OPTIMAL PERSONNEL SCHEDULING UNDER UNCERTAINTY USING CONDITIONAL VALUE-AT-RISK WITH APPLICATION TO HOSPITAL PHARMACIST TIMETABLE ASSIGNMENT PROBLEMS

By

JOHN ROBERT STRIPLING IV

A THESIS PRESENTED TO THE GRADUATE SCHOOL
OF THE UNIVERSITY OF FLORIDA IN PARTIAL FULFILLMENT
OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

UNIVERSITY OF FLORIDA

2010

I dedicate this thesis to the US Air Force whom has given me many great opportunities and has pushed me to complete my master’s degree while working full time.

ACKNOWLEDGMENTS

I would like to first acknowledge Dr. Boginski for his steadfast encouragement and support throughout my thesis research and development. Second, I would like to thank Judi Shivers. She has enabled active duty military students to pursue their higher education despite a very challenging environment of deployments, multiple out of area business trips, and occasionally confusing bureaucracy. I thank Dr. Pardalos for supporting my work on this thesis and becoming a co-chair on my thesis defense board. Finally, I would like to acknowledge my family and friends. The unending support and encouragement I receive from all of you is one of the great blessings in my life.

TABLE OF CONTENTS

APPENDIX

LIST OF TABLES

LIST OF FIGURES

Abstract of the Thesis Presented to the Graduate School
of the University of Florida in Partial Fulfillment of the
Requirements for the Degree of Master of Science

OPTIMAL PERSONNEL SCHEDULING UNDER UNCERTAINTY USING CONDITIONAL VALUE-AT-RISK WITH APPLICATION TO HOSPITAL PHARMACIST TIMETABLE ASSIGNMENT PROBLEMS

By

John Robert Stripling IV

December 2010

Chair: Vladimir Boginski
Major: Industrial and Systems Engineering

Mathematical optimization has become an indispensable tool in many organizations to drive improved efficiencies. There are many techniques to optimize processes mathematically in order to minimize or maximize a given objective function. Conditional Value-at-Risk methodology has become a popular new tool in the optimization community. Implementing a stochastic view of optimization, it permits the confidence level of a solution to be part of the optimization process. This theory is based around the Conditional Value-at-Risk measure which provides an upper bound on the allowed losses or excesses of a solution. We show that it can be applied to general personnel timetable assignment problems to provide a robust and useful optimization solution. This developed optimization techniques allow management level confidence in personnel scheduling solutions and accounts for real world uncertainties facing organizations throughout the world.

One industry that must contend with variable demand and is being driven to cut personnel costs is hospitals. One aspect of hospital manning that makes a great candidate for personnel timetable optimization is the hospital pharmacy. Most hospitals

in the United States have their own pharmacy to handle the large and varied needs of their patient population. These pharmacies employ a variety of different workers. The most expensive employees and ultimately the driving force in the pharmacy are the pharmacists. Their job is to process, monitor, give guidance, and distribute medicine for the hospital's patients. One such hospital with a pharmacy that we considered as a benchmark case study in this thesis is a hospital located in Marietta, Ga. Working with the hospital management, we have been able to get a statistically relevant amount of data on how the pharmacy operates which has opened the door to applying useful optimization techniques to this established way of doing business.

In this study, we utilized Conditional Value-at-Risk constraints in the context of a personnel scheduling integer programming problem to derive a robust solution. The results of the study produced a schedule that saves a substantial amount of money per year while better covering high demand times of the pharmacy. The result of this study is a new optimized schedule that shows the efficiency of using Conditional Value-at-Risk constraints in the considered settings.

CHAPTER 1
OVERVIEW OF PERSONNEL SCHEDULING OPTIMIZATION

1.1 History

There has been a great deal of work regarding personnel timetable assignments, or personnel scheduling for short, optimization, often with a basic goal of simply automating the work that goes into scheduling. Many personnel schedules must be created on a weekly basis in a very dynamic environment where employee availability and business demands are an ever changing variable. Studies have shown that it takes a manager between 8 and 14 hours to schedule 70-100 employees for one week [7]. There is even data showing that hospitals in particular involved 10% of their personnel in the scheduling process [11]. Advanced studies have shown that many medium to large sized organizations find that they have established local optimums in particular departments at the cost of developing larger scale, organization wide schedule optimums [17]. There is great need and demand for optimization. Reducing labor cost, particular unnecessary overtime charges, while still meeting the high demands of the business world are goals of any successful organization.

Linear and integer programming have often been the method of choice to optimize scheduling problems in the last 20 years. Though linear and integer programming are powerful methods, they do have limitations. Many of the implemented integer programming models involved 100 to 500 variables. Typically these variables represent a reduced set of variables than are actually present in the scheduling equation, for example not putting in inputs such as 15 minute breaks or employee preferences [7]. This shows that often the optimized schedules are based on incomplete models and fail to account for enough factors to produce the best schedule.

There are also examples of scheduling problems exceeding the capabilities of popular software optimization tools.

One limitation of personnel schedule optimization work has shown that human schedulers themselves don't trust the solution. Schedulers typically want evaluation tools that show possibilities and can warn of mistakes made in the human schedule versus a mathematically optimized schedule. This shows that in practice, human schedulers are usually unwilling to allow the process to be totally automated. Additionally human schedulers almost always want the ability to ignore certain predefined constraints for different situation which form the basis of optimization solution [17]. Though this is the reality of trying to solve a personnel scheduling problem there is a good bit that can be done to work within this real world limitation.

