OPTIMAL VOLUNTEER ASSIGNMENT WITH AN APPLICATION TO THE DENVER B-CYCLE BIKE SHARING PROGRAM

by

Kaspari, Matthew

B.S., University of Colorado Denver, 2005

A thesis submitted to the

University of Colorado Denver

in partial fulfillment

of the requirements for the degree of

Master of Science

Applied Mathematics

2010

This thesis for the Master of Science

degree by

Kaspari, Matthew

has been approved

by

Alexander Engau

Steven Billups

Harvey J. Greenberg

Loren Cobb

Gary Kochenberger

Date

Matthew, Kaspari, (M.S., Applied Mathematics)

Optimal Volunteer Assignment with an Application to the Denver B-Cycle Bike Sharing Program

Thesis directed by Assistant Professor Alexander Engau

ABSTRACT

Volunteers are critical for many organizations and events to run properly. Assigning volunteers to tasks they prefer will increase the probability that they enjoy themselves and volunteer again in the future. We construct a goal programming model that maximizes the volunteer's preferences in a timely, effective, and fair manner. We use a two-phase model to assign volunteers. The first phase determines if a feasible assignment exists. If a feasible assignment does not exist, the model solution indicates what jobs and what shifts need more volunteers. The second phase of the model is not used unless the first phase solves with an objective function value of zero. The second phase is used to make a feasible assignment of volunteers while maximizing their preferences. Simulation experiments are conducted to study the long-term effects of the volunteer pool over time, demonstrating the importance and potential impact of assigning volunteers using our proposed methodology. Colorado's non-profit corporation "Denver Bike Sharing" used the model to assign volunteers for the launch of Denver B-Cycle, Denver's new bike sharing to promote health, quality of life and preservation of the environment in Denver.

This abstract accurately represents the content of the candidate's thesis. I recommend its publication.

Signed ____________________________________

Alexander Engau

DEDICATION

This thesis is dedicated to Harvey J. Greenberg who introduced me to the volunteer assignment problem. It was my relationship with Harvey that not only got me excited about graduate school but about operations research in general. Thank you Harvey for always being there to mentor me.

ACKNOWLEDGMENT

This thesis would not have been possible without the support of Alexander Engau, my graduate advisor. I appreciate all the time that Alex helped guide and advise me. I also want to thank Jeffery Larson for all the times he would answer his phone late at night to answer one of my many questions. I want to thank my committee for taking time away from their busy lives to help me further my career.

CONTENTS

FIGURES

Figure

TABLES

Table

1. Introduction

Volunteers are critical for many organizations and events to run properly. Many factors determine if someone volunteers for the first time or continues doing so. Smith [21] suggests that more collaboration needs to be done between disciplines in order to better understand what keeps a volunteer returning for future volunteer opportunities. Volunteer coordinators have little control over many of the reasons that keep someone volunteering. If a job is closely connected with a volunteer's motives they are more likely to continue volunteering for that organization [6]. Volunteer coordinators do not have control over someones motivation or many other factors, therefore focusing on what one can do to keep volunteers coming back is essential.

Volunteer coordinators have control over what jobs they assign a volunteer to do if any. Assigning volunteers randomly, or on a first-come first-serve basis, might not be the best option if one wants volunteers to continue volunteering. Assuming volunteers are more satisfied assigning them jobs they prefer than assigning them jobs they do not to want work. Volunteers will not always get their first choice because there are jobs that are fun and jobs that are not as much fun. A volunteer coordinator needs to make sure that all of the jobs are filled when appropriate. This is precisely why one wants to assign volunteers in a way that will maximize the overall satisfaction of the group of volunteers.

When an organization needs volunteer labor to complete various tasks, it is uncommon to have the exact number of volunteers needed. In many cases volunteer coordinators may not have enough volunteers to complete all of the jobs; however, if the volunteer opportunity is exciting volunteer coordinators may have more volunteers than they can use. In 2008 over 23,000 volunteers applied for hundreds of jobs over a two week period for the Democratic National Convention [15]. The volunteer assignment problem differs from the classic assignment problem, which most commonly assigns labor to tasks while minimizing the cost. Compared to the classical assignment problem, the main difference is to maximize volunteers preferences; however, many other features are interesting to look at as well. Since there are costs associated with labor; employers usually want to make an assignment using the least amount of people. Volunteers do not produce a cost, therefore assigning the minimum number is not an issue. If a volunteer feels that an organization is wasting their time, it may negatively affect their future volunteer involvement.

Chapter 2 reviews the literature in order to understand and model the volunteer assignment problem effectively. The literature review consists of three major sections volunteerism, assignment problem and goal programming. We discuss different meanings of the word volunteer, why people volunteer, what motivates a person to volunteer and why volunteering is important to help formulate a model that incorporates the true objectives of both volunteers and volunteer coordinators. The differences of the classical assignment problem and the volunteer assignment problem are addressed as well. The section concludes discussing some goal programming techniques.

In chapter 3 the mathematical model created to assign volunteers while maximizing their preferences is discussed. Introducing the two-phase model that we developed to make assignments independent of the number of volunteers who sign up. The first phase of the model uses ghost volunteers to determine those jobs and shifts that the volunteer coordinator does not have enough volunteers for, if any. If the first phase of the model solves with an objective function value of zero, there exists a feasible assignment and consequently solve the problem using the second phase model. The first phase tells volunteer coordinators if they need to recruit more volunteers or relax some of the volunteer constraints. To the author's knowledge this is the first time this approach has been used in the context of volunteer assignments.

In chapter 4 we discuss a simulation developed to investigate how the volunteer pool is affected by the assignments the program determines. There are many aspects that affect someones future volunteer involvement other than the assignments they were given; however, volunteer coordinators have control over the assignments so we studied how the assignments affect the volunteer pool over time. The two-phase model was not used in our simulation, therefore we introduce a model to run without interruption because the model is always feasible. Defining volunteer satisfaction by the assignments that the volunteers were given, allows the author to observe what happens to the volunteer pool over time. Based on extensive literature review it seems that it is a novel idea to define volunteer satisfaction in this context. We used MATLAB, a high-level technical computing language, to run our simulation and AMPL (A Mathematical Programming Language) to solve our volunteer assignment problem.

In chapter 5 the model that determined the assignment of the volunteers for the launch of the Denver Bike Sharing program is presented. The two-phase model was used and solved using GAMS (General Algebraic Modeling System). Denver Bike Sharing will be changing how people get around in Denver. Bike sharing programs are in Paris, Barcelona, France, Rome, and Montreal. There are many other U.S. cities that are preparing for a bike sharing system in the future; however, Denver will be the first to launch a large scale program in the United States. This section discusses the model in detail that was used to assign 284 volunteers to 35 kiosks located at 7 different geographical locations for 3 shifts during 3 days. On a personal note, assigning the volunteers for the launch was a great opportunity to demonstrate how mathematics can be applied to help solve real-world problems.

We conclude in chapter 6 by reviewing the benefits of using mathematical modeling to make volunteer assignments. Taking volunteer's preferences into consideration will allow volunteer coordinators to make better assignments and create a more pleasurable experience for both the volunteer and the volunteer coordinator. Having loyal volunteers that continue volunteering is important because volunteer labor is a competitive commodity that is limited. We conclude by exploring some possible avenues of future research.

2. Literature Review

This section reviews the models that are available and the reasons why people volunteer. This chapter is divided into three main areas: volunteerism, assignment problem and goal programming. The first section highlights many volunteer motives that volunteer coordinators do not have control over, for example a person's values. Recognizing there are many factors that determine if volunteers are satisfied with their volunteer experience is important.

2.1 Volunteerism

In this section we will address the following questions:

- What is volunteerism?
- Why do people volunteer?
- What motivates someone to volunteer?
- Why volunteering is important and what are the social consequences of volunteering?

The answers to these questions support our hypothesis that volunteer satisfaction is of crucial importance and high relevance as reflected in our objective to assign volunteers to jobs they prefer in order to promote their return in the future.

2.1.1 What is volunteerism?

There are many different ideas of what volunteerism is. Freeman [12] defines volunteer activity to be work that is performed without monetary recompense. He finds that volunteering is a "conscience good" and that people do so when asked, but would rather let someone else do it. He also claims that since volunteers do not get paid they must receive greater utility from volunteering than from working for wages or leisure.

Cnaan and Amrofell [7] look at the different uses and contexts of the word "volunteer". Their aim is to clarify and better understand what volunteer activity is. They use a mapping sentence method which grouped volunteer characteristics such as men and women motives to volunteer, their preferences for certain tasks, and the level of education of volunteers together systematically into ten interrelated facets. Who is the volunteer, what is being volunteered and who manages volunteers are three of the facets they use. They state that the word volunteer is used to cover a wide range of nonsalaried activities.

2.1.2 Why do people volunteer?

Many researchers have tried to answer this question. A fair amount of literature looks at what motivates a person to engage in volunteerism, however very few articles address what keeps them coming back [18]. Some people may feel that it is their responsibility as a member of society to volunteer [8]. Many others volunteer to acquire social status or gain skills. Some even volunteer because they are forced to. Chapman and Morley study college students that are required to volunteer 3-4 hours a week as part of the class requirements [4]. This is interesting because if one forces someone to do something is that considered volunteering? If we use Freeman's definition of volunteering and the person being forced is not getting paid, then one must consider that volunteering. If we use Merriam-Webster's definition of voluntary: *proceeding from the will or from ones own choice or consent; unconstrained by interference*; one would not consider that volunteering because the students are not necessarily choosing to volunteer. In some cases people want to volunteer because they receive a t-shirt, food, or entry into a concert [13]. Besides receiving perks there are many other reasons why people are motivated to volunteer, which will become apparent in the following subsection.

