
    For k = 2 To 3
      If Cells(j, k).Value <> Empty Then
        m = 3
        Do While Sheets("Application Information").Cells(m, 1).Value <> Empty
          If Sheets("Application Information").Cells(m, 1).Value = Cells(j, k).Value Then
            n = 2
            Do While Sheets("Application Information").Cells(2, n).Value <> Empty
              If InStr(1, Sheets("Application Information").Cells(2, n).Value, "Russian", 1) >
0 Then
                   If Sheets("Application Information").Cells(m, n).Value = "(2) Proficient"
Then
                  Cells(j, k).Interior.ColorIndex = 4
                End If
                           If Sheets("Application Information").Cells(m, n).Value = "(1)
Conversational" Then
                  Cells(j, k).Interior.ColorIndex = 6
                End If
                 If Sheets("Application Information").Cells(m, n).Value = "(0) Non-Speaker"
Then
                  Cells(j, k).Interior.ColorIndex = 15
                End If
                If Sheets("Application Information").Cells(m, n).Value = "" Then
                  Cells(j, k).Interior.ColorIndex = 15
                End If
              End If
              n = n + 1
            Loop
          End If
```

```
                m = m + 1
            Loop
        End If
    Next k
Next j

End Sub
Sub RemoveComments()

'Determine the index of the last row (advisor)
Dim LastRow As Integer
LastRow = 8
Do While Cells(LastRow + 1, 1).Value <> Empty
    LastRow = LastRow + 1
Loop

For j = 9 To LastRow
    For k = 2 To 3
        If Not Range(Cells(j, k), Cells(j, k)).comment Is Nothing Then
            Range(Cells(j, k), Cells(j, k)).comment.Delete
        End If
    Next k
Next j

End Sub
Sub RemoveColoring()

'Determine the index of the last row (advisor)
Dim LastRow As Integer
LastRow = 8
Do While Cells(LastRow + 1, 1).Value <> Empty
    LastRow = LastRow + 1
Loop

'Color block above button green
Range("F1", "K1").ClearFormats
Range("F8", "K8").ClearFormats
Range(Cells(9, 2), Cells(LastRow, 3)).ClearFormats
```

```
End Sub
```

Module 5: Step5ResolveModel

```
Sub ResolveModel()

    'Turn off screen updating, set calculations to manual, and disable events while performing tasks
    Application.ScreenUpdating = False
    Application.Calculation = xlManual
    Application.EnableEvents = False

    'Copy unfilled spots, unmatched advisors, and satisfaction score values from the last solve
    Sheets("Recommended Matchings").Select
    Cells(16, 5).Value = "Last Solve:"
    Cells(16, 6).Value = "Unfilled:"
    Cells(17, 6).Value = "Unmatched:"
    Cells(18, 6).Value = "Satisfaction:"
    Range("B3:C5").Copy
    Range("G16:H18").PasteSpecial Paste:=xlPasteValues
    Range("G16:H18").Interior.ColorIndex = 15
    Range("A8:C500").Select
    Selection.Clear
    Range("A1").Select
    Range("F1", "K1").ClearFormats
    Range("F8", "K8").ClearFormats

    Sheets("Site Matchings").Select

    'Determine the index of the last row (advisor)
    Dim LastRow As Integer
    LastRow = 3
    Do While Cells(LastRow + 1, 1).Value <> Empty
        LastRow = LastRow + 1
    Loop

    'Determine the index of the last column (site-term)
    Dim LastCol As Integer
    LastCol = 4
```

```
Do While Cells(3, LastCol + 1).Value <> Empty
   LastCol = LastCol + 1
Loop

'Determine the index of each term
Dim FirstE As Integer
FirstE = 4
Dim LastE As Integer
LastE = FirstE
Do While InStr(1, Cells(3, LastE + 1).Value, "(E)", 1) > 2
   LastE = LastE + 1
Loop