1.2 Personnel Scheduling Methods

One approach that has been used to enhance mathematical programming as a scheduling optimization approach is to utilize complex dissatisfaction criteria in the formula to optimize. Using mathematical programming, one can adjust weights of certain rules, estimated employee dissatisfaction with particular schedules, estimated profits, and ability to match organizational demands to find an optimum schedule that is based on a big picture view of the operation. Focusing the scheduling problem on dissatisfaction of the various parties involved versus a more simplified optimum schedule allows the solution to be more applicable in the real world [8]. Management is then able to adjust the different weighting as various factors change over time to better represent the ever changing priorities of an organization and create a personnel schedule that meets these priorities.

Weighting of dissatisfaction is a management decision highlighting how important management involvement is in creating effective scheduling tools. Such an approach encompasses individual dissatisfactions, such as higher weight for a more senior employee over a junior employee or a particular customer that demands a service or good on a more restrictive timeline. The more specific criteria is added to the formula the more complex and large mathematical program gets [8]. Issues with computational time, as well as the amount of effort and investment required, grows quickly as the scheduling problem includes more and more factors.

A second approach to personnel scheduling optimization that has been utilized in recent years is implementing artificial intelligence into scheduling optimization. This is primarily accomplished by fusing management science with modern artificial intelligence developments. This fusing permits a macro view of optimization, particularly large optimization problems, to be implemented during the optimization process [7]. Multiple criteria can be optimized separately then viewed as a collection to determine where local minimums have been created based on particular criteria. Each criterion has a weight assigned to it that aids in the selection of the overall optimum solution. Implemented with a limited decision memory this approach creates a tolerance for bad optimization decisions within the optimization process.

The artificial intelligence approach also allows for new constraints to be added within the optimization problem, leading to a more efficient optimization as additional schedules are solved [7]. This approach develops an improving process and can avoid some of the local minimum pitfalls of other scheduling optimization methods while actually getting better over time. This method is very sophisticated and requires the use of complex artificial intelligence to implement.

Another popular approach that is used in modern scheduling optimization is a Bayesian Stochastic model [9]. The advantage to this method is that it takes variance from the schedule into account in the optimum solution. The model can be seen as a standard mathematical programming model with two additions. It begins with Bayesian forecasting of the system's statistics to allow for an approximation of the system variance on minimal date. It then adds a stochastic analysis of what data should be used to give reasonable confidence in the optimization through a normal linear program. The use of effective forecasting strengthens the stochastic model's validity. The stochastic model's ability to take into account the variances of the system being optimized through personnel scheduling creates an important real life consideration in the optimization [9].

The strength of this method is that it allows the overall method to change over time and still account accurately for the variance of the process. This is accomplished by constraining the data to recent data and using forecasting to enhance the results in order to determine the statistical characteristics of the process. In addition, the stochastic optimization allows the scheduled solution to better cover the full range of demands and constraints and not just a point prediction.

1.3 Optimization of Conditional Value-at-Risk

The approach selected for this study is the conditional value-at-risk optimization of personnel timetable assignments. This is a recent development in optimization that allows risk levels to be set while using a stochastic solution to the random variables that are present in most optimization problems. This approach focuses on the minimizing losses, or when applied to personnel scheduling minimizing the understaffing, based on the variance of the overall system [14]. Conditional Value-at-Risk optimization was primarily developed for financial and insurance industry purposes but this study shows that the method expands to personnel scheduling optimization as well. This method was chosen to develop an optimized schedule that was not only efficient but would also prevent a major shortfall in critical health care that can occur in the hospital environment. Further details of this method are described later in this document.

CHAPTER 2
OVERVIEW OF LINEAR AND INTEGER PROGRAMMING

2.1 Linear Programming Overview

Linear Programming is a mathematical method of solving optimization problems of many different kinds. Linear programming was made possible in 1947 when George Dantzig developed an algorithm to solve linear programming problems called the simplex method [16]. Put simply, linear programming involves determining values for a set of decision variables which will minimize or maximize an object function subject to a set of constraints [12]. Through a rather straightforward process, the requirements and goals of real world optimization problems can be put into mathematical format [6]. There are limitations to this approach but overall linear optimization is a very popular method used across a broad spectrum of optimization problems.

The simplex method is the method of choice for automating linear programming problems in a computer program. Rather elaborate and complex optimization problem can be solved in minutes and often in mere seconds. The algorithm requires that the objective function, which is maximization or minimization goal of the optimization problem, and the constraints are put into what is called standard form. Standard form is an ordered series of equations translated by the system engineer from object to be optimized and the restraints of the problem [8]. Once the optimization problem is put into standard form the coefficients and products are converted into a matrix. The first step in the simplex method, once this matrix has been formed from the standard form, is to find what is known as an entry variable of the matrix. The entry variable is selected based on row 0’s, the objective function’s, largest positive coefficient. A ratio test of each row following row 0 is performed. The row with the largest ratio is determined based on the coefficient of the row entry variable selected from row 0 divided into the row solution. The row with the smallest positive ratio becomes the pivot row [16]. The matrix is manipulated using elementary row operations focusing on the pivot to create a modified matrix where the entry variable is 1 in the pivot row and 0 in all other now 0 rows [16]. This process is repeated until all basic variables have been discovered and optimized.