2.1.3 What motivates someone to volunteer?

Clary and Snyder [6] look in depth at why someone becomes a volunteer in the first place and why someone would continue volunteering. To understand the answers to these questions they develop six functions or motives of why people volunteer: values, understanding, enhancement, career, social, and protective. They conclude that if volunteers' satisfaction of their service is paired with receiving functionally relevant benefits the volunteer is much more likely to continue volunteering.

Chapman and Morley [4] investigate volunteer motives in college students. They use the six functions to analyze the differences in motives in each gender, whether motives change as a function of service, and to see if motives were predictive of satisfaction. The values motive is linked with high satisfaction among the college students and the social motive was associated with low satisfaction.

Their findings suggest that aligning volunteers experiences with motives that interest the students will have a significant impact on the overall satisfaction of the volunteers and therefore affect their future volunteer involvement.

Green [14] looked at the relationship between altruistic and non-altruistic volunteer motivation and how the two differed in future volunteer involvement. She found that institutions should encourage self-interest as well as altruism in volunteer recruitment because this will increase the chance that the volunteer will come back.

2.1.4 Why is volunteering important and what are the social consequences?

Volunteers help many organizations fiscally as well as socially. Volunteerism encourages innovation, efficiency and social cohesiveness. Freeman and many other authors provide ample evidence that show volunteers play a vital role in many organizations [7, 13, 14, 18, 19]. Shin and Kleiner determine that managing a volunteer correctly is very important in retaining the volunteer [20]. Making use of volunteers and being conscious of their time commitment will lead to a more pleasurable experience for the volunteer and more volunteer time later [11]. Assigning volunteers to tasks they prefer will increase the probability that they enjoy themselves and increase the chance they will volunteer in the future.

2.2 Assignment Problem

The assignment problem is one of the fundamental building blocks in combinatorial optimization and operations research. This thesis complies with this goal from a mathematical point of view, using the techniques and models discussed in the following two sections. Let there be two sets I and J. Each element of I can be assigned to an element of J at a cost or utility C_{ij}. The assignment problem is to match elements of I and J so that every element is assigned to at most one other element from the other set, and that the sum of the costs or utilities is minimized or maximized, respectively. The mathematical programming model is given by

$$\textbf{objective function} \qquad \min \sum_{i \in I} \sum_{j \in J} C_{ij} x_{ij}$$

$$\textbf{subject to} \qquad \sum_{j \in J} x_{ij} \leq 1 \, \forall \, i \in I$$

$$\sum_{i \in I} x_{ij} \leq 1 \, \forall \, j \in J$$

$$x_{ij} \geq 0 \qquad \forall i \in I \,, \, j \in J$$

$$\textbf{where} \qquad x_{ij} = \begin{cases} 1 & \text{if person i is assigned to job j} \\ 0 & \text{otherwise} \end{cases}$$

The objective function is what we are optimizing. The first constraint guarantees that each element in I is assigned to at most one element in J. The second constraint is forcing each element in J to be assigned to at most one element in I.

Considerable amount of work done has been done on the assignment problem over the past 60 years [9]. The assignment problem is polynomial-time solvable as a linear program, and more efficient algorithms have been developed using network flow concepts [2]. An overview of many fundamental network flow algorithms is given by Ahuja, Magnanti, and Orlin [1]. Few papers are mentioned in this thesis that address the assignment problem [3, 9], however, there are many more available.

Many research papers are done in the context of assigning tasks to work stations in manufacturing or data processing settings [20]. Many times it is thought that the person can move from work station to work station, which is the same as assigning labor to machines or labor to tasks. Dantzig [9] determines the total number of booth hours required during a day. In this paper, the simplex method is used to minimize the number of men needed to run the required number of toll booths. Thompson [22] follows Dantzig's method and points out some weaknesses such as different levels of customer service, that were addressed to find a better optimum. Thompson claims that different levels of customer service will affect profits. By changing the number of employees working in a planning period Thompson made schedules that are more profitable than Dantzig's method. Briefly discussing a few papers is important because highlighting the fact that researchers have been looking at these problems and refining their approaches to solve them for many years.

Although there are many papers on the labor assignment problem, very few address the idea of volunteer labor or entertain the idea of labor at no cost. This is a modified form of the traditional assignment problem because we replace costs with preferences and we maximize instead of minimizing. Mathematically there is not a difference, but there are interesting features that come from looking at the problem differently. Emanuele [11] has pointed out that it might not be reasonable to consider volunteer labor as a zero cost or cost-free because if volunteer labor is not utilized correctly there might be a chance of less volunteer activity in the future. Implying that a non-utilized volunteer carries a negative cost, which we address in our model by introducing a penalty for assigning too many volunteers to a particular job.

Although mathematically equivalent in its basic formulation, the volunteer assignment problem is different than the traditional labor assignment problem in many aspects [19] and often subject to additional constraints. The objective function for many traditional labor assignment problems is to minimize the cost while completing all of the tasks. Volunteers do not endure a cost, therefore maximizing volunteer's preferences is an alternative to minimizing labor costs. Sampson [19] shows how the different assumptions in the volunteer assignment lead to significantly different results than those coming from traditional labor assumptions. Both papers [19, 20] find that volunteers that are not utilized at all are much less likely to volunteer in the future. However, Sampson [19] also finds that volunteers that are utilized less than what they sign up for are as likely to volunteer in the future as volunteers that are assigned exactly what they sign up for. Sampson also finds that volunteers that are not utilized at all have less of a chance volunteering than if they are utilized a portion of what they sign up for because future volunteer labor is a function of current task assignment. The author addresses this in chapter 4, when we show how volunteer satisfaction effects future volunteer involvement. Highlighting the fact that it might be worthwhile to involve more volunteers than trying to assign the minimum number of volunteers needed.

Gordan and Erkut [13] developed a novel decision support tool which was used in 2003 at the Edmonton Folk Festival to assign the volunteers. They were able to provide a master schedule and individual schedules for the event. Making assignments by maximizing the volunteers shift preferences while obeying a set of constraints. Each volunteer had to work a minimum of 20 hours, one shift on Thursday, one shift on Friday and could not work more than 6 hours on Saturday or Sunday. Working with a volunteer coordinator they were able to accommodate the coordinator's requests as well as many of the volunteer's requests.

2.3 Goal / Multi-objective Programming

Goal programming (GP), an extension of linear programming (LP), is a branch of multi-objective optimization and multiple-criteria decision making that is well established in the field of operations research. GP differs from LP most notably because its ability to deal with multiple, often conflicting, objective functions. LP deals with single objectives such as maximizing revenue, maximizing return, minimizing cost or minimizing travel time. GP problems often have conflicting objectives; upgrade product quality and reduce production costs, maximize profits and increase wages, maximize return and minimize risk to name a few. Each of the objectives are given a measure and each of these measures are given a target or goal that is desired to be achieved. Deviations from the goals are usually penalized in the objective function. Being able to consider conflicting objectives allows modelers to work on a large number of applied problems.

In 1955 Charnes, Cooper and Ferguson [5] introduce goal programming. They discuss executive compensation plans but noted that one can apply these methods to many other problems such as production scheduling, logistics, and market analysis, to name only a few. However, the term goal program first appears in text in 1961 by Charnes and Cooper [5]. One can solve employee scheduling problems using deterministic or stochastic goal programs. In most cases the models seek to assign employees while minimizing the total operating costs. Mabert and Watts [17] apply a deterministic goal program to tour-shift scheduling problems so that employers can make use of part time employees in order to save money. They were able to identify feasible staffing schedules that maximized productivity and minimized cost. Easton and Rossin [10] show that by using a stochastic goal program they can integrate labor requirements and scheduling decisions to make less costly staffing decisions than many deterministic goal programs. The reason is that many times deterministic goal programs violate labor requirements by assigning a worker to a single period of work. Violations of the labor requirements may penalize the objective function when solving.

Many different procedures exist for estimating penalties for target deviations in goal programming. Lam and Choo [16] look at using a linear goal program to estimate the criterion weights that influence the preferences of all the alternatives presented. If decision makers can make preference judgments on which alternative they prefer and by how much, their method has greater predictive power on choosing the best weights. Their simulation experiment proves that goal programming performed better than ordinary least squares. The following model combines an assignment model with goal programming formulations to allow the simultaneous consideration of both volunteer and volunteer preferences and requirements by the volunteer coordinator.

3. Model

Before we discuss the mathematical model we explain what the model does, what the model needs and how the data is collected. This provides the necessary framework that we can build on. Given an event or volunteer opportunity that offers different jobs at different locations for a specific number of shifts, the model assigns volunteers to tasks in consideration of preferences, geographical location, time availability and demand requirements by a volunteer coordinator. The model makes an optimal assignment of volunteers to tasks by maximizing the volunteers preferences while satisfying a set of relevant constraints. Our formulation takes into account the preferences of the volunteers as well as the requests from one or more volunteer coordinators. Volunteer coordinators are required to provide information on the number of volunteers that need to be at each job during each shift, along with the minimum number of shifts each volunteer is required to work. Preferences are collected by asking volunteers to fill out a survey to gather information regarding their availability and preferences with respect to the type and number of jobs they are willing to work. An example for such a survey can be found in appendix B for the problem discussed in Chapter 5. We ask the volunteer how much they prefer to do each job, not how much they prefer to do each shift which is a slight modification of what the authors did for the Edmonton Folk Festival [13]. This allows us to assign volunteers to what jobs they prefer to do and the shifts they are available to work, which to our knowledge is different than any models presented in the literature.

3.1 Mathematical Model

First we introduce the sets, parameters, and variables that we use in the model, refer to table 3.1. Using roman letters for the variables and Greek letters for the parameters allows us to determine what each symbol is quickly. The primary decision is whether to assign volunteer i to job j during shift k. Therefore, a binary decision variable is defined by

$$x_{ijk} = \begin{cases} 1 & \text{if volunteer i is assigned to job j during shift k;} \\ 0 & \text{otherwise.} \end{cases}$$

This produces a 3-dimensional assignment matrix that shows what job and what shift each volunteer works, if any. The matrix X_{ijk} can be thought of as the master schedule.

