THE IMPLICATIONS OF LINEAR PROGRAMMING IN EMPLOYEE WORK SCHEDULING: A CASE STUDY OF THE TEXAS STATE UNIVERSITY-SAN MARCOS STUDENT LEARNING ASSISTANCE CENTER (SLAC) SCHEDULING PRACTICES THESIS

Presented to the Graduate Council of
Texas State University-San Marcos
in Partial Fulfillment
of the Requirements

for the Degree

Master of SCIENCE

by

Holly C. Floyd, B.S.

San Marcos, Texas
May 2010

ACKNOWLEDGEMENTS

I would like to thank my thesis committee chair Dr. Nathaniel Dean and my committee members Dr. Gregory Passty and Dr. Jian Shen for all their guidance. I would also like to thank all of the wonderful teachers I have had the opportunity to learn from at Texas State University-San Marcos. I am especially grateful to Mr. James Mathews and all the wonderful tutors and staff at the Student Learning Assistance Center.

I thank my family. To my mom, Monna, and my dad, James. To April, Sara, and Phoebe. To my Grandma Knebel, my Grandma Dot, and my Grandpa Bob, who always told me that sometimes two plus two doesn't always equal four.

I would like to thank the Grovers for all their love and support, and I would like to especially thank Mandy Grover. You have been a constant encouragement and I will never be able to thank you enough.

This manuscript was submitted on Friday, March 19, 2010.

TABLE OF CONTENTS

Page

ACKNOWLEDGEMENTS iii

CHAPTER

I. INTRODUCTION 1

Scheduling at SLAC 1

Scheduling Problems 2

II. THE PROBLEM 4

Modeling the Scheduling Problem 4

Definition of a Linear Programming Problem 5

The Scheduling Problem Variables 6

The Scheduling Problem Constraints 9

The Scheduling Problem as a Linear Programming Problem 11

III. THE SOLUTION 14

Linear Programming Problems in *Mathematica* 14

Solving the Scheduling Problem using *Mathematica* 16

Scheduling Based on Preferences 17

IV. CONCLUSION 18

Implications 18

APPENDIX A: SLAC AVAILABILITY FORM 20

APPENDIX B: QUESTIONNAIRE 22

APPENDIX C: DATA....26

C.1....27

C.2....28

C.3....29

C.4....30

APPENDIX D: SCHEDULE....41

BIBLIOGRAPHY....43

CHAPTER I

INTRODUCTION

Scheduling at SLAC

In the fall semester of 2008, I began graduate school at Texas State University-San Marcos. At the same time, I also started working on campus at the Student Learning Assistance Center, or SLAC, as a math tutor. As a SLAC employee, I was introduced to the "P's and A's" system of scheduling. In this system, tutors fill out a grid schedule organized by days of the week, which make up the x-axis of the grid, and by time, which runs the y-axis of the grid and begins at 7 a.m. and ends at 1 a.m. (see Appendix A). Tutors, however, only fill in the time slots when SLAC is actually open, which can range from 9 a.m. to 9 p.m. Tutors fill in both the hours they prefer to work (P's) and the hours they are available to work (A's). They put their preferred hours on a poster board that is almost identical to the form, except it only has the hours SLAC is actually open, and then the tutors turn their P's and A's form in. After all the tutors complete this process, the SLAC Lab Coordinator, Mr. James Mathews, takes the information and creates a schedule by hand. In the fall semester of 2009, I became the SLAC Graduate Research Assistant working directly with Mr. Mathews in the process of making the schedule. Mr. Mathews has used this method–minus the poster board, which was just recently implemented–for the past 17 years. This method is not inherently problematic and its long-standing use demonstrates its success. After making two schedules with Mr. Mathews, one for fall 2009 and one for spring 2010, I wanted to investigate creating a more systematic and mathematical approach to scheduling. In addition, I wanted this system to be less time consuming than creating the schedule by hand. I have mathematically modeled this scheduling problem as a linear programming problem.

Scheduling Problems

The problem of scheduling is not a new problem. The need for scheduling is pervasive in almost every aspect of life. According to Michael Pinedo:

> Scheduling is a decision-making process that is used on a regular basis in many manufacturing and service industries. It deals with the allocation of resources to tasks over given time periods and its goal is to optimize one or more objectives. (Pinero 2008, 1)

An example of a very basic scheduling problem would be the job assignment problem: "We have m jobs and n people, but not all people are qualified for all jobs. Can we fill the jobs with qualified people" (West 2001, 4)?

The job assignment problem can be modeled as a graph:

> We model this using a simple graph H with vertices for the jobs and people; job j is adjacent to person p if p can do it. Each job is to be filled by exactly one person, and each person can hold at most one of the jobs. Thus we seek m pairwise disjoint edges in H (viewing edges as pairs of vertices. (West 2001, 4)

If Hall's Condition is satisfied, then we have m pairwise disjoint edges (West 2001). We use the Hungarian Algorithm to find these edges (West 2001). Although the SLAC scheduling problem could be modeled as a graph with vertices for the tutors and slots, each slot does not necessary have to be filled by one person. Hence, this scheduling procedure does not apply to the SLAC scheduling problem. Generally, scheduling problems are much more complicated than this simple example of job assignments.

Scheduling plays a major role at colleges and universities:

> If one is attempting to construct a schedule of courses at a college or university, it is obvious that two courses taught by the same professor cannot be scheduled for the same time slot. Also, two courses that require the same classroom must be scheduled at different times. Furthermore, suppose a particular student or group of students is required by the curriculum to take two different but related courses (e.g., physics and calculus) concurrently during a semester. In this case, these courses also need to be scheduled at different times in order to avoid conflict. The problem of determining the minimum number (or a reasonable number) of time slots needed to schedule all the courses subject to restrictions such as those above is a graph coloring problem. (Redl 2004, 20)

Redl uses applications of graph theory to construct a schedule of courses at a university or college. Unlike the job assignment example, this problem is an optimization problem. It is an optimization problem because he is seeking the minimum number of slots needed to schedule all the courses. Redl also discusses the use of linear programming to solve scheduling problems, which is what we will use to solve the SLAC scheduling problem.