Dim FirstA As Integer
FirstA = LastE + 1
Dim LastA As Integer
LastA = FirstA
Do While InStr(1, Cells(3, LastA + 1).Value, "(A)", 1) > 2
   LastA = LastA + 1
Loop

Dim FirstB As Integer
FirstB = LastA + 1
Dim LastB As Integer
LastB = FirstB
Do While InStr(1, Cells(3, LastB + 1).Value, "(B)", 1) > 2
   LastB = LastB + 1
Loop

Dim FirstC As Integer
FirstC = LastB + 1
Dim LastC As Integer
LastC = FirstC
Do While InStr(1, Cells(3, LastC + 1).Value, "(C)", 1) > 2
   LastC = LastC + 1
Loop

Dim FirstD As Integer
FirstD = LastC + 1
```

```
    Dim LastD As Integer
    LastD = LastCol

    'Find all cells with Y's and N's and change the solver constraint
    For j = 4 To LastRow
      For k = 4 To LastCol
        If Cells(j, k).Value = "Y" Then
          SolverDelete cellRef:=Cells(j, k), relation:=2, formulatext:=0
          SolverAdd cellRef:=Cells(j, k), relation:=2, formulatext:=1
          Cells(j, k).Value = 1
        End If
        If Cells(j, k).Value = "N" Then
          SolverDelete cellRef:=Cells(j, k), relation:=2, formulatext:=1
          SolverAdd cellRef:=Cells(j, k), relation:=2, formulatext:=0
        End If
      Next k
    Next j

    'Perform OpenSolver
    Dim Result As OpenSolverResult
    Result = RunOpenSolver(False, True) ' do not relax IP, do hide dialogs

    'Highlight all matches (cells containing a 1)
    Range(Cells(4, 4), Cells(LastRow, LastCol)).ClearFormats
    For j = 4 To LastRow
      For k = 4 To LastCol
        If Cells(j, k).Value = 1 Then
          Cells(j, k).Interior.ColorIndex = 43
        End If
      Next k
    Next j

    Sheets("Recommended Matchings").Select

    'Calculate number of unfilled spots, unmatched advisors, and satisfaction score
       Cells(1, 2).Value = "=Sum('Site Matchings'!" & Range(Cells(4, 4), Cells(LastRow,
  LastCol)).Address & ")"
    Range(Cells(1, 2), Cells(2, 2)).Font.Color = vbWhite
    Cells(3, 4).Value = "=Sum('Site Matchings'!" & Range(Cells(2, 4), Cells(2, LastCol)).Address
& ")"
    Cells(3, 3).Value = Cells(3, 4).Value
    Cells(3, 2).Value = Cells(3, 3).Value - Cells(1, 2).Value
    Cells(4, 4).Value = "=CountIf('Site Matchings'!" & Range(Cells(4, 2), Cells(LastRow,
2)).Address & ", " & "0" & ")"
    Cells(4, 3).Value = LastRow - 3
    Cells(4, 2).Value = Cells(4, 4).Value
    Cells(5, 4).Value = "=SumProduct('Site Matchings'!" & Range(Cells(4, 4), Cells(LastRow,
LastCol)).Address & ", 'Site Rankings'!" & Range(Cells(4, 2), Cells(LastRow, LastCol -
2)).Address & ")"
    Cells(5, 3).Value = (Cells(1, 2).Value) * 2
    Cells(5, 2).Value = Cells(5, 4).Value
    Range(Cells(3, 4), Cells(5, 4)).Font.Color = vbWhite