A secondary decision variable y_i will be used to turn on constraints in the model and is denoted by

$$y_i = \begin{cases} 1 & \text{if volunteer } i \text{ is assigned to any job at all;} \\ 0 & \text{otherwise.} \end{cases}$$

In order to ensure that y_i takes the correct value we introduce the constraint

$$x_{ijk} \leq y_i \ \ \forall\, i.$$

If a volunteer is not assigned to any shift at all then we do not need to satisfy the constraints that pertain to the number of shifts they want to work or are required to work.

Table 3.1: Nomenclature

Symbol	Description
Sets / Indicies	
$i \in \{1, ..., V\}$	Volunteer
$j \in \{1, ..., J\}$	Job
$k \in \{1, ..., S\}$	Shift
$t \in \{1, ..., T\}$	Year
C	Conflict Set
Decision Variables	
x_{ijk}	Equals 1 if we assign volunteer i to job j during shift k, 0 otherwise
y_i	Equals 1 if volunteer i is assigned to any job at all, 0 otherwise
Preferences / Penalties	
π_{ijk}	How much volunteer i prefers to do job j during shift k
ρ^1_{jk}	Penalty for assigning less volunteers to job j during shift k than the goal
ρ^2_{jk}	Penalty for assigning more volunteers to job j during shift k than the goal
ρ^3_i	Penalty for assigning volunteer i less shifts than their goal
ρ^4_i	Penalty for assigning volunteer i more shifts than their goal
ρ^5_i	Penalty for assigning a volunteer less shifts than they are required to work
Variables	
G_{jk}	Number of ghost volunteers assigned to job j during shift k
u^+_{jk}	The number of volunteers assigned to job j during shift k above the goal
u^-_{jk}	The number of volunteers assigned to job j during shift k below our goal
v^+_i	The number of shifts assigned to volunteer i above their goal
v^-_i	The number of shifts assigned to volunteer i below their goal
w^-_i	The number of volunteers assigned to job j during shift k below the required number
w^+_i	The number of volunteers assigned to job j during shift k above the required number
Parameters	
α_{ik}	Equals 1 if volunteer i is available to work shift k, 0 otherwise
β_{ij}	Equals 1 if volunteer i has the required skill needed for job j
μ_{jk}	Goal for the number of volunteers to be assigned to job j during shift k
ν_i	Goal for the number of shifts volunteer i wants to work
ω_i	Minimum number of shifts volunteer i is required to work
μ^+_{jk}	Maximum deviation from the assignment goal of job j during shift k
μ^-_{jk}	Minimum deviation from the assignment goal of job j during shift k
ν^+_i	Maximum deviation from the number of shifts goal of volunteer i
ν^-_i	Minimum deviation from the number of shifts goal of volunteer i

To obtain results that are reasonable we have to constrain our variables using the information obtained from the volunteers and the volunteer coordinators. We assume that each volunteer can be assigned to at most one job per shift, if they are available

$$\sum_j x_{ijk} \leq \alpha_{ik} \;\; \forall\, i, k.$$

We define the parameter $\alpha_{ik} = 1$ if volunteer i is available to work during shift k. This constraint will make sure that we do not assign a volunteer to a job if they are not available and we will not assign a volunteer to more than one job per shift.

The volunteer may be required to lift a certain number of pounds, have a driver license or be proficient with a particular kind of software. These are just a few of the possible requirements that may need to be addressed; there are obviously many others that may be event specific. In order for a volunteer to be assigned a particular job they need to have the required skill to work the job if any is required at all

$$x_{ijk} \leq \beta_{ij} \;\; \forall\, i, j, k.$$

We define the parameter $\beta_{ij} = 1$ if volunteer i has the required skill needed to work job j.

The volunteer coordinator provides us with appropriate values for the minimum, maximum, and ideal number of volunteers needed to work each job during each of the shifts. These values will be represented by $\mu_{jk}^{\min}$, $\mu_{jk}^{\max}$, and μ_{jk} respectively. We calculate the minimum and maximum deviations from the target number of volunteers working job j during shift k as follows

$$\mu_{jk}^{-} = \mu_{jk} - \mu_{jk}^{\min},$$

$$\mu_{jk}^{+} = \mu_{jk}^{\max} - \mu_{jk}.$$

The variables u_{jk}^{-} and u_{jk}^{+} are the number of volunteers assigned below and above the goal, respectively. They will be bounded below by 0 and above by the minimum and maximum deviation from the volunteer assignment goal

$$0 \leq u_{jk}^{+} \leq \mu_{jk}^{+},$$

$$0 \leq u_{jk}^{-} \leq \mu_{jk}^{-}.$$

The constraint for assigning the target number of volunteers μ_{jk} needed to work each job j and shift k is

$$\sum_{i} x_{ijk} + u_{jk}^{-} - u_{jk}^{+} = \mu_{jk} \ \ \forall j, k.$$

The volunteer coordinator also gives us the minimum number of shifts that volunteer i is required to work, denoted by ω_i. In order to ensure that we do not assign a volunteer less shifts than they are required we need the constraint

$$\sum_{j} \sum_{k} x_{ijk} + w_{i}^{-} - w_{i}^{+} = y_i \omega_i \ \ \forall i.$$

We can think of w_i^{-} and w_i^{+} as non-negative slack and surplus variables. If w_i^{+} is positive for any i that means that we assigned a volunteer less shifts then they are required to work and the objective function will be penalized.

The constraint for volunteer i is activated if and only if volunteer i is assigned to any job at all.

Each volunteer is required to provide us with the minimum, maximum, and ideal number of shifts they want to work. These values will be represented by $\nu_i^{\min}$, $\nu_i^{\max}$, and ν_i respectively. Similarly as above these requests will be used in our model. We calculate the minimum and maximum deviations from the target number of shifts volunteer i is willing to work as follows:

$$\nu_i^- = \nu_i - \nu_i^{\min},$$

$$\nu_i^+ = \nu_i^{\max} - \nu_i.$$

The variables v_i^+ and v_i^- represent the number of shifts above and below their goal, respectively. They are bounded below by 0 and above by the minimum and maximum deviation from the volunteer shift goal

$$0 \leq v_i^+ \leq \nu_i^+,$$

$$0 \leq v_i^- \leq \nu_i^-.$$

The constraint for assigning a volunteer the number of shifts they want to work is:

$$\sum_j \sum_k x_{ijk} + v_i^- - v_i^+ = y_i \nu_i \;\; \forall\, i.$$

If any volunteers are not assigned any job at all we do not have to satisfy their shift constraints because $y_i = 0$ and therefore $v_i^- = v_i^+ = 0$. The variables $v_i^- = v_i^+ = 0$ because they have to be equal in this situation and if we allowed either one to be greater than zero our objective will be penalized, thus the model forces them to be equal to zero.

Sometimes there will be jobs that can not be worked consecutively because of a number of different reasons. Two common conflicts may be if two jobs are located too far apart or if a volunteer coordinator does not want a volunteer to work too many shifts in one day. We define the conflict set as

C = $\{(j,j')$: job j' cannot be assigned to the same volunteer in the shift that follows job $j\}$.

The model captures a conflict set by putting a constraint in our model that will not allow assigning volunteers if there is a conflict. The objective function combines several components that are discussed in detail in the following paragraphs. The first term

$$\sum_i \sum_j \sum_k x_{ijk} p_{ijk},$$

represents the sum of the volunteer preferences and is considered the total preferences of all of the volunteers.

The second term

$$\sum_j \sum_k \rho^1_{jk} u^-_{jk},$$

penalizes the objective if we assign less volunteers to a particular job than the specified goal. The number of volunteers we are short of the goal for each job j during shift k is denoted by u^-_{jk}. It is important to note that the penalty can be different for the same job occurring at different shifts because there may be times when assigning more or less is suitable. For example trash pick up in the beginning or middle of the day might be different than trash clean up at the end of the day. There are jobs that are more critical than others and assigning appropriate penalties allows us to assign some jobs with a higher priority.

The third term

$$\sum_j \sum_k \rho^2_{jk} u^+_{jk},$$

corresponds to assigning more volunteers than desired to job j during shift k. We penalize the objective function differently for some jobs because having more volunteers at some jobs may not be as important as other jobs. The number of volunteers that we assign above the target specified for each job j and shift k is expressed by u_{jk}^{+}.

The fourth term addresses the shortage of assignments of volunteer i, using

$$\sum_i \rho_i^3 v_i^-,$$

in the objective penalizes if we assign a volunteer to less shifts than their goal. The shortage of shifts volunteer i is assigned, compared to their target number is denoted by v_i^-. Sampson [19] found that volunteers that were assigned less work than they signed up for were just as satisfied as the people who were assigned the number they signed up for, there was not a negative affect on their future volunteer involvement. Therefore the penalty for assigning a volunteer less shifts than they signed up for may be small.

The fifth term

$$\sum_i \rho_i^4 v_i^+,$$

represents the penalty if we assign a volunteer to more shifts than his or her goal. The number of shifts that volunteer i is assigned above their goal is denoted by v_i^+. The penalty ρ_i^4 might be constant because assigning someone more shifts than their goal may have the same negative impact no matter who the person is. This penalty might be larger given that if one asks a volunteer to work more than they desire the volunteer coordinator runs the risk of them not coming at all or feeling taken advantage of and once again negatively affecting their future volunteering [19].

The last term

$$\sum_i \rho_i^5 w_i^-,$$

penalizes the objective if we must assign volunteers to less shifts than the volunteer coordinator requires them to work. We represent the number of shifts volunteer i is assigned less than the required number by w_i^-. The penalty on this term depends on how much the volunteer coordinator wants to enforce this rule.