2.2 Integer Programming Overview

The difference between integer programming and linear programming is that the input variables to the linear program must be integers. At first glance this may seem somewhat trivial. In cases where the optimum solution inputs are very small, such as 2.2 employees to work a shift, the usefulness of the linear programming optimization without integer restrictions can become minimal. Restricting some or all of the input variables to integers transforms the program into an integer programming problem. Restricting some but not all of the input variables to integers is also known as a mixed integer programming [16]. Integer programming is computed differently than linear programming and has a number of different mathematical properties.

Integer programming and mixed integer programming are generally categorized in a computational class called nondeterministic polynomial time (NP) hard problems. NP-hard problems cannot be easily solved computationally and often can become unsolvable due to the very large computational time to produce a solution. Integer programming requires heuristic search methods to determine an optimum versus a convergent search method seen in most linear programming algorithms [7]. For optimization problems that have large number of variables, selecting an integer or mixed integer programming approach may make the solution impossible to produce with current technology.

There are a number of algorithms that have been developed over the years to solve integer problems. Three of the most popular algorithms that are currently used are the Branch-and-Bound method, Implicit Enumeration, and the Cutting Plane algorithm. In general the Branch-and-bound method is best used for standard integer programming, implicit enumeration is best used for Boolean integer programming, and cutting plane is best used for mixed integer programming [16]. LINGO, the program used to solve the optimization problem set forth in this thesis, and in most integer program solving software, uses the branch-and-bound method.

The branch-and-bound method begins with the basic principle that if one solves the integer or mixed integer program as a linear program and the solution is all integers then the optimum integer program is the same linear program optimum. The integer program feasible solution region is a subset of the linear program's feasible solution region [16]. This allows the use of some conditional based linear programming to be applied to account for the integer requirement. The branch-and-bound method begins by taking the equations in standard form, removing the integer requirement, and producing a linear program solution. It then establishes this output as an upper bound to the integer program since it is a subset of the linear program. If the solution is not an integer programming solution, meaning the solution contains integers for the variables required to be integers then the algorithm moves on to the next step. The algorithm takes an arbitrary integer variable and selects the two integers nearest to the optimum value found in the linear program. From there it branches out two linear programs based on either of the integer values. It will continue branching using the two surrounding integers to the optimal solution until the either the linear program solution is an integer only solution or all the integer variables have been branched off as integers. Work is saved by pruning branches and all their respective sub-branches when the solution to that branch is either infeasible or yields an output less than an already established possible integer solution [16]. Once all the branches have been pruned or solved with an integer solution the solution to the integer programming problem has been computed.

CHAPTER 3
CONDITIONAL VALUE AT RISK OPTIMIZATION

The general premise of optimization using conditional Value-at-Risk constraints is its focus on minimizing the risk of losses. This minimization is based a calculated conditional Value-at-Risk (CVaR) rather than a more well known Value-at-Risk (VaR). When VaR is used as the upper bound to losses there are a number of mathematical difficulties that arise such as lack of sub-additivity and convexity. Utilizing CVaR helps solve optimization problems by working with an upper bound that does not have these mathematical difficulties [14]. This allows for conditional CVaR to be used in mathematical programming to determine a more useful optimization. Because CVaR ≥ VaR this approach can be broadly accepted as a more conservative method that produces a much easier computationally optimization solution [14].

3.1 CVaR

One of the most well-known measures used in robust optimization under conditions of uncertainty is VaR. The VaR establishes an upper bound for a particular loss distribution [18]. A percentage is assigned, α, that is the chance expected losses would exceed a set maximum acceptable loss amount, ς, defined as

$$\psi(x,\zeta) = \mathrm{P}\{L(x,Y) \le \zeta\} \tag{3.1}$$

VaR is the most widely applied risk measure for stochastic optimizations mainly because it is conceptually easy to grasp and apply [18]. An attractive alternative to using VaR as a measure of loss in optimization problems is to use CVaR.

CVaR is derived by taking a weighted average between the value at risk and losses exceeding the value at risk. Mathematically CVaR is broken down to this equation

$$F_{\alpha}(x,\zeta) := \zeta + \frac{1}{(1-\alpha)} E\{\max\{L(x,Y)-\zeta,0\}\} \tag{3.2}$$

CVaR has the mathematical properties of being sub-additive and convex. These properties allow it be used in optimization using mathematical programming [16]. CVaR is more conservative than VaR and is a conditional expectation measure of potential loss.