CHAPTER II

THE PROBLEM

Modeling the Scheduling Problem

The problem is how do we create a schedule where during each hour of operation at SLAC we have the desired number of tutors scheduled. For example, if we want three math tutors to be scheduled from 9 a.m. to 10 a.m., then we would check the P's and A's sheets of every math tutor to see who is available for this time and decide which three we want scheduled. However, we have a problem if we do not have enough people available to fill the time slot with the desired quantity of tutors we want scheduled during that time. In this case, we want to get as close to the number of desired tutors for that time slot as possible. Glover and McMillan note that "The objective in the shift scheduling problem generally is to approximate as closely as possible the desired number of employees on duty, either by minimizing the overage or minimizing the 'shortage/overage' mix" (Glover and McMillan 1986, 564). Thus, the goal is to fill all the time slots closest to but not exceeding the number of tutors we desire for each slot. If we scheduled every tutor for every time slot she or he was available, then we would more than likely exceed the number of tutors we desire for each slot. If there are too many instances where we do not have enough tutors to meet our needs, then this would indicate that we would want to hire more tutors so that we could fill those times where we need more tutors. Referring back to the example, if six math tutors are available to work the 9 a.m. to 10 a.m. time slot and we only need three math tutors for that time slot, then we cannot schedule all six people. Therefore, we cannot schedule every tutor for every time she or he is available. Scheduling a tutor for every time she or he is available may also violate Texas State's rules regarding the number of hours a tutor can legally work at the university or result in SLAC exceeding its budget. We also have to consider how many hours each tutor is willing to work per week. For example, we may schedule a tutor for 20 hours per week, but she or he can only work 10 of those hours. Or, tutors may want to seek another place of employment if they did not get as many hours as they wanted after the schedule was posted. If every tutor was scheduled the number of hours the tutor preferred to work, this would ensure that all of our time slots would be filled closest to the desired amount possible. However, it is still very unlikely that we would be able to schedule every tutor the number of hours the tutor prefers to work per week. Hence, we want to minimize the difference between the number of hours each tutor wants to work and the number of hours each tutor is scheduled to work. The best or optimal solution is when this difference is zero, because then every tutor is scheduled to work exactly the number of hours she or he prefers to work. More importantly, if this number is zero it guarantees that the slots are filled as close to the desired amount as possible.

Anytime we try to minimize a quantity, it is an optimization problem. Linear programming problems are a type of optimization problem. Thus, we will model the scheduling problem as a linear programming problem.

Definition of a Linear Programming Problem

According to Smythe and Johnson, the definition of a linear programming is:

Given real numbers $c_1, c_2, \ldots, c_n,\ b_1, b_2, \ldots, b_m$, and a_{ij} ($i = 1, \ldots, m;\ j = 1, \ldots, n$) we wish to maximize (or minimize) the function z of n real variables defined by

$$z(x_1, x_2, \ldots, x_n) = c_1x_1 + c_2x_2 + \cdots + c_nx_n$$

subject to the conditions

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n\ (\leq, =, \geq)\ b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n\ (\leq, =, \geq)\ b_2 \\ \cdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n\ (\leq, =, \geq)\ b_m, \end{cases} \tag{1}$$

where, in each line of (1), exactly one of the symbols $\leq$, $=$, and, $\geq$ appears. The function z is called the *objective function.* Conditions (1) are referred to as the *constraints.* A *feasible solution* of the problem is an n-dimensional column vector $\mathbf{x} = [x_1, x_2, \ldots, x_n]$ whose components satisfy all of constraints (1). An *optimal* solution of the problem is a feasible solution for which the corresponding value of the objective function is a maximum (in the case of a maximizing problem) or a minimum (in the case of a minimizing problem), considering all possible values of the objective function corresponding to feasible solutions. By the *solution* of the problem, we shall mean the determination of an optimal solution and the corresponding value of the objective function, or the proof that no feasible solutions exist, or the proof that, although feasible solutions exist, there is no optimal solution. (Smythe and Johnson 1966, 67-68)

It is important to note that the objective function and the constraints are linear.

The Scheduling Problem Variables

We will now define each variable in our problem. Let n represent the number of tutors and

$p_i =$ the number of slots tutor i would prefer to work per week, where $i = 1, 2, ..., n$ and slots refers to each hour and some half hours SLAC is open.

Each p_i is data. Let

$$a_{i,j} = \begin{cases} 1 & \text{if tutor } i \text{ is available for slot } j \\ 0 & \text{otherwise.} \end{cases}$$

Each $a_{i,j}$ is data. Let m represent the total number of slots and

$$x_{i,j} = \begin{cases} 1 & \text{if tutor } i \text{ is assigned to slot } j \\ 0 & \text{otherwise,} \end{cases}$$

where $i = 1, 2, ..., n$ and $j = 1, 2, ...m$.

The $x_{i,j}$'s are variables that we are solving for in the problem. The $x_{i,j}$'s that equal one tell us who works each slot which in turn gives us the tutor schedule. Let

$$s_i = \sum_{j=1}^{m} a_{i,j} x_{i,j}.$$

The s_i variable is the sum of the total number of slots tutor i is assigned per week. Let

$$u_i = \text{the minimum number of slots a tutor can work per week.}$$

The variable u_i is data. A tutor must work at least one slot per week. Let

v_i = the maximum number of slots a tutor is allowed to work per week.

The variable v_i is data. It is university policy that a student worker cannot work more than twenty hours per week for a department. There are three types of tutors at SLAC: business, verbal, and quantitative. Let

e_j = the maximum number of business tutors are allowed to work in slot j,

f_j = the maximum number of verbal tutors are allowed to work in slot j, and

g_j = the maximum number of quantitative tutors are allowed to work in slot j.

The variables e_j, f_j, and g_j are data based on peak times and the budget established by SLAC. Let

$$d_i = \mid s_i - p_i \mid .$$

The variable d_i is the difference between the number of slots tutor i is assigned to work per week and the number of slots tutor i prefers to work per week. Let

$$z = \sum_{i=1}^{n} d_i.$$

The variable z tells us the sum of the differences between the number of slots tutor i is assigned to work per week and the number of slots tutor i prefers to work per week for all i. Because we want all of the slots to be filled to capacity, we want z to be as small as possible. Hence, z is the variable we want to minimize. Since $d_i \geq 0$ for all i, then $z \geq 0$.

The Scheduling Problem Constraints

Now, we are ready to formulate the constraints. Recall from the definition of a linear programming problem that we must have a real number on the right hand side of the relation symbol. Thus, from the definition of z we have the following constraint:

$$\sum_{i=1}^{n} d_i - z = 0.$$

From the definition of d_i we have the following constraint:

$$d_i - | s_i - p_i | = 0.$$

Notice that this equation is not linear. Recall from the definition of a linear progamming problem that the constraints must be linear equations. However, we can transform this equation into a set of linear equations using the definition of absolute value:

$$d_i + s_i - p_i \geq 0 \text{ and } d_i - s_i + p_i \geq 0.$$

Since for each i, p_i is data we have

$$d_i + s_i \geq p_i \text{ and } d_i - s_i \geq -p_i.$$

From our definition of $x_{i,j}$ we have the following constraint:

$$x_{i,j} = \begin{cases} 1 & \text{if tutor } i \text{ is assigned to slot } j \\ 0 & \text{otherwise.} \end{cases}$$

Notice that this equation is not linear. However, we can transform this constraint into a set of linear inequalities:

$$x_{i,j} \geq 0 \text{ and } x_{i,j} \leq 1, \text{ where } x_{i,j} \text{ is an integer.}$$

From the definition of s_i we have the following constraint:

$$\sum_{j=1}^{m} a_{i,j} x_{i,j} - s_i = 0.$$

Since there is a minimum and a maximum number or slots a tutor can work per week, then we have the following constraints:

$$s_i \geq u_i \text{ and } s_i \leq v_i.$$

Since there is maximum number of business tutors that can work each slot, then we have the following constraint:

$$\sum_{i=1}^{c} a_{i,j} x_{i,j} \leq e_j, \text{ where } c \text{ is the number of business tutors.}$$

Since there is maximum number of verbal tutors that can work each slot, then we have the following constraint:

$$\sum_{i=1}^{l} a_{i,j} x_{i,j} \leq f_j,$$ where l is the number of verbal tutors.