The model will maximize the objective function. This implies that we are making assignments in such a way that maximizes our volunteer preferences, which is captured by the first term. The remaining terms in the objective function are subtracted from the first term because they capture violations of the requests of volunteers and volunteer coordinators. If the model violates any of the targets specified by a volunteer or volunteer coordinator one or more of the variables u_{jk}^-, u_{jk}^+, v_i^-, v_i^+, and w_i^- will have a value greater than 0. Since these variables penalize the objective the model seeks to minimize them. However, if a volunteer's preference to work a job is higher than the penalty for assigning the volunteer, the model will make the assignment because it will increase our objective.

Now that the components of the model have been discussed in detail we present the goal programming model, refer to figure 3.1.

3.2 Two-Phase Approach

The model discussed above can not make an assignment if the model is not feasible. If we do not have enough qualified volunteers available to work all of the tasks, then the model above will not be able to make an assignment. In order to test feasibility we develop a two-phase approach to the volunteer assignment problem, because it is likely that in many situations volunteer coordinators will not have the required number of skilled volunteers to complete all of the tasks. If there is not enough volunteers to complete all of the tasks it will be helpful to provide the volunteer coordinators with what jobs and what shifts they need more volunteers. Solving the first-phase of our model will tell us what volunteer shortages we have if we have any at all. Before one knows if the problem is feasible, the first-phase model will have to be solved with an objective function value of zero. When that happens we can solve the second-phase model because there is enough volunteers to work all of the tasks. We need to introduce the new variables

$$G_{jk} = \text{number of ghost volunteers assigned to job } j \text{ during shift } k,$$

which are referred to as the ghost variables. G_{jk} equals the number of volunteers short of the required number needed to work job j during shift k. The fourth constraint in the model above is changed to

$$\sum_i x_{ijk} + G_{jk} - u_{jk}^- - u_{jk}^+ = \mu_{jk} \quad \forall\, j, k.$$

The objective function is changed to

$$\min \sum_{j} \sum_{k} G_{jk}.$$

We want to minimize the number of ghost volunteers that are assigned. If the revised model solves with an objective function value of zero, that means the model is feasible and can be solved using the original model. If the objective function value is not zero the volunteer coordinator needs to recruit more volunteers, relax some of the constraints, or change some of the parameters. This is very valuable to volunteer coordinators because if there is not enough volunteers they will know precisely what jobs and shifts where the shortages occur.

This phase of the model solves even if the volunteer coordinator does not have the required number of volunteers and the model shows them what particular jobs and shifts they need more volunteers for. This is useful information to provide the volunteer coordinator with. To our knowledge this approach has not been used in any previous volunteer assignment model and is first proposed in this thesis.

Figure 3.1: Phase 2 Model

objective function:

$$\max \sum_{ijk} x_{ijk}\pi_{ijk} - \sum_{jk} \rho^1_{jk} u^-_{jk} - \sum_{jk} \rho^2_{jk} u^+_{jk} - \sum_{i} \rho^3_{i} v^-_{i} - \sum_{i} \rho^4_{i} v^+_{i} - \sum_{i} \rho^5_{i} w^-_{i}$$

subject to:

$$\sum_{j} x_{ijk} \leq \alpha_{ik} \quad \forall\, i, k$$

$$x_{ijk} \leq \beta_{ij} \quad \forall\, i, j, k$$

$$x_{ijk} \leq y_i \quad \forall\, j, k$$

$$\sum_{i} x_{ijk} + u^-_{jk} - u^+_{jk} = \mu_{jk} \quad \forall\, j, k$$

$$\sum_{jk} x_{ijk} + v^-_{i} - v^+_{i} = y_i \nu_i \quad \forall\, i$$

$$\sum_{jk} x_{ijk} + w^-_{i} - w^+_{i} = y_i \omega_i \quad \forall\, i$$

$$0 \leq u^+_{jk} \leq \mu^+_{jk} \quad \forall\, j, k$$

$$0 \leq u^-_{jk} \leq \mu^-_{jk} \quad \forall\, j, k$$

$$0 \leq v^+_{i} \leq \nu^+_{i} \quad \forall\, i$$

$$0 \leq v^-_{i} \leq \nu^-_{i} \quad \forall\, i$$

$$w^-_{i},\, w^-_{i} \geq 0 \quad \forall\, i, j, k$$

where

$$x_{ijk} = \begin{cases} 1 \text{ if we assign volunteer } i \text{ to job } j \text{ during shift } k \\ 0 \text{ otherwise} \end{cases}$$

$$y_i = \begin{cases} 1 \text{ if volunteer } i \text{ is assigned to any job at all} \\ 0 \text{ otherwise} \end{cases}$$

4. Simulation

There are a many different ways one can go with this model. One possible avenue of research is to look at the volunteer assignment problem over many years where the volunteer pool is a result of how satisfied the volunteers were the previous year. A simulation allows us to validate that the model increases volunteer satisfaction improving the event and increases the chance of a volunteer coming back. The simulation also allows us to find appropriate penalties and parameters in the model and justify our objective function. This is the topic of this chapter.

We modified the previous model to a one-phase model in order to simulate multiple years without having the program break down. If the volunteer coordinator does not have enough volunteers to satisfy all of the constraints, the previous model will solve using ghost volunteers. In order to solve without using ghost volunteers we construct a third goal program that will allow us to make assignments without having the required number of volunteers.

4.1 Motivation and Description

The new model we construct will assign volunteers in such a way that maximizes their preferences subject to a set of constraints given any number of volunteers signing up. We built a simulation model that allows one to observe how the number of volunteers change over time and to gain insight to make suggestions to volunteer coordinators that help them assign volunteers more strategically. Strategically means assigning volunteers more fairly, efficiently and in less time.

Since the model makes the assignments looking at how these assignments effect the volunteer's future involvement is important.

We assume that if volunteer coordinators assign volunteers to jobs they prefer to do the volunteers are more likely to comeback and volunteer the following year than if they were given a job they did not want to do. We also assume that if volunteers are satisfied then there is a chance that they may invite a friend to volunteer with them the following year. Our last assumption is that if a volunteer is not assigned any jobs in the current year there is still a chance that they will return the following year.

Next we discuss the new parameters and penalties that effect the volunteer assignments and the volunteer pool over time. The return rate of a non-assigned volunteer is represented by η, the return rate of an assigned volunteer is denoted by ζ, and the rate at which a volunteer invites someone is θ. Since these parameters are unknown, in general, we want to illustrate what impact they have on the results of our model to help show volunteer coordinators the importance of assigning volunteer's according to their preferences. The penalties directly effect the number of people that are assigned each year because they effect whether we assign a volunteer to more shifts than they signed up for or if the model allows to assign more or less volunteers to a particular job. The number of volunteers that volunteer the following year depends on the number of volunteers being assigned because the more volunteers that are assigned increases the chance that more volunteers return and are invited the following year.

Now we introduce n_{it}, which represents the number of shifts volunteer i is assigned during year t. Determining a volunteer's satisfaction accurately is very hard to provide a metric for. We quantify a volunteer's satisfaction in terms of the assignments they were given because we think it makes sense:

$$s_{it} = \begin{cases} \frac{\sum_i x_{ijk} p_{ijk}}{n_{it} \max p_{ijk}} & \text{if } n_{it} > 0 \\ 0 & \text{if } n_{it} = 0. \end{cases}$$

This ratio helps determine the volunteer pool for our simulation because it is used to determine the rate that a volunteer returns next year and the rate that the volunteer invites someone else to participate. Note that when a volunteer is not assigned a job, volunteer satisfaction is defined to be 0.

The rate that a volunteer returns the following year is defined by

$$R_{it} = \begin{cases} \eta & \text{if } s_{it} = 0 \\ \zeta s_{it} & \text{if } s_{it} > 0. \end{cases}$$

Although it seems reasonable to assume that the more satisfied volunteers are with their job the more likely they are to continue volunteering and invite someone else to participate, we understand that volunteers' satisfaction levels are not the only factor that determines their future volunteer involvement.

The rate that a volunteer invites someone else to volunteer is given by

$$I_{it} = \theta s_{it}.$$

Note that if a volunteer was not assigned a job during year t, the rate that the volunteer will invite someone is 0. This is not true all the time, but seems a valid assumption, in general, because if one did not have the volunteer experience they may not be as eager to convince or invite someone else to partake. The model assumes that $\zeta \geq \theta$, based on the assumption that a volunteer is more likely to volunteer them self than invite someone else to volunteer.

The volunteer pool for the following year is a result of how satisfied the volunteers were the previous year, new volunteers signing up and volunteers not coming back. There are a number of reasons why a person might volunteer for the first time, refer to section 2.1.2, however people may have heard of the opportunity on marketing collateral, a website, through the grapevine, or many other sources. There are also a number of reasons why a person will not volunteer in the future; people may have moved out of the area, died, or just do not want to volunteer again. There will be people that volunteer or do not volunteer for a particular reason which has nothing to do with the volunteer assignments, however we do not consider them in our simulation and may assume for simplicity that the effects cancel each other out.

The volunteer pool for year t is represented by

$$(VP)_t = \sum_i R_{it} + \sum_i I_{it}.$$

The new volunteer pool is used as input data into the model for the following year. Before the model makes an assignment the following year we have to randomly generate new data for the new volunteers. The data consists of preference levels for the jobs, maximum number of shifts they are willing to work and their availability. Then we use the goal programming assignment model presented in this chapter to assign the new volunteer pool, calculate the satisfaction levels of the new volunteers, and continue this process for the desired number of years.

4.2 Simulation Model

Before we discuss the model the new parameters and penalties are presented; refer to table 4.1. Next we introduce the simulation model, refer to figure 4.1.