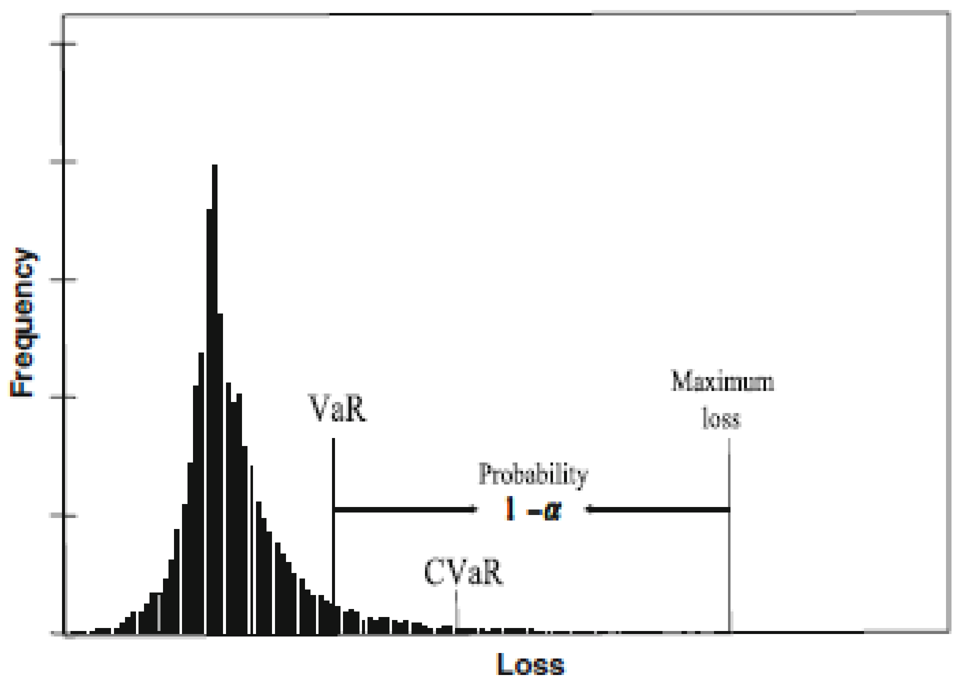

Figure 3-1. A graphic depiction of VaR and CVaR [18]

Optimization solutions based on CVaR will always satisfy the VaR upper bound. When the return-loss function is normal, the VaR and CVaR solutions are identical [16]. With these beneficial properties of CVaR in place a new approach to optimization is made possible.

3.2 The Approach

Let L(x,Y) be a loss function for a given personnel schedule where the decision variables for employees working are a vector x. The variable personnel demands are the y vector standing for the uncertainty of employee requirements. This Y vector can be assumed to have a probability density function p(y), but can be relaxed so as not to be strictly adhered to [14]. This probability density function can be approximated for p(y). This can be accomplished through either a derived analytical expression or a Monte Carlo simulation for drawing samples from p(y).

The optimization is now formulated to show a probability of not exceeding an upper bound, α, with the following equation

$$\psi(x,\alpha) = \int_{L(x,y)>\alpha} p(y)dy \tag{3.3}$$

This equation can be manipulated to take into account both VaR and CVaR in to one new function labeled $F_\alpha(x,\zeta)$ .This combined equation is defined as

$$F_\alpha(x,\zeta) = \zeta + \frac{1}{1-\alpha} \int_{L(x,y)>\zeta} (L(x,y)-\zeta)p(y)dy \tag{3.4}$$

Because the min CVaR is equal to min $F_\alpha(x,\zeta)$ we are able to optimize CVaR and find the VaR in one equation [16]. In addition to that quality the equation is convex and linear allowing linear programming to utilized.

3.3 Constraints

The basis of optimization through CVaR is that each organization has a specific amount of risk they are willing to take in a given situation. In the case of hospital pharmacy scheduling, the risk they can accept is somewhat high. That is if the manning happens to fall below demand they get behind and perhaps some problems arise but can be caught up on later shifts. In cases such as emergency room manning, the manning must meet demand or people can die. The ability to constrain the amount of risk associated with manning optimization is critical to producing a meaningful and effective optimization. Based on the CVaR constraints explained earlier, a mathematical programming model can be developed that will optimize a personnel schedule with adjustable risk associated with that optimization.

The mathematical programming model will have the standard objective function to minimize and any other relevant constraints that were explained in detail in a later section. The key element of the model to implement CVaR based optimization is adding the risk constraints. To approximate the random distribution of scheduling demand, scenarios are developed based on observations of the variations and employing Monte Carlo simulations to those observations. These individual scenarios and the sum of the losses in these scenarios are factored into the optimization in the following equation [16]

$$\overline{F}_{\alpha}(x,\zeta) = \zeta + \frac{1}{S(1-\alpha)} \sum_{s=1}^{S} \max\{L(x,Y) - \zeta, 0\} \tag{3.5}$$

where ς the VaR, C is the max tolerable loss, and α is the probability that the max tolerable loss is not exceeded. S is the number of scenarios used to represent the random distribution and t_s is an extra set of variables. There will be 2S + 1 constraints added to linearize the model. These constraints can be expressed as follows:

$$\begin{aligned} &t_s \geq L(x, y^s) - \zeta, \forall s = 1,2,\ldots,S \\ &t_s \geq 0, \forall s = 1,2,\ldots,S \end{aligned} \tag{3.6}$$

CHAPTER 4
HOSPITAL PHARMACY OVERVIEW

4.1 What A Hospital Pharmacy Does

To continue this thesis, some background information on the hospital pharmacy is in order. Pharmacies are often known from the commercial stores where prescription medicine is normally given out after a visit to the doctor. In reality, pharmacies are in many different locations and have different roles. Most pharmacies can be easily identified by a few symbols that are used to mark a pharmacy or something pharmacy related. There are a number of different kinds of pharmacies. As stated earlier, the most commonly known pharmacy is known as a “Retail Pharmacy.” Though this is a well known aspect of pharmacy, this thesis is focused on another kind of pharmacy known as a “Hospital Pharmacy.” Pharmacies within hospitals may have more complex medication management issues whereas pharmacists in community pharmacies often have more complex business and customer issues [12].

Another notable difference for the hospital pharmacies compared to other pharmacies is the medicine needed can be very critical and needed in a short period of time with a patient’s life hanging in the balance [4]. Due to the complexity, severity, and timeliness required out of the hospital pharmacy, they must maintain a 24 hours manned schedule that is able to meet the needs of the hospital [4].