Since there is maximum number of quantitative tutors that can work each slot, then we have the following constraint:

$$\sum_{i=1}^{r} a_{i,j} x_{i,j} \leq g_j,$$ where r is the number of quantitative tutors.

We have listed all the necessary constraints we need for the problem.

The Scheduling Problem as a Linear Programming Problem

Since we want to minimize z we have the objective function t, where

$$t(z, d_1, d_2, \ldots, d_n, s_1, s_2, \ldots, s_n, x_{1,1}, x_{1,1}, \ldots, x_{n,m}) = z$$

subject to the constraints:

$$\begin{cases} -z + d_1 + d_2 + \cdots + d_n = 0 & \\ d_i + s_i \geq p_i & i = 1, 2, \ldots, n \\ d_i - s_i \geq -p_i & i = 1, 2, \ldots, n \\ -s_i + a_{i,1}x_{i,1} + a_{i,2}x_{i,2} + \cdots + a_{i,m}x_{i,m} = 0 & i = 1, 2, \ldots, n \\ a_{1,j}x_{1.j} + a_{2,j}x_{2,j} + \cdots + a_{c,j}x_{c,j} \leq e_j & j = 1, 2, \ldots, m \\ a_{1,j}x_{1.j} + a_{2,j}x_{2,j} + \cdots + a_{l,j}x_{l,j} \leq f_j & j = 1, 2, \ldots, m \\ a_{1,j}x_{1.j} + a_{2,j}x_{2,j} + \cdots + a_{r,j}x_{r,j} \leq g_j & j = 1, 2, \ldots, m \\ s_i \geq u_i & i = 1, 2, \ldots, n \\ s_i \leq v_i & i = 1, 2, \ldots, n \\ x_{i,j} \geq 0 & i = 1, 2, \ldots, n \text{ and } j = 1, 2, \ldots, m \\ x_{i,j} \leq 1 & i = 1, 2, \ldots, n \text{ and } j = 1, 2, \ldots, m. \end{cases}$$

The optimal solution is the column vector

$$\mathbf{x} = [z, d_1, d_2, \ldots, d_n, s_1, s_2, \ldots, s_n, x_{1,1}, x_{1,2}, \ldots, x_{n,m}]'$$

where the value of z is minimum.

To gather the necessary data for the problem, an anonymous online questionnaire was created at www.surveymonkey.com (see Appendix B). At the time when the questionnaire was posted, there was a total of 40 tutors working for SLAC. Out of the 40 tutors, 21 completed the questionnaire. We will only be using the data from the 21 tutors who completed the survey. Out of the 21 tutors who completed the survey there are four business tutors, six verbal tutors, and 11 quantitative tutors. SLAC's hours of operation amounts to a total of 51 slots. The slot list, tutor list, and all data used in this problem can be found in Appendix C. All of the data and variables are non-negative integers. Since all of the variables are integers, then the scheduling problem is an integer linear programming problem.

CHAPTER III

THE SOLUTION

Linear Programming Problems in *Mathematica*

We will use *Mathematica* software to solve our scheduling problem. More specifically, we will be using *Mathematica*'s linear programming function to solve our problem. *Mathematica* by default will choose automatically from the simplex, revised simplex, or interior point method to solve the linear programming problem based on problem size and precision.

The linear programming function in *Mathematica* is defined in several different ways. For this problem we will be using the following definition for *Mathematica*'s linear programming function :

$$\texttt{LinearProgramming}[c, m, \{\{b_1, s_1\}, \{b_2, s_2\}, \ldots\}, \{\{l_1, u_1\}, \{l_2, u_2\}, \ldots\}],$$

where using the previous definition of a linear programming problem

$$c = \{c_1, c_2, \ldots, c_n\},$$

$$m = \begin{bmatrix} a_{1,1} & a_{1,2} & \cdots & a_{1,n} \\ a_{2,1} & a_{2,2} & \cdots & a_{2,n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m,1} & a_{m,2} & \cdots & a_{m,n} \end{bmatrix},$$

and

$$\{b_1, b_2, \ldots, b_n\} = \begin{bmatrix} b_1 \\ b_2 \\ \vdots \\ b_n \end{bmatrix}.$$

This command finds a column vector $x = [x_1, x_2, \ldots, x_n]$ which minimizes *cx* subject to $x \geq 0$, the linear constraints specified by the matrix *m*, the pairs $\{b_i, s_i\}$, and $l_i \leq x_i \leq u_i$. For each row m_i of *m*, the corresponding constraint is $m_i x \geq b_i$ if $s_i = 1$, $m_i x = b_i$ if $s_i = 0$, or $m_i x \leq b_i$ if $s_i = -1$. In this problem, *c* has 1114 terms, and *m* is a matrix with 217 rows and 1114 columns. The $\{b_i, s_i\}$ pairs in this problem form a matrix with 217 rows and 2 columns. And the $\{l_i, u_i\}$ pairs in this problem form a matrix with 1114 rows and 2 columns. The column vector *x* in this problem has 1114 terms. Since we have such a large-scale problem, we will input the values for *c*, *m*, $\{\{b_1, s_1\}, \{b_2, s_2\}, \ldots\}$, and $\{\{l_1, u_1\}, \{l_2, u_2\}, \ldots\}$ into Microsoft *Excel*. We will import each of *c*, *m*, $\{\{b_1, s_1\}, \{b_2, s_2\}, \ldots\}$, and $\{\{l_1, u_1\}, \{l_2, u_2\}, \ldots\}$ from *Excel* into *Mathematica* using the import function:

`Import["`*name.ext*`"]`.

This command imports data assuming a format deduced from the file name.

We will export the solution x from *Mathematica* into *Excel* using the export function:

Export["*name.ext*",*expr*].

This command exports data in a format deduced from the file name.

<u>Solving the Scheduling Problem using *Mathematica*</u>

To solve the scheduling problem we input the following into *Mathematica*:

```
Export["Documents/x_solution.xls",
LinearProgramming[Flatten
[Import["Documents/c_coefficients.csv]],
Import["Documents/m_constraints.csv"],
Import["Documents/b_s_constraints.csv"],
Import["Documents/l_u_constraints.csv"]]].
```

We have to use the `Flatten` command when we import c to remove the extra set of brackets that results from importing files. Each imported file uses the file extension .csv which is an acronym for "comma-separated values" because we need the imported values to be separated by commas. If we have an optimal solution we have the following output from *Mathematica*:

```
Documents/x_solution.xls.
```

Mathematica solved the scheduling problem in 52.274474 seconds with $z = 48$. Recall that the solution to the scheduling problem is a column vector with 1114 rows. The solution x is stored in rows 1-1114 of column A in the file "x_solution.xls". We can create a table in *Excel* to display the resulting tutor schedule (see Appendix D). The linear programming function in *Mathematica* by default assumes all variables are real numbers. Recall that all of the variables in the problem are integers. Even though we did not specify that every variable be an integer, every variable in the solution that *Mathematica* returned for the problem was an integer.