Table 4.1: Simulation Nomenclature

Symbol	Description
Sets / Indicies	
$i \in \{1, ..., V\}$	Volunteer
$j \in \{1, ..., J\}$	Job
$k \in \{1, ..., S\}$	Shift
$t \in \{1, ..., T\}$	Year
C	Conflict Set
Decision Variable	
x_{ijk}	Equals 1 if we assign volunteer i to job j during shift k, 0 otherwise
Preferences / Penalties	
π_{ijk}	How much volunteer i prefers to do job j during shift k
ρ_i^6	Penalty for assigning less than minimum number of volunteers needed
ρ_i^7	Penalty for assigning more than maximum number of volunteers needed
ρ_i^8	Penalty for assigning a volunteer more shifts than maximum number indicated
Variables	
s_{it}	Satisfaction percentage of volunteer i during year t
n_{it}	Number of shifts volunteer i is assigned during year t
a_{jk}^+	Number of volunteers above the minimum needed
a_{jk}^-	Number of volunteers below the minimum needed
b_{jk}^+	Number of volunteers above the maximum needed
b_{jk}^+	Number of volunteers below the maximum needed
c_{jk}^+	Number of shifts above the maximum determined by volunteer i
c_{jk}^-	Number of shifts below the maximum determined by volunteer i
Parameters	
α_{ik}	Equals 1 if volunteer i is available to work shift k, 0 otherwise
γ_{jk}	Minimum number of volunteers needed to work job j during shift k
δ_{jk}	Maximum number of volunteers needed to work job j during shift k
τ_i	Maximum number of shifts volunteer i is willing to work
R_{it}	Rate that volunteer i will come back in year t
I_{it}	Rate that volunteer i will invite someone for year t
η	Return rate of non-assigned volunteer
ζ	Return rate of assigned volunteer
θ	Rate that volunteer invites someone else
$(VP)_t$	Volunteer pool during year t

The main differences compared to the previous model are that all of the con-

Figure 4.1: Simulation Model

objective function:

$$\max \sum_{ijk} x_{ijk}\pi_{ijk} - \sum_{jk} \rho^5_{jk} a^-_{jk} - \sum_{jk} \rho^6_{jk} b^+_{jk} - \sum_{i} \rho^8_{i} c^-_{i}$$

subject to:

$$\sum_{j} x_{ijk} \le \alpha_{ik} \quad \forall\, i,k$$

$$\sum_{i} x_{ijk} + a^-_{jk} - a^+_{jk} = \gamma_{jk} \quad \forall\, j,k$$

$$\sum_{jk} x_{ijk} + b^-_{jk} - b^+_{jk} = \delta_{jk} \quad \forall\, i$$

$$\sum_{jk} x_{ijk} + c^-_{i} - c^+_{i} = \tau_{i} \quad \forall\, i$$

$$a^-_{jk},\, a^+_{jk},\, b^-_{jk},\, b^+_{jk},\, c^-_{i},\, c^+_{i} \ge 0 \quad \forall\, i,j,k$$

where $$x_{ijk} = \begin{cases} 1 \text{ if we assign volunteer } i \text{ to job } j \text{ during shift } k \\ 0 \text{ otherwise.} \end{cases}$$

straints are considered goals, which we can violate. The objective function is penalized if the model violates the goal or target. This allows one to make an assignment even if there is not the required number of volunteers needed. We want to make an assignment without introducing ghost volunteers, so we modified the model.

4.3 Scenarios

Looking at a few specific examples highlights some of the characteristics of the model and helps show how the penalties and rates effect the average number of volunteers and the total preferences. Starting with a base case that we simulate for 30 years; refer to table 4.2. Then we change one penalty or parameter at a time to see how the change effects the model. We can see that in the base

Table 4.2: Simulation: Base Case - $\rho^5_{jk} = 6$, $\rho^7_{jk} = 6$, $\rho^8_{jk} = 6$, $\eta = .25$, $\zeta = .90$, $\theta = .50$

Problem ID	# of Jobs	Start # of Vol.	Avg. # of Vol.	Total Pref.	Avg. Sat. of Vol.
1	5	5	8.20	23.27	2.84
2	5	10	8.01	23.43	2.91
3	5	15	8.03	23.93	2.97
4	5	20	7.77	23.77	3.06
5	10	5	15.07	47.50	3.15
6	10	10	15.23	48.73	3.19
7	10	15	15.43	48.83	3.16
8	12	5	17.77	56.57	3.18
9	12	10	19.93	59.63	3.03
10	12	15	17.60	59.20	3.36

case, the average number of volunteers is slightly higher than the number of jobs. This occurs because we successfully assign volunteers to the exact number of jobs and therefore volunteers might not only return themselves but they may invite someone as well. By setting all of the penalties sufficiently large we do not encourage assigning more than the maximum number or less than the minimum number of volunteers needed and we do not assign a volunteer to more shifts than they signed up for. We learned that as the number of volunteers increases the average satisfaction of the volunteers usually decreases. We plot the volunteer pool over time and the average satisfaction of volunteers for problem ID 1 and ID 10 in the base case, refer to figure 4.2.

If we increase $\eta = .50$ that means if you were not assigned a job then you

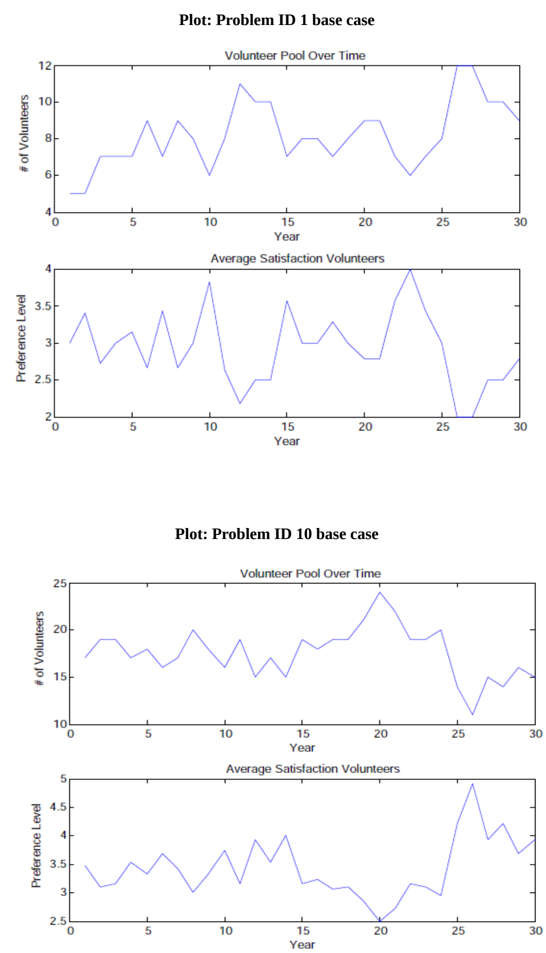

Figure 4.2: Simulation Plot

have a 50% chance of volunteering again. The average number of volunteers increases slightly because more volunteers come back each year; refer to table 4.3. When we increase the return rate of a volunteer that did not get assigned,

Table 4.3: Simulation: $\eta = .50$

Problem ID	# of Jobs	Start # of Vol.	Avg. # of Vol.	Total Pref.	Avg. Sat. of Vol.
1	5	5	10.20	23.30	2.28
2	5	10	8.93	23.67	2.66
3	5	15	10.97	24.23	2.20
4	5	20	10.40	23.87	2.30
5	10	5	19.90	47.60	2.39
6	10	10	22.50	48.80	2.16
7	10	15	21.93	49.10	2.23

then the average number of volunteers increases, total preferences and average satisfaction stays approximately the same.

Next observe what happens to the volunteer pool when we change ζ and θ, the rates that volunteers return and invite someone respectively. The results for $\zeta = .7$ are given in table 4.4. We can observe that the average number of volunteers is much smaller for all cases which is because we reduced the rate that volunteers would come back. One interpretation of $\zeta = .9$ is that he more satisfied the volunteer is the more likely they will return. By reducing the rate to .70 we relax the dependence of our assignments on the rate at which the volunteers come back, resulting in a lower average number of volunteers. The total preferences is slightly lower as well because less volunteers are available to be assigned to jobs.

Table 4.4: Simulation: $\zeta = .\mathbf{70}$

Problem ID	# of Jobs	Start # of Vol.	Avg. # of Vol.	Total Pref.	Avg. Sat. of Vol.
1	5	5	4.43	14.70	3.32
2	5	10	5.13	17.70	3.45
3	5	15	4.50	15.40	3.42
4	5	20	5.60	19.10	3.41
5	10	5	9.33	38.33	4.40
6	10	10	10.80	40.73	3.96
7	10	15	11.57	45.23	3.91
8	12	5	13.57	52.53	3.87
9	12	10	14.97	55.13	3.68
10	12	15	15.97	57.93	3.63

Results of changing $\theta = .25$ are given in table 4.5. Since we reduced θ to .25 there is smaller chance that volunteers will invite someone, which is why determine the average number of volunteers decreases. We think that the average satisfaction increases slightly because there are not as many new volunteers to assign so the average satisfaction is slightly higher than the base case.

If we change the penalty for assigning more volunteers than the maximum number needed to a value less than the maximum preference

$$\rho^7_{jk} < \max\ \pi_{ijk},$$

then the average number of volunteers and the total preferences increases in almost every case. The average satisfaction stays about the same, as shown in table 4.6. We stopped the simulation after half the time due to the apparent explosion in the number of volunteers, caused by allowing the model to assign volunteers that have a job preference larger than the penalty for assigning the

Table 4.5: Simulation: $\theta = \mathbf{.25}$

Problem ID	# of Jobs	Start # of Vol.	Avg. # of Vol.	Total Pref.	Avg. Sat. of Vol.
1	5	5	3.60	15.03	4.17
2	5	10	5.83	22.67	3.89
3	5	15	4.60	16.57	3.60
4	5	20	6.13	22.17	3.61
5	10	5	10.30	43.57	4.23
6	10	10	9.93	41.67	4.19
7	10	15	11.43	47.57	4.16
8	12	5	12.93	51.73	4.00
9	12	10	13.97	57.03	4.08
10	12	15	13.47	56.87	4.22

maximum number of volunteers needed. If we change the penalty for assigning

Table 4.6: Simulation: $\rho^{\mathbf{7}}_{\mathbf{jk}} = \mathbf{4.5}$

Problem ID	# of Jobs	Start # of Vol.	Avg. # of Vol.	Total Pref.	Avg. Sat. of Vol.
1	5	5	10.33	33.40	3.14
2	5	10	14.80	50.20	3.39
3	5	15	9.33	29.53	3.16
4	5	20	23.20	78.86	3.40

less volunteers than the minimum number needed it will not effect the number of people being assigned because assigning a volunteer increases the objective function so assigning less than the minimum needed will not happen. Therefore we do not need to look at changing ρ^6_{jk}. If we change the penalty for assigning a volunteer more shifts than they signed up for to a value less than the maximum preference

$$\rho^8_{jk} < \max \pi_{ijk},$$

we allow assigning a volunteer to more shifts than they signed up for. We do not recommend allowing this to occur because it increases the chance that the volunteer has a positive volunteer experience. We do not show results on this penalty because the student version of $AMPL$ reaches its maximum number of variables very quickly.