4.2 Background Information on the Marietta Hospital

The Marietta hospital is run by a health care cooperation. The health care cooperation is a corporation that specializes in providing healthcare services. The corporation currently runs five hospitals in Georgia. In addition, they provide a variety of other healthcare services and employ over 11,000 people [10]. The Marietta hospital is located in Marietta, GA and serves most of central and northern Cobb County. It is not a trauma center and is not a teaching hospital [12]. The importance of these facts is that trauma centers have far more requirements. The Marietta hospital's pharmacy requirements would be very different and more complex if it was a trauma center. There is no educational portion of the hospital pharmacy which would also affect staffing requirements of the hospital pharmacy.

4.3 Pharmacy Organization

The Marietta hospital pharmacy has a fairly flat organizational structure. The pharmacist all work as equals in each shift regardless of seniority. There are pharmacist technicians that take direction from the pharmacists as needed and have their own requirements and scheduling. The pharmacists all report to the hospital pharmacy manager who typically works from 8 a.m. to 5 p.m. each weekday [1]. The hospital pharmacy manager reports to the director of hospital pharmacy. He is in charge of overall daily operations of the many aspects of a running a hospital pharmacy. Oversight of the director is provided by a Pharmacy and Therapeutics Committee selected by the health care cooperation [2]

CHAPTER 5
MODELS AND COMPUTATIONAL RESULTS

5.1 The Objective and Constraints

The hospital pharmacy has been operating for many years based on an established pharmacist rotation schedule. This schedule has the advantages of being simple to administer, of being established, and meets the requirements of both the hospital and the pharmacy staff. The current shifts used are seen in the following tables.

Name	Shifts	Weekday
Day Shift	0700-1600	7 Pharmacist
Evening Shift	1300-2400	6 Pharmacist
Night Shift	2300-0900	3 Pharmacist

Figure 5-1. The weekday day pharmacy schedule

Name	Shifts	Weekend
Day Shift	0700-1600	4 Pharmacist
Evening Shift	1300-2400	4 Pharmacist
Night Shift	2300-0900	3 Pharmacist

Figure 5-2. The weekend day pharmacy schedule

The shifts are eight hour shifts for the regular day, and two ten hours shifts for the evening and night shifts. That general shift schedule applies for both weekdays and weekends. The pharmacists have periods of overlap in their schedules to help high demand hours throughout the day. The weekend has fewer pharmacists because there are fewer hospital requirements on the weekend. The number of pharmacist working at a given hour is less of an exact science than a schedule that has been effective over the years. The objective of this thesis is to utilized CVaR constraints with an integer programming problem to create an optimum personnel schedule for the hospital.

Information on the costs that go along with each scheduled pharmacist has been provided by the hospital pharmacy management. Each pharmacist's pay is a little different based on a number of factors but they are all paid hourly not salary. The average hourly wage of $50 per hour is used for this thesis. It is a good average pay for the pharmacist based on an overall assessment of pharmacist pay. Certain hours of the 24 hour day have extra hourly pay associated with it as compensation for working outside of normal hours. This incentive pay structure is seen in the following table.

Hours	**Shift Incentive Pay**	
0700-1600	$0.00	per hour
1700-2200	$5.00	per hour
2300-0600	$8.00	per hour

Figure 5-3. The pharmacist's shift incentive pay table

These amounts are added to the hourly wage of the pharmacists to give an incentive for working a less desirable schedule. This pay scale is included in the objective formula for minimizing costs.

The required number of pharmacist per hour is not an established value. It has been informally established by the standard set schedule. Thanks to the management software used by the Marietta hospital pharmacy, the hospital data with pages of requests for pharmacy support broken down per hour was provided for this thesis. The raw number of requests per hour of the pharmacy can be found in Appendix A. The break down according to pharmacist professionals at the hospital is that a pharmacist can handle 27 pages of requests per hour. By taking multiple days of pharmacy requests data and determining an average, a pharmacy requirement can be established based on the hour of a given day. Based on the collected data from the hospital pharmacy and the consensus that a pharmacist typically handles 27 requests per hour, a requirement for pharmacists per hour was created. This requirement is seen in the following table.

Time Period	Weekdays Pharmacists	Weekend Pharmacists
0	3	2
1	2	2
2	2	2
3	2	2
4	2	1
5	2	1
6	2	2
7	2	2
8	5	4
9	6	4
10	7	6
11	8	7
12	8	7
13	7	6
14	7	6
15	7	6
16	8	7
17	7	5
18	7	4
19	5	3
20	4	3
21	4	3
22	3	3
23	3	3

Figure 5-4. A table of required Pharmacist for any given hour

As seen from the above table the required pharmacist broken into two schedules, weekday and weekend. This is based on the pharmacist staff requirements.

The pharmacy staff has a few requirements that must be factored into the optimization problem. The pharmacists require that they have a steady schedule over the week and a steady schedule over the weekend. This means that whatever optimized schedule is created must be the same for all the days of the week and a consistent schedule for the two days of the weekend. Another requirement of the pharmacists is that when they work a shift they must work either a full eight hour or ten hours shift. As seen from above, the pharmacists' requirements are broken down into a weekend day and weekend requirement to match the scheduling requirements. The eight hour and ten hour shift requirement is factored into the objective function.

5.2 The Model

The model was formulated based on the previous section's objective goal and constraints. The objective function is to minimize the cost of employing pharmacists while still meeting the required pharmacists needed by the hospital for any given hour in the day, 24 hours a day, and seven days a week. Microsoft Office Excel's solver was used as the tool to solve the optimization problem. Excel was chosen because it is used by the Marietta hospital staff, it is easy to use, and its uses the branch-and-bound as its solver algorithm.