Scheduling Based on Preferences

An exciting aspect about this method is that by changing the constraints we can easily and quickly explore certain scheduling preferences. For example, we may want the schedule to be fair in the sense that each tutor is scheduled within two time slots of the tutor's preferred number of time slots. So, instead of $d_i \geq 0$ we would have the constraint $0 \leq d_i \leq 2$ for each i. It is important to note that there may not be a feasible solution that satisfies this constraint. Another scheduling preference may be for the tutors who have work study to be scheduled all the hours they want to work. The tutors who have work study are not paid out of SLAC's budget so it saves SLAC money when the work-study tutors work as many hours as possible. For example, suppose tutor 4 has work study. So, instead of $d_4 \geq 0$ we have $d_4 = 0$. Again, we may have no feasible solution that satisfies this constraint. As long as $z \geq 0$, and $d_i \geq 0$ for all i, then we will have a feasible solution. Since each tutor must work one time slot per week, $d_i \geq 1$ for all i. With this constraint we risk having no feasible solution. However, it is very unlikely that we cannot schedule every tutor at least one time slot.

CHAPTER IV

CONCLUSION

Implications

We accomplished one goal of the scheduling problem by finding a mathematical approach at creating a schedule. Another goal of this project was the ability to create a schedule faster than creating a schedule by hand. Although *Mathematica* solved our problem in less than a minute, we have not considered how much time it takes to input the data. Recall that we input c, m, $\{\{b_1, s_1\}, \{b_2, s_2\}, \ldots\}$, and $\{\{l_1, u_1\}, \{l_2, u_2\}, \ldots\}$ into Microsoft *Excel*. Since we have such a large-scale problem it would take a considerable amount of time to enter all the data cell by cell. However, we can use tools in *Excel* so that we do not have to input the data cell by cell. The only variables that will change from semester to semester are the number of tutors, each tutor's availibility, and the number of slots each tutor prefers to work. Using tools in *Excel* we can input the availability for each tutor one place and have it linked to all the cells in m where the availibility values are located. This is the bulk of the input data. Ideally, we do not want to spend any time on inputting data. This project could be taken further by designing a computer program where instead of tutors turning in a P's and A's sheet or writing on a poster, they would input their availibility and preferred number of hours into a computer wherein the data would automatically be stored in the corresponding cells in *Excel*. Using linear programming to create the SLAC tutor schedule has the possibilities of saving time, and the option to easily and quickly explore possible schedules based on preferences.

APPENDIX A

SLAC AVAILABILITY FORM

Currently, each tutor employed by SLAC fills out the following form. Tutors fill in both the hours they prefer to work (P's) and the hours they are available to work (A's). The SLAC Lab Coordinator creates a schedule based on this information.

SLAC Daily Schedule

	MON	TUES	WED	THURS	FRI	SAT	SUN
7-8							
8-9							
9-10							
10-11							
11-12							
12-1							
1-2							
2-3							
3-4							
4-5							
5-6							
6-7							
7-8							
8-9							
9-10							
10-11							
11-12							
12-1							

APPENDIX B

QUESTIONNAIRE

This is the online questionnaire that was created at www.surveymonkey.com. Tutors were emailed a link to this questionnaire, and asked to complete this questionnaire. Questions one, two, and five could only be answered one way. For questions three, four, and six, tutors could check multiple boxes.

SLAC Scheduling Questionnaire

Exit this survey

1. Default Section

*** 1. What is your job title?**

Business tutor

Quantitative tutor

Verbal tutor

*** 2. Do you have work study?**

Yes

No

*** 3. Please check the time slots that you are AVAILABLE to work.**
These are the times SLAC is open:
M-W 9am-8:30pm; Th 9am-4:30pm; F 9am-12pm; Sun 5pm-9pm.

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
9am-10am							
10am-11am							
11am-12pm							
12pm-1pm							
1pm-2pm							
2pm-3pm							
3pm-4pm							
4pm-5pm							
5pm-6pm							
6pm-7pm							
7pm-8pm							
8pm-9pm							

4. Please check the time slots that you PREFER to work.

	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
9am-10am							
10am-11am							

11am-12pm							
12pm-1pm							
1pm-2pm							
2pm-3pm							
3pm-4pm							
4pm-5pm							
5pm-6pm							
6pm-7pm							
7pm-8pm							
8pm-9pm							

***5. How many hours per week would you like to work?**

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18

19

20

***6. Please check the courses that you know how to tutor?**

Please check all that apply.

Accounting

Biology

Chemistry

CIS

Economics

Finance

History

Math 2358 (Discrete Math I)

Math 3398 (Discrete Math II)

Math 2472 (Calculus II)

Philosophy

Physics

Political Science

Spanish

Statistics

Thank you very much for your participation!

Done

APPENDIX C

DATA

The data used in this study are listed in this appendix as follows:

C.1: This table is a list of the 51 slot assignments at SLAC. For example, $j = 1$ represents the time slot from 9 a.m. - 10 a.m. on Monday.

C.2: This table is list of tutor assignments and the number of hours each tutor prefers to work per week assignments. For example, $i = 1$ represents business tutor one, and $p_1 = 9$ represents the preferred number of hours per week for business tutor one. This data was gathered from the questionnaire.

C.3: This table lists the maximum number of tutors that can be scheduled for each time slot. Column e_i, f_i, and g_i represent the maximum number of business, verbal, and quantitative tutors respectively that can work the ith time slot. For example, $e_1 = 1$ means we can schedule at maximum one business tutor for slot one.

C.4: This table represents all the availabilities for each tutor. For example, $a_{i,j} = 1$ means that tutor i is available to work slot j. If $a_{i,j} = 0$, then tutor i is not available to work slot j. This data was gathered from the questionnaire.

C.1

j		j		j	
1	Monday, 9 a.m. -10 a.m.	22	Tuesday, 6 p.m. -7 p.m.	43	Thursday, 3 p.m. -4 p.m.
2	Monday, 10 a.m. -11 a.m.	23	Tuesday, 7 p.m. -8 p.m.	44	Thursday, 4 p.m. -4:30 p.m.
3	Monday, 11 a.m. -12 p.m.	24	Tuesday, 8 p.m. -8:30 p.m.	45	Friday, 9 a.m. -10 a.m.
4	Monday, 12 p.m. -1 p.m.	25	Wednesday, 9 a.m. -10 a.m.	46	Friday, 10 a.m. -11 a.m.
5	Monday, 1 p.m. -2 p.m.	26	Wednesday, 10 a.m. -11 a.m.	47	Friday, 11 a.m. -12 p.m.
6	Monday, 2 p.m. -3 p.m.	27	Wednesday, 11 a.m. -12 p.m.	48	Sunday, 5 p.m. - 6 p.m.
7	Monday, 3 p.m. -4 p.m.	28	Wednesday, 12 p.m. -1 p.m.	49	Sunday, 6 p.m. - 7 p.m.
8	Monday, 4 p.m. -5 p.m.	29	Wednesday, 1 p.m. -2 p.m.	50	Sunday, 7 p.m. - 8 p.m.
9	Monday, 5 p.m. -6 p.m.	30	Wednesday, 2 p.m. -3 p.m.	51	Sunday, 8 p.m. - 9 p.m.
10	Monday, 6 p.m. -7 p.m.	31	Wednesday, 3 p.m. -4 p.m.		
11	Monday, 7 p.m. -8 p.m.	32	Wednesday, 4 p.m. -5 p.m.		
12	Monday, 8 p.m. -8:30 p.m.	33	Wednesday, 5 p.m. -6 p.m.		
13	Tuesday, 9 a.m. -10 a.m.	34	Wednesday, 6 p.m. -7 p.m.		
14	Tuesday, 10 a.m. -11 a.m.	35	Wednesday, 7 p.m. -8 p.m.		
15	Tuesday, 11 a.m. -12 p.m.	36	Wednesday, 8 p.m. -8:30 p.m.		
16	Tuesday, 12 p.m. -1 p.m.	37	Thursday, 9 a.m. -10 a.m.		
17	Tuesday, 1 p.m. -2 p.m.	38	Thursday, 10 a.m. -11 a.m.		
18	Tuesday, 2 p.m. -3 p.m.	39	Thursday, 11 a.m. -12 p.m.		
19	Tuesday, 3 p.m. -4 p.m.	40	Thursday, 12 p.m. -1 p.m.		
20	Tuesday, 4 p.m. -5 p.m.	41	Thursday, 1 p.m. -2 p.m.		
21	Tuesday, 5 p.m. -6 p.m.	42	Thursday, 2 p.m. -3 p.m.		