4.3.1 Simulation Insight

One can run the simulation with infinitely many different penalties, return rates and invitation rates of the volunteers; we highlighted a view instances in this chapter so that one can see how the parameters and penalties effect the number of people volunteering each year. We simulated several representative scenarios to highlight how changing the penalties or parameters the model makes different assignments. To learn more about the effects of penalties and rates a sensitivity analysis could be applied to the simulation and larger problems would need to be investigated. This is a good starting point for researchers to continue this preliminary investigation of the potential impact of volunteer assignments and how they effect future volunteer involvement. Understanding the effect that assignments have on a volunteer returning or inviting someone is why we conduct this investigation. Since the model makes the assignments investigating the impact of these assignments is an interesting extension of the volunteer assignment problem.

5. Application to Denver Bike Sharing

Denver will be the first U.S. city to have a bike sharing program as large as other programs around the world. Kiosks are located throughout the city which allows a person to ride a bike and drop it off at different locations. The vision of Denver Bike Sharing is to change how people get around in Denver. The program promotes a healthier life style and an alternative way of getting around the city.

The Denver Bike Sharing volunteer assignment problem consisted of assigning volunteers for three different shifts to 35 kiosks that were located in 7 geographical areas throughout Denver for three days. On April 22, 2010, Denver B-Cycle launched Denver Bike Sharing.

We helped solve the Denver Bike Sharing volunteer assignment problem by applying our model. We collected information from the volunteers and worked very closely with the volunteer coordinator Piep Van Heuven. Participants were required to fill out a volunteer survey; to provide information on what shifts they were available, how much they prefer to work at each of the locations, and the number of shifts they are willing to work. The volunteer coordinator was asked how many kiosks needed volunteers, how many volunteers were needed at each kiosk, and what days were we assigning volunteers for. The volunteer coordinator determined that a volunteer can work one shift, however with a preference that they work at least two shifts. This information was used to formulate the goal programming model we created. However, after the model output the assignments the volunteer coordinator realized that some people had a large number of shifts and she wanted to spread the jobs more evenly. This was accomplished by adding an additional constraint to our model that did not allow a volunteer to be assigned more than 4 shifts.

A description of the sets, variables, and parameters used in the model is given in table 3.1.

Decision Variables

Decision variables x_{ijk} and y_i are described in section 3.1.

Constraints

A volunteer can only be assigned to at most one kiosk during each shift if they are available, because we assume a person can not be at two different kiosks at the same time:

$$\sum_j x_{ijk} \leq \alpha_{ik} \;\; \forall\, i, k.$$

Volunteers that are assigned to the first shift need to be able to ride a bike because the volunteers were riding bikes in the parade that launched Denver Bike Sharing:

$$x_{ij1} \leq \beta_{ij} \;\; \forall\, i, j$$

where

$$\beta_{ij} = \begin{cases} 1 \text{ if volunteer } i \text{ can ride a bike;} \\ 0 \text{ otherwise.} \end{cases}$$

The model needed to assign the required number of volunteers to each kiosk during each shift:

$$\sum_i x_{ijk} = \mu_{jk} \;\; \forall\, j, k.$$

Assigning volunteer i the ideal number of shifts, if any at all:

$$\sum_{jk} x_{ijk} + v_i^- - v_i^+ = y_i \nu_i \ \ \forall\, i.$$

Assign each volunteer the minimum number of shifts required to work:

$$\sum_{jk} x_{ijk} + w_i^- - w_i^+ = 2y_i \forall\, i.$$

Do not allow a volunteer to be assigned to more than 4 shifts:

$$\sum_{jk} x_{ijk} \leq 4 \ \ \forall\, i.$$

A volunteer cannot work a consecutive shift at a conflicting area, where we define a conflict to be kiosks that are located at different locations located too far apart:

$$x_{ijk} + x_{ij'k+1} \leq 1 \ \ \forall\, i, k,\, (j, j') \in C.$$

A volunteer can not work 3 consecutive shifts:

$$\sum_{j} x_{ijk} + \sum_{j} x_{ijk+1} + \sum_{j} x_{ijk+2} \leq 2 \ \ \forall\, i,\ k = 1, ..., S - 2.$$

Objective Function

The objective was to make an assignment in such a way that maximized the preferences of the volunteers while satisfying all of the needs of the volunteer coordinator. The objective function was penalized if the model did not assign the volunteers their ideal number of shifts requested. We choose to use a penalty of 10 for assigning a volunteer more or less shifts than they signed up for because we wanted to choose a number larger than the maximum preference indicated to encourage the model to assign the volunteers their ideal number of shifts. We could have chosen any positive number, however we choose appropriate penalties to get an assignment that the volunteer coordinator approved. We penalized the objective function if we assigned a volunteer only one job and not at least two. We made the penalty larger for assigning a volunteer less than two jobs because the volunteer coordinator determined that it was more important than assigning a volunteer a different number of shifts than their ideal number. Mathematically, the bojective function is:

$$\max \sum_{ijk} x_{ijk}\pi_{ijk} - 10\sum_i v_i^- - 10\sum_i v_i^+ - 20\sum_i w_i^-.$$

Denver Bike Sharing was able to use the assignments that our model made. Piep was very impressed with the final schedule produced and pleased with the amount of time the model saved her as well. A letter from the client can be found in Appendix A. We produced a master schedule in Excel and created an on-call volunteer list that the volunteer coordinator used to make assignments the day of if there was a cancellation. The master schedule was sent out to all of the volunteers and was available on line. A few requests came in the last minute so we made a minimal number of manual adjustments.

6. Conclusion

We propose a two-phase modeling approach to solve the volunteer assignment problem. The first phase is used to tell volunteer coordinators if an assignment of volunteers is possible, given the number of qualified volunteers that are currently signed up and that are needed for every job during each shift. The second phase makes the assignment of volunteers. Our two-phase approach to the volunteer assignment problem is very helpful in determining if there exists a feasible assignment and making the assignment if one exists. To our knowledge this is the first time anyone in the literature presents solving the volunteer assignment problem this way.

We construct a simulation program to observe how assignments effect volunteer's future involvement. A series of test scenarios show that the penalties and rates affect the number of volunteers being assigned and the number of volunteers coming back. The insight gained from the simulation allows the modeler to choose appropriate penalties to accomplish what each volunteer coordinator wants to achieve.

Our two-phase model assigned the volunteers for the launch of Denver B-Cycle, Denver's new bike sharing program. A master schedule and an on call volunteer list were produced. The volunteer coordinator determined that the model was able to make fair, efficient, and timely assignments.

6.1 Future Work

This thesis offers many avenues for further research. Applying the model to larger problems is a possible extension. Determining how satisfied volunteers are given their assignments will allow us to justify rates used in our simulation. One can apply the two-phase modeling approach to other volunteer assignment problems as well as find new assignment problems where the model may be applicable. Building on the simulation and doing sensitivity analysis would be worthwhile as well.

APPENDIX A. BikeDenver Client Letter

BikeDenver
A Voice for Denver Cyclists
Box 801, 1536 Wynkoop St., Denver, CO 80202
bikedenver.org

July 13, 2010

To Whom It May Concern:

It was a pleasure to work with graduate student Matt Kaspari of the University of Colorado at Denver's Department of Mathematics and Statistical Sciences on the volunteer coordination project for the Denver Bike Sharing launch this April.

BikeDenver was contracted to provide volunteer services on a complex multi-day event to support the launch of Denver's new bike share program this April and working with Matt helped simplify the process enormously! My challenges included:

- **Placing volunteers at multiple locations during 3 shift times over a 3-day period.**
- **Placing volunteers during dates and times and at locations that fit their availability and interest.**
- **Identifying "experienced" volunteers and placing them at high-profile or high-traffic locations before assigning less experienced volunteers.**
- **Restricting the number of shifts that a volunteer could sign up for in order to stay in keeping with the community volunteer effort concept that would engage more people in the project.**
- **Identifying volunteers willing to be "on-call" on certain days.**
- **Communicating the complex master schedule once placements were made.**

The math model Matt created took all of these complex considerations into account and helped us assign over 300 volunteers in a manner that was fair, quick, and easy to explain.

It was a pleasure to work with Matt on the project and the volunteer effort proved to be successful even given some weather events that required us to cancel one day and re-schedule all volunteers to the following Saturday. Please contact me at piep@bikedenver.org if I can provide any additional information about the project.

Sincerely,

/s/ Piep van Heuven

Piep van Heuven
Executive Director, BikeDenver

APPENDIX B. BikeDenver Volunteer Survey

BikeDenver Volunteer Application Survey

Exit this survey

BikeDenver Volunteer Application

1. Denver B-Cycle Launch Days Application

All volunteers that want to be eligible for a job will need to answer all of the survey questions. We will be using a mathematical model to make the volunteer assignments. Preference will be given to those volunteers that sign up for at least 2 shifts.