The first step to applying integer programming on this problem is by putting the objective and constraints into standard form. The first equation of the standard form is the objective function. To develop the objective function the cost of starting an eight hour shift or a ten hour shift at any given hour must be calculated. The first step of this process is to establish the hourly wage of a pharmacist per hour of the day based on the incentive pay and the average base pay of $50 dollar an hour. The following table is a breakdown of the hourly wage of an average pharmacist working on a given hour.

Time Period	Hourly Wage
0000	$58.00
0100	$58.00
0200	$58.00
0300	$58.00
0400	$58.00
0500	$58.00
0600	$58.00
0700	$50.00
0800	$50.00
0900	$50.00
1000	$50.00
1100	$50.00
1200	$50.00
1300	$50.00
1400	$50.00
1500	$50.00
1600	$50.00
1700	$55.00
1800	$55.00
1900	$55.00
2000	$55.00
2100	$55.00
2200	$55.00
2300	$58.00

Figure 5-5. The table of hourly wages based on a given hour of the day

Once the hourly wage per hour has been calculated the cost of starting a pharmacist to work either an eight hour or ten hour shift on any given hour of the day must be calculated. These calculations are in the following table.

Time Period	8 Hour Shift	10 Hour Shift
0000	$456.00	$556.00
0100	$448.00	$548.00
0200	$440.00	$540.00

Figure 5-6. The cost of starting a pharmacist shift on any given hour of the day

Time Period	8 Hour Shift	10 Hour Shift
0300	$432.00	$532.00
0400	$424.00	$524.00
0500	$416.00	$516.00
0600	$408.00	$508.00
0700	$400.00	$500.00
0800	$400.00	$505.00
0900	$400.00	$510.00
1000	$405.00	$515.00
1100	$410.00	$520.00
1200	$415.00	$525.00
1300	$420.00	$530.00
1400	$425.00	$538.00
1500	$430.00	$546.00
1600	$438.00	$554.00
1700	$446.00	$562.00
1800	$449.00	$565.00
1900	$452.00	$568.00
2000	$455.00	$571.00
2100	$458.00	$574.00
2200	$461.00	$569.00
2300	$464.00	$564.00

Figure 5-6.Continued

With the cost of starting a pharmacist on a given hour on either of the shifts is calculated the objective function can be completed. The objective function is seen below.

$$Min\ Z = \sum_{i=0}^{1}\sum_{j=0}^{23} N_{ij} X_{ij} \tag{5.1}$$

Where N_{ij} is the number of pharmacist starting an eight hour shift i = 0 or a ten hour shift i = 10 on the j'th hour which is from 0 – 23. The number of pharmacist starting for a particular hour and shift is multiplied times the costs of an eight or ten hour shift, variable X_{ij}, with an eight hour shift i = 0 or a ten hour shift i = 1, on the j'th hour which is from 0 to 23.

The constraints are calculated based on the number pharmacists required per hour to meet the hospital requirements. These requirements are found in Table 3-4. These constraints are put into standard form as follows, adding up the eight hour shift pharmacists from the last seven hours and the ten hour shift pharmacists from the last nine hours plus the shifts starting during that hour to add up the total number of pharmacist working at that given hour. The total number must equal or be greater than the required number of pharmacists required. The constraint put into standard form is as follows

$$
\begin{aligned}
& s.t. \\
& \sum_{j=0-7}^{0} N_{0j} + \sum_{j=0-9}^{0} N_{1j} \geq R_0 \\
& \vdots \qquad \vdots \qquad \vdots \\
& \sum_{j=23-7}^{23} N_{0j} + \sum_{j=23-9}^{23} N_{1j} \geq R_{23}
\end{aligned}
\tag{5.2}
$$

Where N_{ij} is the number of pharmacist starting an eight hour shift i = 0 or a ten hour shift i = 10, on the j'th hour of the requirement, and k is either the weekday, I = 0, or weekend shifts, i = 1. The R_j is the from Table 3-4 which is the number pharmacist required for that hour of the day where j is the hour of the day, 0 through 23. There are 24 requirements based on the general form seen in figure 3-3. Each one corresponds to an hour of a 24 hour day.

Constraints must be put on the inputs themselves. The variables must be greater than zero and that they must be integers. This allows the overall standard form equation to be formed. The requirement that the input variables must be integers equates to a pharmacist must be a whole unit and cannot be negative which reflects reality. The ability to use a pharmacist in a fraction form is not in line with the pharmacist shift requirements. These two requirements put into standard form are seen in the following figure.

$$\begin{aligned}&\sum_{i=0}^{1}\sum_{j=0}^{23}N_{ij},\sum_{j=0}^{23}R_j : Integer\\&\sum_{i=0}^{1}\sum_{j=0}^{23}N_{ij},\sum_{j=0}^{23}R_j,\sum_{i=0}^{1}\sum_{j=0}^{23}X_{ij}\geq 0\end{aligned} \tag{5.3}$$

The combined integer programming optimization problem in standard form to be optimized is seen

$$\begin{aligned}&Min\ Z=\sum_{i=0}^{1}\sum_{j=0}^{23}N_{ij}X_{ij}\\&s.t.\\&\sum_{j=0-7}^{0}N_{0j}+\sum_{j=0-9}^{0}N_{1j}\geq R_0\\&\vdots\qquad\vdots\qquad\vdots\\&\sum_{j=23-7}^{23}N_{0j}+\sum_{j=23-9}^{23}N_{1j}\geq R_{23}\\&\sum_{i=0}^{1}\sum_{j=0}^{23}N_{ij},\sum_{j=0}^{23}R_j : Integer\\&\sum_{i=0}^{1}\sum_{j=0}^{23}N_{ij},\sum_{j=0}^{23}R_j,\sum_{i=0}^{1}\sum_{j=0}^{23}X_{ij}\geq 0\end{aligned} \tag{5.4}$$