C.2

i		i	p_i
1	business tutor 1	1	9
2	business tutor 2	2	15
3	business tutor 3	3	17
4	business tutor 4	4	20
5	verbal tutor 1	5	20
6	verbal tutor 2	6	20
7	verbal tutor 3	7	14
8	verbal tutor 4	8	10
9	verbal tutor 5	9	15
10	verbal tutor 6	10	20
11	quantitative tutor 1	11	13
12	quantitative tutor 2	12	15
13	quantitative tutor 3	13	18
14	quantitative tutor 4	14	15
15	quantitative tutor 5	15	10
16	quantitative tutor 6	16	15
17	quantitative tutor 7	17	20
18	quantitative tutor 8	18	8
19	quantitative tutor 9	19	10
20	quantitative tutor 10	20	15
21	quantitative tutor 11	21	12

C.3

i	e_i	i	e_i	i	e_i	i	f_i	i	f_i	i	f_i	i	g_i	i	g_i	i	g_i
1	1	22	1	43	2	1	1	22	1	43	2	1	3	22	5	43	5
2	1	23	1	44	2	2	1	23	1	44	2	2	3	23	4	44	5
3	2	24	1	45	1	3	2	24	1	45	1	3	5	24	4	45	3
4	2	25	1	46	1	4	2	25	1	46	1	4	5	25	3	46	3
5	2	26	1	47	1	5	2	26	1	47	1	5	5	26	3	47	3
6	2	27	2	48	1	6	2	27	2	48	2	6	5	27	5	48	5
7	2	28	2	49	1	7	2	28	2	49	2	7	5	28	5	49	5
8	2	29	2	50	1	8	2	29	2	50	2	8	5	29	5	50	5
9	1	30	2	51	1	9	1	30	2	51	2	9	5	30	5	51	5
10	1	31	2			10	1	31	2			10	5	31	5		
11	1	32	2			11	1	32	2			11	4	32	5		
12	1	33	1			12	1	33	1			12	4	33	5		
13	1	34	1			13	1	34	1			13	3	34	5		
14	1	35	1			14	1	35	1			14	3	35	4		
15	2	36	1			15	2	36	1			15	5	36	4		
16	2	37	1			16	2	37	1			16	5	37	3		
17	2	38	1			17	2	38	1			17	5	38	3		
18	2	39	2			18	2	39	2			18	5	39	5		
19	2	40	2			19	2	40	2			19	5	40	5		
20	2	41	2			20	2	41	2			20	5	41	5		
21	1	42	2			21	1	42	2			21	5	42	5		

C.4

i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$
1,1	1	1,22	1	1,43	1	2,1	0	2,22	1	2,43	0
1,2	1	1,23	1	1,44	1	2,2	0	2,23	1	2,44	0
1,3	1	1,24	1	1,45	1	2,3	0	2,24	1	2,45	0
1,4	0	1,25	1	1,46	1	2,4	1	2,25	0	2,46	0
1,5	0	1,26	1	1,47	1	2,5	1	2,26	0	2,47	0
1,6	0	1,27	1	1,48	1	2,6	1	2,27	0	2,48	1
1,7	0	1,28	0	1,49	1	2,7	0	2,28	1	2,49	1
1,8	1	1,29	0	1,50	1	2,8	0	2,29	1	2,50	0
1,9	1	1,30	0	1,51	1	2,9	0	2,30	1	2,51	0
1,10	1	1,31	0			2,10	1	2,31	0		
1,11	1	1,32	1			2,11	1	2,32	0		
1,12	1	1,33	1			2,12	1	2,33	0		
1,13	0	1,34	1			2,13	0	2,34	0		
1,14	0	1,35	1			2,14	0	2,35	0		
1,15	0	1,36	1			2,15	0	2,36	0		
1,16	0	1,37	0			2,16	0	2,37	0		
1,17	0	1,38	0			2,17	0	2,38	0		
1,18	1	1,39	0			2,18	0	2,39	0		
1,19	1	1,40	0			2,19	0	2,40	0		
1,20	1	1,41	0			2,20	0	2,41	0		
1,21	1	1,42	1			2,21	0	2,42	0		

i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$
3,1	1	3,22	1	3,43	1	4,1	0	4,22	1	4,43	1
3,2	1	3,23	1	3,44	1	4,2	0	4,23	1	4,44	1
3,3	1	3,24	1	3,45	1	4,3	0	4,24	1	4,45	0
3,4	1	3,25	1	3,46	1	4,4	0	4,25	0	4,46	0
3,5	1	3,26	1	3,47	1	4,5	0	4,26	0	4,47	0
3,6	0	3,27	1	3,48	1	4,6	0	4,27	0	4,48	0
3,7	0	3,28	1	3,49	1	4,7	0	4,28	0	4,49	0
3,8	0	3,29	1	3,50	1	4,8	0	4,29	0	4,50	0
3,9	1	3,30	0	3,51	1	4,9	0	4,30	0	4,51	0
3,10	1	3,31	0			4,10	0	4,31	0		
3,11	1	3,32	0			4,11	0	4,32	0		
3,12	1	3,33	1			4,12	0	4,33	1		
3,13	0	3,34	1			4,13	0	4,34	1		
3,14	1	3,35	1			4,14	0	4,35	1		
3,15	1	3,36	1			4,15	0	4,36	1		
3,16	1	3,37	0			4,16	0	4,37	0		
3,17	0	3,38	1			4,17	0	4,38	0		
3,18	0	3,39	1			4,18	1	4,39	0		
3,19	0	3,40	1			4,19	1	4,40	0		
3,20	0	3,41	0			4,20	1	4,41	0		
3,21	0	3,42	0			4,21	1	4,42	1		