*** 1. CONTACT INFORMATION**

First Name

Last Name

Street Address

City

State

Zip Code

Daytime Phone Number

Night Time Phone Number

Email Address

*** 2. HELP US GET TO KNOW YOU**

	Yes	No
Do you volunteer for non-profit groups or events at least once a year?		
Did you volunteer at the DNC Freewheelin' bike share project?		
Have you volunteered for a bike group or bike event in the past year (not		

including Freewheelin')?	○	○
Did you participate in the Denver Bike Initiative?	○	○
Are you personally acquainted with a staff member at Denver Bike Sharing?	○	○
Have you taken a League of American Bicyclist's education class?	○	○
Do you work in the outdoor industry, or for a group that encourages outdoor activities?	○	○
Do you work in a professional field with an emphasis on alternative modes of transportation?	○	○
Are you an appointed member of the Mayor's Bicycle Advisory Committee?	○	○
Would you be willing to ride a bicycle for 3 or more miles?	○	○

✱ 3. What t-shirt size and style do you wear?

	Small	Medium	Large	X Large
Men	○	○	○	○
Women	○	○	○	○

✱ 4. What time(s) are you available to work?
(check yes for all that apply)
***Preference will be given to volunteers committed to two or more shifts.**

	Yes	No
Thursday 4/22 Morning (7:00am - 10:30am)	○	○
Thursday 4/22 Afternoon (11:00am - 1:00pm)	○	○
Thursday 4/22 Evening (4:00pm - 6:00pm)	○	○
Friday 4/23 Morning (7:00am - 9:00am)	○	○
Friday 4/23 Afternoon (11:00am - 1:00pm)	○	○
Friday 4/23 Evening (4:00pm - 6:00pm)	○	○
Saturday 4/24 Morning (9:00am - 11:00am)	○	○
Saturday 4/24 Afternoon (11:00am - 2:00pm)	○	○
Saturday 4/24 Evening (2:00pm - 5:00pm)	○	○
Monday 4/26 Morning (7:00am - 9:00am)	○	○
Monday 4/26 Afternoon (11:00am - 1:00pm)	○	○
Monday 4/26 Evening (4:00pm - 6:00pm)	○	○

✱ 5. Are you available to be an "on-call" volunteer for any of the days?

	Yes	No
Thursday 4/22	○	○
Friday 4/23	○	○

Saturday 4/24	○	○
Monday 4/26	○	○

*** 6. For the following areas of town determine how willing you are to work at the particular area.**

	Highly Desirable	Desirable	Indifferent	Not Desirable	Unable
University of Denver	○	○	○	○	○
Cherry Creek	○	○	○	○	○
Capitol Hill	○	○	○	○	○
Central Downtown	○	○	○	○	○
Union Station	○	○	○	○	○
North Downtown	○	○	○	○	○
Highlands	○	○	○	○	○

*** 7. Number of Shifts**

	1	2	3	4	5	6	7	8	9	10	11	12
What is the minimum number of shifts you are willing to work?	○	○	○	○	○	○	○	○	○	○	○	○
Ideally how many shifts do you want to work?	○	○	○	○	○	○	○	○	○	○	○	○
What is the maximum number of shifts you are willing to work?	○	○	○	○	○	○	○	○	○	○	○	○

*** 8. I WILL ATTEND ONE OF THE MANDATORY VOLUNTEER TRAINING SESSIONS***

- ○ Monday 4/19 (6:00pm - 7:30pm)
- ○ Tuesday 4/20 (6:00pm - 7:30pm)
- ○ Wednesday 4/21 (6:00pm - 7:30pm)

*** 9. YOUR COMMITMENT**

	Yes	No
I understand that only 2 volunteers will be assigned per kiosk per shift and I commit to volunteering when and where I am assigned.	○	○
I will be on time for my assigned shifts.	○	○
I understand that I will be representing Denver Bike Sharing and the City of Denver at a high profile event and I will dress and act in a professional manner.	○	○