The final constraints and variables to add to the standard form equation for this optimization problem are those relating to the CVaR constraints. There must be a constraint for loss tolerance shown here

$$\zeta+\frac{1}{S(1-\alpha)}\sum_{s=1}^{S}t_s\leq C \tag{5.5}$$

The loss function is defined in this optimization problem as

$$L(x,y^s)=\sum_{j=0}^{23}(R_j-\sum_{j=j-7}^{j}N_{0j}-\sum_{j=j-9}^{j}N_{1j}) \tag{5.6}$$

This loss function is converted into a constraint in standard form with the t_s variables introduced to take into account the random distribution of requirements observed for this optimization. The CVaR constraints are

$$\begin{aligned} & t_s \geq \sum_{j=0}^{23}(R_j - \sum_{j=j-7}^{j} N_{0j} - \sum_{j=j-9}^{j} N_{1j}) - \zeta, \forall s = 1,2,...,S \\ & t_s \geq 0, \forall s = 1,2,...,S \end{aligned} \tag{5.7}$$

The complete CVaR optimized integer problem to be solved with the included CVaR constraints is

$$\begin{aligned} & Min\ Z = \sum_{i=0}^{1}\sum_{j=0}^{23} N_{ij} X_{ij} \\ & s.t. \\ & \sum_{j=0-7}^{0} N_{0j} + \sum_{j=0-9}^{0} N_{1j} \geq R_0 \\ & \vdots \qquad \vdots \qquad \vdots \\ & \sum_{j=23-7}^{23} N_{0j} + \sum_{j=23-9}^{23} N_{1j} \geq R_{23} \\ & \zeta + \frac{1}{S(1-\alpha)} \sum_{s=1}^{S} t_s \leq C \\ & t_s \geq \sum_{j=0}^{23}(R_j - \sum_{j=j-7}^{j} N_{0j} - \sum_{j=j-9}^{j} N_{1j}) - \zeta, \forall s = 1,2,...,S \\ & t_s \geq 0, \forall s = 1,2,...,S \\ & \sum_{i=0}^{1}\sum_{j=0}^{23} N_{ij}, \sum_{j=0}^{23} R_j : Integer \\ & \sum_{i=0}^{1}\sum_{j=0}^{23} N_{ij}, \sum_{j=0}^{23} R_j, \sum_{i=0}^{1}\sum_{j=0}^{23} X_{ij} \geq 0 \end{aligned} \tag{5.8}$$

5.3 The Results of the Case Study

To first get a standard integer programming solution as a base for the implementation of CVaR constraints, the standard form without CVaR constraints based on the average demand per hour was implemented. The solver was used on the objective function with the goal of minimizing costs. The integer option was selected for all inputs. The solver was run for an optimized weekday schedule and weekend schedule each. The following are the results of both optimization calculations.

Weekday Schedule:

0200 - 1 person starts a 8 hr shift and 1 person starts a 10 hour shift
0400 - 1 person starts a 10 hr shift
0800 - 3 people start an 10 hr shift
0900 - 1 person starts an 8 hr shift
1000 - 1 person starts a 10 hr shift
1100 - 2 people start an 8 hr shift
1500 - 1 person starts an 8 hr shift
1800 - 2 people start an 8 hr shift and one person starts a 10 hr shift

Figure 5-5. The schedule results of the minimization solver for the weekday schedule.

Weekend Schedule:

0300 - 2 people start a 8 hr shift
0600 - 1 person starts a 10 hr shift
0700 - 1 person starts an 8 hr shift
0900 - 2 people start an 8 hr shift
1000 - 2 people start an 8 hr shift
1100 - 1 person starts an 8 hr shift
1600 - 2 people start a 10 hr shift
1800 - 1 person starts an 8 hr shift

Figure 5-6. The schedule results of the minimization solver for the weekend schedule.

The raw results and spreadsheet answers generated by the solver as well as the set up of the standard form equations in excel are all found in Appendix B. The number of input factors for the equations is 48 inputs. The file used for the optimization is available for further analysis.

The results of the integer programming were encouraging. The cost of running the pharmacy for a weekday with the current schedule is $7,847.00 and the cost of running the pharmacy during a weekday with the optimized schedule is $6,647.00. The cost of running the pharmacy on a weekend day with the current schedule is $5,412.00 and the cost of running the pharmacy on a weekend day with the optimized schedule is $5,349.00. The cost saving over a year equate to $312,000 for the weekday shift and $6,552 for the weekend shift. The overall yearly saving based on this optimization schedule is $318,552.00. Just as important as costs is that the number of pharmacists is better distributed for the demands of the hospital.

With this baseline established, the CVaR constraints were added to the standard form of the personnel scheduling problem. A Monte Carlo method was used to pick from the available data on the pharmacy orders per hour to create ten scenarios used for the implementation of the CVaR constraints. Working with the pharmacist team the value for α was set at 0.9. The loss function was implemented for each of the ten scenarios with the t_s variables added as constraints. The resulting calculations produced the following results.