i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$
5,1	1	5,22	1	5,43	0	6,1	1	6,22	0	6,43	1
5,2	1	5,23	1	5,44	0	6,2	1	6,23	0	6,44	1
5,3	1	5,24	1	5,45	1	6,3	0	6,24	0	6,45	0
5,4	0	5,25	1	5,46	1	6,4	0	6,25	0	6,46	0
5,5	0	5,26	1	5,47	1	6,5	0	6,26	0	6,47	0
5,6	0	5,27	1	5,48	0	6,6	0	6,27	0	6,48	1
5,7	0	5,28	0	5,49	0	6,7	0	6,28	0	6,49	1
5,8	0	5,29	0	5,50	0	6,8	0	6,29	0	6,50	1
5,9	1	5,30	0	5,51	0	6,9	1	6,30	0	6,51	1
5,10	1	5,31	0			6,10	1	6,31	0		
5,11	1	5,32	0			6,11	1	6,32	0		
5,12	1	5,33	1			6,12	1	6,33	1		
5,13	1	5,34	1			6,13	0	6,34	1		
5,14	1	5,35	1			6,14	0	6,35	1		
5,15	1	5,36	1			6,15	0	6,36	1		
5,16	0	5,37	1			6,16	0	6,37	0		
5,17	0	5,38	1			6,17	0	6,38	0		
5,18	0	5,39	1			6,18	0	6,39	0		
5,19	0	5,40	0			6,19	0	6,40	0		
5,20	0	5,41	0			6,20	0	6,41	0		
5,21	1	5,42	0			6,21	0	6,42	0		

i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$
7,1	0	7,22	0	7,43	0	8,1	1	8,22	0	8,43	0
7,2	0	7,23	0	7,44	1	8,2	1	8,23	0	8,44	0
7,3	0	7,24	1	7,45	1	8,3	1	8,24	0	8,45	1
7,4	0	7,25	0	7,46	1	8,4	1	8,25	1	8,46	0
7,5	0	7,26	0	7,47	1	8,5	0	8,26	1	8,47	0
7,6	0	7,27	0	7,48	0	8,6	0	8,27	1	8,48	1
7,7	1	7,28	0	7,49	0	8,7	0	8,28	1	8,49	1
7,8	1	7,29	0	7,50	0	8,8	0	8,29	0	8,50	0
7,9	1	7,30	0	7,51	0	8,9	0	8,30	0	8,51	0
7,10	1	7,31	0			8,10	0	8,31	0		
7,11	1	7,32	0			8,11	0	8,32	0		
7,12	1	7,33	1			8,12	0	8,33	0		
7,13	0	7,34	1			8,13	0	8,34	0		
7,14	0	7,35	1			8,14	0	8,35	0		
7,15	0	7,36	1			8,15	1	8,36	0		
7,16	0	7,37	0			8,16	0	8,37	0		
7,17	0	7,38	0			8,17	0	8,38	0		
7,18	0	7,39	0			8,18	1	8,39	1		
7,19	0	7,40	0			8,19	1	8,40	0		
7,20	0	7,41	0			8,20	1	8,41	0		
7,21	0	7,42	0			8,21	0	8,42	1		

i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$
9,1	1	9,22	1	9,43	0	10,1	1	10,22	1	10,43	1
9,2	1	9,23	1	9,44	0	10,2	1	10,23	1	10,44	1
9,3	1	9,24	1	9,45	1	10,3	0	10,24	1	10,45	1
9,4	0	9,25	1	9,46	1	10,4	0	10,25	1	10,46	1
9,5	0	9,26	1	9,47	1	10,5	0	10,26	1	10,47	1
9,6	0	9,27	1	9,48	0	10,6	0	10,27	0	10,48	0
9,7	0	9,28	0	9,49	0	10,7	1	10,28	0	10,49	0
9,8	1	9,29	0	9,50	0	10,8	1	10,29	0	10,50	0
9,9	1	9,30	0	9,51	0	10,9	1	10,30	0	10,51	0
9,10	1	9,31	0			10,10	1	10,31	1		
9,11	1	9,32	1			10,11	1	10,32	1		
9,12	0	9,33	1			10,12	1	10,33	1		
9,13	1	9,34	1			10,13	0	10,34	1		
9,14	1	9,35	0			10,14	0	10,35	1		
9,15	1	9,36	0			10,15	0	10,36	1		
9,16	1	9,37	1			10,16	0	10,37	1		
9,17	1	9,38	1			10,17	0	10,38	1		
9,18	1	9,39	1			10,18	0	10,39	0		
9,19	0	9,40	1			10,19	1	10,40	0		
9,20	0	9,41	1			10,20	1	10,41	1		
9,21	1	9,42	1			10,21	1	10,42	1		

i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$
11,1	0	11,22	1	11,43	0	12,1	0	12,22	0	12,43	1
11,2	0	11,23	1	11,44	0	12,2	0	12,23	0	12,44	0
11,3	0	11,24	1	11,45	0	12,3	1	12,24	0	12,45	0
11,4	0	11,25	0	11,46	0	12,4	1	12,25	0	12,46	0
11,5	0	11,26	0	11,47	0	12,5	1	12,26	0	12,47	0
11,6	0	11,27	0	11,48	0	12,6	0	12,27	1	12,48	0
11,7	0	11,28	0	11,49	1	12,7	0	12,28	1	12,49	0
11,8	0	11,29	0	11,50	1	12,8	1	12,29	1	12,50	0
11,9	0	11,30	0	11,51	1	12,9	1	12,30	0	12,51	0
11,10	1	11,31	0			12,10	1	12,31	0		
11,11	1	11,32	0			12,11	1	12,32	0		
11,12	1	11,33	0			12,12	1	12,33	0		
11,13	0	11,34	1			12,13	0	12,34	0		
11,14	0	11,35	1			12,14	0	12,35	0		
11,15	0	11,36	1			12,15	0	12,36	0		
11,16	0	11,37	0			12,16	1	12,37	0		
11,17	0	11,38	0			12,17	1	12,38	0		
11,18	0	11,39	0			12,18	1	12,39	0		
11,19	0	11,40	0			12,19	1	12,40	1		
11,20	0	11,41	0			12,20	0	12,41	1		
11,21	0	11,42	0			12,21	0	12,42	1		

i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$
13,1	1	13,22	1	13,43	0	14,1	1	14,22	0	14,43	1
13,2	1	13,23	1	13,44	0	14,2	1	14,23	1	14,44	1
13,3	1	13,24	1	13,45	1	14,3	1	14,24	1	14,45	1
13,4	1	13,25	1	13,46	1	14,4	1	14,25	1	14,46	1
13,5	1	13,26	1	13,47	1	14,5	0	14,26	1	14,47	1
13,6	0	13,27	0	13,48	1	14,6	0	14,27	1	14,48	1
13,7	0	13,28	0	13,49	1	14,7	0	14,28	1	14,49	1
13,8	0	13,29	0	13,50	1	14,8	1	14,29	0	14,50	1
13,9	0	13,30	0	13,51	1	14,9	1	14,30	0	14,51	1
13,10	0	13,31	0			14,10	1	14,31	0		
13,11	0	13,32	1			14,11	1	14,32	1		
13,12	0	13,33	1			14,12	1	14,33	1		
13,13	0	13,34	1			14,13	1	14,34	1		
13,14	0	13,35	1			14,14	1	14,35	1		
13,15	0	13,36	1			14,15	1	14,36	1		
13,16	0	13,37	0			14,16	1	14,37	1		
13,17	0	13,38	0			14,17	1	14,38	1		
13,18	0	13,39	0			14,18	0	14,39	1		
13,19	0	13,40	0			14,19	0	14,40	1		
13,20	0	13,41	0			14,20	0	14,41	1		
13,21	1	13,42	0			14,21	0	14,42	1		