    'Add term and site labels and fill in names of assigned advisors
    j = 8
    Cells(j, 1).Value = "E Term Sites"
    Cells(j, 1).Interior.ColorIndex = 15
    j = j + 1
    For k = 4 To LastE
        Cells(j, 1).Value = Sheets("Site Matchings").Cells(3, k).Value
        For m = 4 To LastRow
            If Sheets("Site Matchings").Cells(m, k).Value <> 0 Then
                If Cells(j, 2).Value <> Empty Then
                    Cells(j, 3).Value = Sheets("Site Matchings").Cells(m, 1)
                End If
                If IsEmpty(Cells(j, 2).Value) Then
                    Cells(j, 2).Value = Sheets("Site Matchings").Cells(m, 1)
                End If
            End If
        Next m
        j = j + 1
    Next k
    Cells(j, 1).Value = "A Term Sites"
    Cells(j, 1).Interior.ColorIndex = 15
    j = j + 1
    For k = FirstA To LastA
        Cells(j, 1).Value = Sheets("Site Matchings").Cells(3, k).Value
        For m = 4 To LastRow
            If Sheets("Site Matchings").Cells(m, k).Value <> 0 Then
                If Sheets("Site Matchings").Cells(m, k).Value <> "N" Then
                    If Cells(j, 2).Value <> Empty Then
                        Cells(j, 3).Value = Sheets("Site Matchings").Cells(m, 1)
                    End If
                    If IsEmpty(Cells(j, 2).Value) Then
                        Cells(j, 2).Value = Sheets("Site Matchings").Cells(m, 1)
                    End If
                End If
            End If
        Next m
        j = j + 1
    Next k
    Cells(j, 1).Value = "B Term Sites"
    Cells(j, 1).Interior.ColorIndex = 15
    j = j + 1
    For k = FirstB To LastB
        Cells(j, 1).Value = Sheets("Site Matchings").Cells(3, k).Value
        For m = 4 To LastRow
            If Sheets("Site Matchings").Cells(m, k).Value <> 0 Then
                If Sheets("Site Matchings").Cells(m, k).Value <> "N" Then
                    If Cells(j, 2).Value <> Empty Then
                        Cells(j, 3).Value = Sheets("Site Matchings").Cells(m, 1)
                    End If
                    If IsEmpty(Cells(j, 2).Value) Then
                        Cells(j, 2).Value = Sheets("Site Matchings").Cells(m, 1)
                    End If
                End If
            End If
        Next m
        j = j + 1
    Next k
    Cells(j, 1).Value = "C Term Sites"
    Cells(j, 1).Interior.ColorIndex = 15
    j = j + 1
    For k = FirstC To LastC
        Cells(j, 1).Value = Sheets("Site Matchings").Cells(3, k).Value
        For m = 4 To LastRow
```

```
            If Sheets("Site Matchings").Cells(m, k).Value <> 0 Then
                If Sheets("Site Matchings").Cells(m, k).Value <> "N" Then
                    If Cells(j, 2).Value <> Empty Then
                        Cells(j, 3).Value = Sheets("Site Matchings").Cells(m, 1)
                    End If
                    If IsEmpty(Cells(j, 2).Value) Then
                        Cells(j, 2).Value = Sheets("Site Matchings").Cells(m, 1)
                    End If
                End If
            End If
        Next m
        j = j + 1
    Next k
    Cells(j, 1).Value = "D Term Sites"
    Cells(j, 1).Interior.ColorIndex = 15
    j = j + 1
    For k = FirstD To LastD
        Cells(j, 1).Value = Sheets("Site Matchings").Cells(3, k).Value
        For m = 4 To LastRow
            If Sheets("Site Matchings").Cells(m, k).Value <> 0 Then
                If Sheets("Site Matchings").Cells(m, k).Value <> "N" Then
                    If Cells(j, 2).Value <> Empty Then
                        Cells(j, 3).Value = Sheets("Site Matchings").Cells(m, 1)
                    End If
                    If IsEmpty(Cells(j, 2).Value) Then
                        Cells(j, 2).Value = Sheets("Site Matchings").Cells(m, 1)
                    End If
                End If
            End If
        Next m
        j = j + 1
    Next k

    'Turn on screen updating, set calculations to automatic, enable events after performing tasks
    Application.ScreenUpdating = True
    Application.Calculation = xlAutomatic
    Application.EnableEvents = True

End Sub
```