10. Do you have any other comments?
(ex. I want to work with Sally. I can not lift heavy things.)

APPENDIX C. Simulation Program

```
%This function writes an ampl data file that we use in VolAssignGP.mod



function [obj_avg, vol_avg, vol_sat_avg] = Program(V, J, S, N)

rand('seed',10);


%%%%% Preference, Availability, Max Volunteer, Min Volunteer, Max Shift


Pref = round(rand(V,J)*5);

Avail = ceil(rand(V,S));


for i = 1:J;

    for j = 1:S;

     MinVol(i,j) = 1;

     MaxVol(i,j) = 1;

    end

end


MaxShift = ceil(rand(1,V)*V);


%%%%%vCreate file to write to
```

```
fn = 'Program.dat';

fid = fopen(fn,'w'); %The 'w' dentoes writing privilages.



%%%%%Create Sets

fprintf(fid,'set VOL := ');

for i = 1:V

    fprintf(fid, 'v%d \t', i);

end

fprintf(fid, '; \n');


fprintf(fid,'set JOBS := ');

for i = 1:J

    fprintf(fid, 'j%d \t', i);

end

fprintf(fid, '; \n');


fprintf(fid,'set SHIFTS := ');

for i = 1:S

    fprintf(fid, 's%d \t', i);

end

fprintf(fid, '; \n \n');
```

```
%%%%% Create param preference
fprintf(fid,'param preferences: \t');
for j = 1:J
    fprintf(fid, 'j%d \t', j);
end
fprintf(fid, ':= \n');


for i = 1:V
    fprintf(fid, 'v%d \t\t\t', i);
    for j = 1:J
        fprintf(fid, '%d \t', Pref(i,j));
    end
    if i == V
        fprintf(fid, '; \n \n');
    else
        fprintf(fid, '\n');
    end
end


%%%%% Create param availability
fprintf(fid,'param availability: \t');
for k = 1:S
    fprintf(fid, 's%d \t', k);
end
```

```
fprintf(fid, ':= \n');

for i = 1:V
    fprintf(fid, 'v%d \t\t\t', i);
    for k = 1:S
        fprintf(fid, '%d \t', Avail(i,k));
    end
    if i == V
        fprintf(fid, '; \n \n');
    else
        fprintf(fid, '\n');
    end
end

%%%%% Create param MaxVol
fprintf(fid,'param maxVol: \t');
for k = 1:S
    fprintf(fid, 's%d \t', k);
end
fprintf(fid, ':= \n');

for j = 1:J
    fprintf(fid, 'j%d \t\t', j);
    for k = 1:S
```

```
        fprintf(fid, '%d \t', MaxVol(j,k));
    end
    if j == J
        fprintf(fid, '; \n \n');
    else
        fprintf(fid, '\n');
    end
end

%%%%% Create param MinVol
fprintf(fid,'param minVol: \t');
for k = 1:S
    fprintf(fid, 's%d \t', k);
end
fprintf(fid, ':= \n');

for j = 1:J
    fprintf(fid, 'j%d \t\t', j);
    for k = 1:S
        fprintf(fid, '%d \t', MinVol(j,k));
    end
    if j == J
        fprintf(fid, '; \n \n');
    else
```

```
        fprintf(fid, '\n');
    end
end

%%%%% Create param Penalty
fprintf(fid,'param penalties:= 1 6 2 6 3 6; \n');

%%%%% Create param MaxShift
fprintf(fid,'param maxShift: \t');
for i = 1:V
    fprintf(fid, 'v%d \t', i);
end
fprintf(fid, ':= \n');

for j = 1
    fprintf(fid, 'numShift%d \t\t', j);
    for i = 1:V
        fprintf(fid, '%d \t', MaxShift(j,i));
    end
    if j == 1
        fprintf(fid, '; \n \n');
    else
        fprintf(fid, '\n');
    end
```

```
end

fclose(fid);

%this runs ampl
! ./Program.bash

%%%This prints zp
z = load('-ascii', 'zp.mat');

%%we need these for the first loop
newV = V;
newPref = Pref;
zNEW = z;

%%%%%%%%%%%%Run ProgramNew%%%%%%%%%%%%%%%%%%%%%
for y = 1:N %where N =  number of years we want to run our simulation
a = zeros(newV,J);
w = zeros(newV,J);
for i = 1:S;
    a = a + zNEW((((i-1)*newV)+1):(i*newV), 1:J);
end

for i = 1:newV;
```

```
    for j = 1:J;
        w(i,j) = a(i,j)*newPref(i,j);
    end
end

% # of shifts a volunteer works
numShiftsVol = [];
sumWrow = [];
for i = 1:newV;
    numShiftsVol(i,1) = sum(a(i,:));
end

% Sum of the job preferences of the jobs the volunteer works
for i = 1:newV;
      sumWrow(i,1) = sum(w(i,:));
end

%intialize
percentage = [];
%this calculates the % of satisfaction each volunteer is
for i = 1:newV;

    if numShiftsVol(i,1) >= 1;
        percentage(i,1) = sumWrow(i,1) / (numShiftsVol(i,1)*max(newPref(i,:)))
```

```
    else
        percentage(i,1) = .25;

    end
end

%%intialize
r = [];
q = [];
%find out what volunteer is coming back
r = 1 - .9*percentage;
for i = 1:newV;
    if newV > 0;
        q = rand(newV,1);
    else
    q(i,1) = 0;
    end
end

dif = q-r;
comeBack = find(dif > 0);
sizeCB = size(comeBack,1);
Vhold = [];
for i = 1:newV;
```

```
    if dif(i,1) > 0;
      Vhold = [Vhold; newPref(i,:)];
    end
end

%find out what volunteer is invited
if newV > 0

percentage2 = .5*percentage;
r2 = 1 - percentage2;
q2 = rand(newV,1);
dif2 = q2-r2;
invite = find(dif2 > 0);
sizeI = size(invite,1);

else
    invite = 0;
    sizeI = 0;
end

%generate new preference, avail, maxshift matrices
Vnewhold2 = [];
if sizeI > 0;
    for i = 1:sizeI
```

```
        Vnewhold2 = round(rand(i,J)*5);
    end
    newPref = [Vhold;Vnewhold2];
    else newPref = Vhold;
end

%%%if no one is coming back and no one is invited bring in a new volunteer
if size(newPref,1) == 0;
    newPref = round(rand(1,J)*5);
    newV = 1;
    newAvail = ones(1,S);
    newMaxShift = ones(1,1);
else
    newV = size(newPref,1);
    newAvail = ones(newV,S);
    newMaxShift = ones(1,newV)*S;
end

%%%%%need to get the right 0,1 variable for ampl to solve
sumNewAvail = [];
shiftNewAvail = [];
for i = 1:newV
    for k = 1:S
    sumNewAvail(i,k) = sum(newAvail(i,:));
```

```
    if sumNewAvail >= 1;
   shiftNewAvail(i,k) = 1;
    else
        shiftNewAvail(i,k) = 0;
    end
    end
end



%%%%%Vector for plot
plot_newV(1,y) = [newV];

%%%%%vCreate file to write to


fn = ['ProgramNew','.dat'];
fid = fopen(fn,'w'); %The 'w' dentoes writing privilages.



%%%%%Create Sets
fprintf(fid,'set VOL := ');
for i = 1:newV
    fprintf(fid, 'v%d \t', i);
end
fprintf(fid, '; \n');
```

```
fprintf(fid,'set JOBS := ');
for i = 1:J
    fprintf(fid, 'j%d \t', i);
end
fprintf(fid, '; \n');

fprintf(fid,'set SHIFTS := ');
for i = 1:S
    fprintf(fid, 's%d \t', i);
end
fprintf(fid, '; \n \n');

%%%%% Create param preference
fprintf(fid,'param preferences: \t');
for j = 1:J
    fprintf(fid, 'j%d \t', j);
end
fprintf(fid, ':= \n');

for i = 1:newV
    fprintf(fid, 'v%d \t\t\t', i);
    for j = 1:J
        fprintf(fid, '%d \t', newPref(i,j));
```

```
    end
    if i == newV
        fprintf(fid, '; \n \n');
    else
        fprintf(fid, '\n');
    end
end

%%%%% Create param availability
fprintf(fid,'param availability: \t');
for k = 1:S
    fprintf(fid, 's%d \t', k);
end
fprintf(fid, ':= \n');

for i = 1:newV
    fprintf(fid, 'v%d \t\t\t', i);

    for k = 1:S
        fprintf(fid, '%d \t', shiftNewAvail(i,k));
    end
    if i == newV
        fprintf(fid, '; \n \n');
    else
```

```
        fprintf(fid, '\n');
    end
end

%%%%% Create param MaxVol
fprintf(fid,'param maxVol: \t');
for k = 1:S
    fprintf(fid, 's%d \t', k);
end
fprintf(fid, ':= \n');

for j = 1:J
    fprintf(fid, 'j%d \t\t', j);
    for k = 1:S
        fprintf(fid, '%d \t', MaxVol(j,k));
    end
    if j == J
        fprintf(fid, '; \n \n');
    else
        fprintf(fid, '\n');
    end
end

%%%%% Create param MinVol
```

```
fprintf(fid,'param minVol: \t');
for k = 1:S
    fprintf(fid, 's%d \t', k);
end
fprintf(fid, ':= \n');

for j = 1:J
    fprintf(fid, 'j%d \t\t', j);
    for k = 1:S
        fprintf(fid, '%d \t', MinVol(j,k));
    end
    if j == J
        fprintf(fid, '; \n \n');
    else
        fprintf(fid, '\n');
    end
end

%%%%% Create param Penalty
fprintf(fid,'param penalties:= 1 6 2 6 3 6; \n');

%%%%% Create param MaxShift
fprintf(fid,'param maxShift: \t');
for i = 1:newV
```

```
    fprintf(fid, 'v%d \t', i);
end
fprintf(fid, ':= \n');

for j = 1
    fprintf(fid, 'numShift%d \t\t ', j);
    for i = 1:newV
        fprintf(fid, '%d \t', newMaxShift(j,i));
    end
    if j == 1
        fprintf(fid, '; \n \n');
    else
        fprintf(fid, '\n');
    end
end

fclose(fid);

%%This runs external program ampl
! ./ProgramNew.bash

%%%This prints z2
zNEW = load('-ascii', 'zNew.mat');
%%%%% need to make zNEW work for our plots
```

```
for k=1:S
    for i=1:newV
        for j=1:J
            Znew(i,j) = zNEW((i+((k-1)*newV)),j);
        end
    end
 %%%Get objective function value so we can plot
            objMat = times(newPref,Znew);
            objValrow = sum(objMat);
            objValShift = sum(objValrow);
            objVal = sum(objValShift);
            objectiveValue(1,k)= objVal;

Znew=[]; %reintialize...program needed it
end
 objectiveValueSum = sum(objectiveValue);
 plot_objVal(1,y) = objectiveValueSum;

%%%Printing Cell Arrays of data
NEWpref{y} = newPref;
NEWv{y} = newV;
NEWavail{y} = newAvail;
NEWmaxshift{y} = newMaxShift;
```

```
%%%Print results to file so that we can retreive later
fn = ['ProgramNew_info','.dat'];
fid = fopen(fn,'w');    %The 'w' dentoes writing privilages.
fprintf(fid,'newV = ');
fprintf(fid,'%g', newV);
fprintf(fid, '\n');
fprintf(fid,'newAvail = ');
fprintf(fid,'%g ', newAvail);
fprintf(fid, '\n');
fclose(fid);

%%%reinitialize everything

end % end of for loop ProgramNew
obj_avg = sum(plot_objVal)/size(plot_newV,2)
vol_avg = sum(plot_newV)/size(plot_newV,2)
vol_sat_avg = obj_avg / vol_avg

%%%%%%help nargin%%%%%this will allow me to
%%%%plot number of volunteers at event
figure
subplot(2,1,1);
plot(plot_newV)
```

```
xlabel('Year')
ylabel('# of Volunteers')
title('Volunteer Pool Over Time')

%%%%%%plot objective function/newV this will give us the average happiness of
avgHap = (plot_objVal./plot_newV);
subplot(2,1,2);
plot(avgHap)
xlabel('Year')
ylabel('Preference Level')
title('Average Satisfaction Volunteers')

end
```

REFERENCES

[1] R.H. Ahuja, T.L. Magnanti, and J.B. Orlin. Some recent advances in network flows. *Society for Industrial and Applied Mathematics*, 33(2):175–219, 1991.

[2] R.K. Ahuja, Magnanti T.L., and Orlin J.B. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

[3] D.P. Bertsekas. The auction algorithm. *Annals of Operations Research*, 14(1):105–123, 1988.

[4] J.G. Chapman and R. Morley. Collegiate service-learning: Motives underlying volunteerism and satisfaction with volunteer service. *Journal of Prevention and Intervention in the Community*, 18(1):19–33, 1999.

[5] A. Charnes, W.W. Cooper, and R.O. Ferguson. Optimal estimation of executive compensation by linear programming. *Management Science*, 1(2):138–151, 1955.

[6] E.G. Clary and M. Snyder. The motivations to volunteer: Theoretical and practical considerations. *Current Directions in Psychological Science*, 8(5):156–159, 1999.

[7] R.A. Cnaan and L. Amrofell. Mapping volunteer activity. *Nonprofit and Voluntary Sector Quarterly*, 23(4):335–351, 1994.

[8] J.E. Crandall and M.D. Harris. Social interest, cooperation, and altruism. *Journal of Individual Psychology*, 32:50–54, 1976.

[9] G.B. Dantzig. A comment on edie's "traffic delays at toll booths". *Journal of the Operations Research Society of America*, 2(3):339–341, 1954.

[10] F.F. Easton and D.F. Rossin. A stochastic goal program for employee scheduling. *Decision Sciences*, 27(3):541–568, 1996.

[11] R. Emanele. Is there a (downward sloping) demand curve for volunteer labour. *Annals of Public and Cooperative Economics*, 67(2):193–208, 1996.

[12] R.B. Freeman. Working for nothing: The supply of volunteer labor. *The Journal of Labor Economics*, 15(1):S140–S166, 1997.

[13] L. Gordon and E. Erkut. Improving volunteer scheduling for the edmonton folk festival. *Interfaces*, 34(5):367–376, 2004.

[14] S.K. Green et al. Volunteer motivation and its relationship to satisfaction and future volunteering. *Paper presented at the Annual Convention of the American Psychological Association*, 0(0):1–15, 1984.

[15] H.J. Greenberg and T. (eds.) Morrison. Preparing for the democratic national convention. Mathematics Clinic Report MC08S001, University of Colorado Denver, Denver, CO, 2008.

[16] K.F. Lam and E.U. Choo. Goal programming in preference decomposition. *The Journal of the Operational Research Society*, 46(2):205–213, 1995.

[17] V.A. Mabert and C.A. Watts. A simulation analysis of tour-shift construction procedures. *Management Science*, 28(5):520–532, 1982.

[18] R.L. Ryan, R. Kaplan, and R.E. Grese. Predicting volunteer commitment in enviromental stewardship programmes. *Journal of Enviromental Planning and Mangement*, 44(5):629–648, 2001.

[19] S.E. Sampson. Optimization of volunteer labor assignments. *Journal of Operations Management*, 24(4):363–377, 2006.

[20] S. Shin and B.H. Kleiner. How to manage unpaid volunteers in organisations. *Management Resource News*, 26(2-4):63–71, 2003.

[21] D.H. Smith. Determinants of voluntary association participation and volunteering: A literature review. *Nonprofit and Voluntary Sector Quarterly*, 23(3):243–263, 1994.

[22] G.M. Thompson. Labor scheduling using npv estimates of the marginal benefit of additional labor capacity. *Journal of Operations Management*, 13(1):67–86, 1995.