i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$
15,1	1	15,22	1	15,43	0	16,1	0	16,22	1	16,43	1
15,2	1	15,23	1	15,44	0	16,2	0	16,23	1	16,44	1
15,3	0	15,24	1	15,45	1	16,3	0	16,24	1	16,45	0
15,4	0	15,25	1	15,46	1	16,4	0	16,25	0	16,46	0
15,5	0	15,26	1	15,47	1	16,5	0	16,26	0	16,47	0
15,6	0	15,27	0	15,48	1	16,6	1	16,27	0	16,48	1
15,7	1	15,28	0	15,49	1	16,7	0	16,28	0	16,49	1
15,8	1	15,29	0	15,50	1	16,8	0	16,29	0	16,50	1
15,9	1	15,30	0	15,51	1	16,9	0	16,30	1	16,51	1
15,10	1	15,31	1			16,10	0	16,31	0		
15,11	1	15,32	1			16,11	0	16,32	0		
15,12	1	15,33	1			16,12	0	16,33	0		
15,13	0	15,34	1			16,13	1	16,34	0		
15,14	0	15,35	1			16,14	1	16,35	0		
15,15	1	15,36	1			16,15	1	16,36	0		
15,16	0	15,37	0			16,16	1	16,37	1		
15,17	0	15,38	0			16,17	1	16,38	1		
15,18	0	15,39	1			16,18	1	16,39	1		
15,19	0	15,40	0			16,19	1	16,40	1		
15,20	0	15,41	0			16,20	1	16,41	1		
15,21	1	15,42	0			16,21	1	16,42	1		

i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$
17,1	1	17,22	1	17,43	1	18,1	0	18,22	1	18,43	1
17,2	1	17,23	1	17,44	1	18,2	0	18,23	1	18,44	0
17,3	1	17,24	1	17,45	1	18,3	0	18,24	1	18,45	0
17,4	0	17,25	1	17,46	1	18,4	1	18,25	0	18,46	0
17,5	0	17,26	1	17,47	1	18,5	1	18,26	0	18,47	0
17,6	0	17,27	1	17,48	1	18,6	0	18,27	0	18,48	0
17,7	0	17,28	0	17,49	1	18,7	0	18,28	1	18,49	0
17,8	0	17,29	0	17,50	1	18,8	0	18,29	1	18,50	0
17,9	0	17,30	0	17,51	1	18,9	0	18,30	0	18,51	0
17,10	0	17,31	0			18,10	0	18,31	0		
17,11	1	17,32	0			18,11	0	18,32	0		
17,12	1	17,33	0			18,12	0	18,33	0		
17,13	0	17,34	0			18,13	0	18,34	0		
17,14	0	17,35	0			18,14	0	18,35	0		
17,15	0	17,36	0			18,15	0	18,36	0		
17,16	1	17,37	0			18,16	0	18,37	0		
17,17	1	17,38	0			18,17	0	18,38	0		
17,18	1	17,39	0			18,18	0	18,39	0		
17,19	1	17,40	1			18,19	0	18,40	0		
17,20	1	17,41	1			18,20	0	18,41	1		
17,21	1	17,42	1			18,21	1	18,42	1		

i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$
19,1	0	19,22	0	19,43	1	20,1	1	20,22	1	20,43	0
19,2	0	19,23	0	19,44	0	20,2	1	20,23	1	20,44	0
19,3	0	19,24	0	19,45	1	20,3	0	20,24	1	20,45	1
19,4	0	19,25	0	19,46	1	20,4	0	20,25	1	20,46	1
19,5	0	19,26	0	19,47	1	20,5	0	20,26	1	20,47	0
19,6	0	19,27	0	19,48	0	20,6	0	20,27	1	20,48	1
19,7	0	19,28	0	19,49	0	20,7	0	20,28	1	20,49	1
19,8	0	19,29	0	19,50	0	20,8	0	20,29	1	20,50	1
19,9	0	19,30	1	19,51	0	20,9	0	20,30	1	20,51	1
19,10	0	19,31	1			20,10	0	20,31	0		
19,11	0	19,32	1			20,11	1	20,32	0		
19,12	0	19,33	0			20,12	1	20,33	0		
19,13	0	19,34	0			20,13	0	20,34	0		
19,14	0	19,35	0			20,14	0	20,35	1		
19,15	0	19,36	0			20,15	0	20,36	1		
19,16	0	19,37	0			20,16	0	20,37	0		
19,17	0	19,38	0			20,17	0	20,38	0		
19,18	1	19,39	0			20,18	0	20,39	0		
19,19	1	19,40	0			20,19	0	20,40	0		
19,20	1	19,41	0			20,20	1	20,41	0		
19,21	0	19,42	1			20,21	1	20,42	0		

i,j	$a_{i,j}$	i,j	$a_{i,j}$	i,j	$a_{i,j}$
21,1	1	21,22	0	21,43	0
21,2	1	21,23	0	21,44	0
21,3	0	21,24	0	21,45	1
21,4	1	21,25	1	21,46	1
21,5	1	21,26	1	21,47	1
21,6	1	21,27	0	21,48	0
21,7	1	21,28	0	21,49	0
21,8	0	21,29	0	21,50	0
21,9	0	21,30	0	21,51	0
21,10	0	21,31	0		
21,11	0	21,32	0		
21,12	0	21,33	0		
21,13	0	21,34	0		
21,14	0	21,35	0		
21,15	0	21,36	0		
21,16	0	21,37	0		
21,17	0	21,38	0		
21,18	0	21,39	0		
21,19	0	21,40	0		
21,20	0	21,41	0		
21,21	0	21,42	0		

APPENDIX D

SCHEDULE

This is the schedule that resulted from the solution to the scheduling problem. The tutors that work each time slot are located in the row of the labeled time slot. For example, we have business tutor three, verbal tutor 6, quantitative tutor 7, 10, and 11 scheduled on Monday from 9 a.m.-10 a.m.

SLAC TUTOR SCHEDULE

M	9am-10am	-	-	b3	-	-	-	-	-	-	v6	-	-	-	-	-	-	q7	-	-	q10	q11
	10am-11am	-	-	b3	-	-	-	-	-	-	v6	-	-	q3	-	-	-	-	-	-	q10	q11
	11am-12pm	b1	-	b3	-	v1	-	-	-	v5	-	-	q2	q3	q4	-	-	q7	-	-	-	-
	12pm-1pm	-	b2	b3	-	-	-	-	v4	-	-	-	q2	q3	q4	-	-	-	q8	-	-	q11
	1pm-2pm	-	b2	b3	-	-	-	-	-	-		-	q2	q3	-	-	-	-	q8	-	-	q11
	2pm-3pm	-	b2	-	-	-	-	-	-	-		-	-	-	-	-	q6	-	-	-	-	q11
	3pm-4pm	-	-	-	-	-	-	v3	-	-	v6	-	-	-	-	q5	-	-	-	-	-	q11
	4pm-5pm	b1	-	-	-	-	-	-	-	v5	v6	-	q2	-	q4	q5	-	-	-	-	-	-
	5pm-6pm	-	-	b3	-	-	-	-	-	v5	-	-	q2	-	q4	q5	-	-	-	-	-	-
	6pm-7pm	-	b2	-	-	-	-	-	-	v5	-	q1	q2	-	q4	q5	-	-	-	-	-	-
	7pm-8pm	-	b2	-	-	-	-	-	-	v5	-	q1		-		q5	-	q7	-	-	q10	-
	8pm-8:30pm	-	b2	-	-	-	-	v3	-	-	-	q1	-	-	-	q5	-	q7	-	-	q10	-
T	9am-10am	-	-	-	-	v1	-	-	-	-	-	-	-	-	q4	-	q6	-	-	-	-	-
	10am-11am	-	-	b3	-	v1	-	-	-	-	-	-	-	-	q4	-	q6	-	-	-	-	-
	11am-12pm	-	-	b3	-	v1	-	-	-	v5	-	-	-	-	q4	q5	q6	-	-	-	-	-
	12pm-1pm	-	-	b3	-	-	-	-	-	v5	-	-	q2	-	q4	-	q6	q7	-	-	-	-
	1pm-2pm	-	-	-	-	-	-	-	-	v5	-	-	q2	-	q4	-	q6	q7	-	-	-	-
	2pm-3pm	b1	-	-	b4	-	-	-	v4	v5	-	-	q2	-		-	q6	q7	-	q9	-	-
	3pm-4pm	b1	-	-	b4	-	-	-	v4	-	v6	-	q2	-		-	q6	q7	-	q9	-	-
	4pm-5pm	b1	-	-	b4	-	-	v3	v4	-	-	-	-	-		-	q6	q7	-	q9	q10	-
	5pm-6pm	-	-	-	b4	-	-	v3	-	-	-	-	-	q3	-	-	q6	q7	q8	-	q10	-
	6pm-7pm	-	-	-	b4	v1	-	-	-	-	-	q1	-	q3	-	-	-	q7	q8	-	q10	-
	7pm-8pm	-	b2	-	-	v1	-	-	-	-	-	q1	-	-	-	-	-	q7	q8	-	q10	-
	8pm-8:30pm	-	b2	-	-	v1	-	-	-	-	-	q1	-	-	-	-	-	q7	q8	-	q10	-
W	9am-10am	-	-	b3	-	v1	-	-	-	-	-	-	-	-	-	-	-	q7	-	-	q10	q11
	10am-11am	-	-	b3	-	v1	-	-	-	-	-	-	-	-	-	-	-	q7	-	-	q10	q11
	11am-12pm	b1	-	b3	-	v1	-	-	v4	-	-	-	q2	-	q4	-	-	q7	-	-	q10	-
	12pm-1pm	-	b2	b3	-	-	-	-	v4	-	-	-	q2	-	q4	-	-	-	q8	-	q10	-
	1pm-2pm	-	b2	b3	-	-	-	-	-	-	-	-	q2	-	-	-	-	-	q8	-	q10	-
	2pm-3pm	-	b2	-	-	-	-	-	-	-	-	-	-	-	-	-	q6	-	-	q9	q10	-
	3pm-4pm	-	-	-	-	-	-	-	-	-	v6	-	-	-	-	q5	-	-	-	q9	-	-
	4pm-5pm	b1	-	-	-	-	-	-	-	v5	v6	-	-	q3	q4	q5	-	-	-	q9	-	-
	5pm-6pm	-	-	-	b4	-	v2	-	-	-	-	-	-	q3	q4	q5	-	-	-	-	-	-
	6pm-7pm	-	-	-	b4	-	v2	-	-	-	-	q1	-	q3	q4	-	-	-	-	-	-	-
	7pm-8pm	-	-	-	b4	-	v2	-	-	-	-	q1	-	q3	-	-	-	-	-	-	-	-
	8pm-8:30pm	-	-	-	b4	-	v2	-	-	-	-	q1	-	q3	-	-	-	-	-	-	-	-
Th	9am-10am	-	-	-	-	v1	-	-	-	-	-	-	-	-	-	-	q6	-	-	-	-	-
	10am-11am	-	-	b3	-	v1	-	-	-	-	-	-	-	-	-	-	q6	-	-	-	-	-
	11am-12pm	-	-	b3	-	v1	-	-	v4	-	-	-	-	-	-	-	q6	-	-	-	-	-
	12pm-1pm	-	-	b3	-	-	-	-	-	v5	-	-	q2	-	-	-	q6	q7	-	-	-	-
	1pm-2pm	-	-	-	-	-	-	-	-	v5	v6	-	q2	-	-	-	-	q7	-	-	-	-
	2pm-3pm	b1	-	-	b4	-	-	-	v4	v5	-	-	-	-	-	-	-	q7	-	q9	-	-
	3pm-4pm	b1	-	-	b4	-	v2	-	-	-	v6	-	-	-	-	-	-	q7	-	q9	-	-
	4pm-4:30pm	-	-	-	b4	-	v2	v3	-	-	-	-	-	-	-	-	-	-	-	-	-	-
F	9am-10am	-	-	-	-	v1	-	-	-	-	-	-	-	q3	-	-	-	-	-	q9	-	q11
	10am-11am	-	-	-	-	v1	-	-	-	-	-	-	-	q3	-	-	-	-	-	q9	-	q11
	11am-12pm	-	-	-	-	v1	-	-	-	-	-	-	-	q3	-	-	-	-	-	-	-	q11
S	5pm-6pm	-	b2	-	-	-	v2	-	v4	-	-	-	-	q3	-	-	-	-	-	-	-	-
	6pm-7pm	-	b2	-	-	-	v2	-	v4	-	-	q1	-	q3	-	-	-	-	-	-	-	-
	7pm-8pm	-	-	-	-	-	v2	-	-	-	-	q1	-	q3	-	-	-	-	-	-	-	-
	8pm-9pm	-	-	-	-	-	v2	-	-	-	-	q1	-	q3	-	-	-	-	-	-	-	-

bi=business tutor i, i=1,2,..,4
vi=verbal tutor i, i=1,2,...,6
qi=quantitative tutor i, i=1,2,...,11

BIBLIOGRAPHY

Glover, Fred, and Claude McMillan. "The General Employee Scheduling Problem: An Integration of MS and AI." *Comp. & Ops. Res.* 13, no. 5 (1986): 563-573.

Pinedo, Michael. *Scheduling: Theory, Algorithms, and Systems.* 3rd ed. New York: Springer, 2008.

Redl, T. "A Study of University Timetabling that Blends Graph Coloring with the Satisfaction of Various Essential and Preferential Conditions." PhD diss., Rice University 2004.

Smythe, William R., and Lynwood A. Johnson. *Introduction to Linear Programming, With Applications.* Englewood Cliffs, N.J.: Prentice-Hall, 1966.

West, Douglas B. *Introduction to Graph Theory.* 2nd ed. Upper Saddle River, N.J.: Prentice Hall, 2001.

VITA

Holly Christine Floyd was born in Mt. Edgecumbe, Alaska, on May 22, 1980, the daughter of Monna Marie Floyd and James William Floyd. She received the degree of Bachelor of Science from Northeastern State University in Oklahoma in Spring 2008. In August 2008, she entered the Graduate College of Texas State University-San Marcos.

Email address: hf1046@txstate.edu

This thesis was typed by Holly C. Floyd