Weekday Schedule:

0700 - 5 people start a 8 hr shift
0900 - 2 people start a 8 hr shift
1100 - 2 people start a 8 hr shift
1200 - 1 person starts a 8 hr shift
1500 - 4 people start a 8 hr shift
2300 - 3 people start a 8 hr shift

Figure 5-7. The schedule results of the solver with CVaR Constraints for weekday schedule.

Weekend Schedule:

0300 - 1 person starts a 8 hr shift
0600 - 1 person starts a 10 hr shift
0700 - 1 person starts a 10 hr shift

Figure 5-8. The schedule results of the solver with CVaR Constraints for weekday schedule.

0800 - 1 person starts a 8 hr shift
0900 - 2 people start a 8 hr shift
1000 - 1 person starts a 8 hr shift
1100 - 1 person starts a 8 hr shift
1500 - 1 person starts a 8 hr shift
1600 - 1 person starts a 10 hr shift
1700 - 1 person starts a 10 hr shift
2300 - 1 person starts a 8 hr shift

Figure 5-8. Continued

With the CVaR constraints factored in, the costs of running the pharmacy on the weekdays is $7144.00 per day and for the weekends is $5464.00 per day. This shows substantial savings in costs for the weekdays compared to $7,847.00 for the current schedule. The weekends costs actually are higher than the current schedule. This is because the current weekend schedule causes the pharmacy to get undermanned and behind on orders during some high demand weekends. The costs of running the pharmacy on a weekend day is $5,412.00 unoptimized versus $5,464.00 based on the CVaR constrained optimized schedule. This slight increase in cost allows much better pharmacy coverage of hospital demands on the weekends. This difference in costs of the new CVaR optimized schedule compared to the current schedule is $182,780.00 for weekdays and -$5,408.00 for the weekends over the span of a year. Overall this equates to $177,372 in saving per year while better covering the needs of the hospital and giving a statistically significant solution. This amount is less than the $318,552.00 saving based on integer programming without the CVaR constraints but ensures that the pharmacy will have coverage during higher demand times while still saving money based on the current schedule.

CHAPTER 6
CONCLUSION

The results of the optimization of the personnel scheduling problem utilizing CVaR constraints show an effective application of this new theory. This thesis proves that CVaR constraints can easily be added to personnel scheduling problems. The possible substantial amount of yearly savings is a great example of how money can be saved utilizing mathematical programming to find a more optimal way of doing business while at the same time allowing organizations to determine acceptable levels of risk to achieve those cost savings. Translating all the constraints and inputs into standard form and establishing the integer programming parameters took time and effort but was straight forward and easy to alter once initially setup. The integer program with CVaR constraints setup made it very easy to import in more data as demand changed over time. This showed how this model is easy to alter for future needs. With a few easy changes the hospital pharmacy can alter how much risk they are willing to take with their pharmacy schedule to better serve their needs. In conclusion, integer programming with CVaR constraints was successfully applied to a real world optimization problem in this thesis proving that CVaR is viable and useful method to producing personnel timetable assignments.

CHAPTER 7
RECOMMENDATIONS

The results of the optimization problem are valid and would enhance pharmacist coverage of needed hospital requirements while saving money. Though the results appear great in this thesis, the reality is that the resulting schedule is more complicated than the previous schedule. There is a great deal of pharmacists coming and going as the shifts are very spread out. This would be a major shift in the way business is done by the pharmacy. Upon showing the results of this optimization problem to the management of the Marietta pharmacy there was great reluctance to implement such a scheduling change. The results speaks for themselves, the schedule easily shows how many pharmacist end up working at any given hour giving the manager confidence that the schedule does indeed cover the hospital needs. The real challenge with this optimization is people. Change can be difficult and it would require significant management push to implement this for the day shift pharmacists in particular who are typically senior and prefer their current schedule. If management is willing to make the personnel push and work through the employees reluctance then I would recommend implementing this new schedule. As stated previously, this schedule would save money, handle the hospital needs more effectively, and have the risk of being short staffed factored into the scheduling decisions.

APPENDIX A
RAW DATA

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
65	69	79	55	78	73	69
50	49	66	59	70	56	55
34	35	31	62	73	76	37
44	44	39	48	52	63	53
25	44	58	75	42	46	33
44	55	48	43	63	41	34
57	57	55	37	63	66	46
50	62	53	59	76	58	55
82	142	114	132	110	147	142
111	148	162	182	173	149	118
140	176	153	193	204	198	215
164	228	203	196	202	244	249
217	199	201	185	198	222	237
193	157	208	202	152	186	193
195	204	185	178	178	177	177
139	194	170	217	179	227	199
126	196	167	201	258	202	220
118	191	165	192	174	195	128
85	187	167	190	145	172	112
80	116	138	117	113	147	89
64	106	97	124	113	112	80
77	98	98	93	139	148	65
63	61	92	73	120	131	98
57	87	69	78	110	83	89

Table A-1. One week of raw pharmacy requests broken down by hour

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
78	69	56	55	71	74	69
70	49	23	59	90	71	55
73	35	38	62	23	37	37
52	44	41	48	18	48	53
42	44	48	75	64	48	33
63	55	39	43	59	36	34
63	57	38	37	58	46	46
76	62	64	59	91	81	55
110	142	132	132	134	150	142
173	148	152	182	190	180	118
204	176	176	193	212	195	215
202	228	242	196	232	217	249
198	199	227	185	222	199	237
152	157	197	202	210	207	193
178	204	183	178	172	175	177
179	194	178	217	199	232	199
258	196	247	201	192	220	220
174	191	186	192	183	188	128
145	187	169	190	168	181	112
113	116	118	117	159	130	89
113	106	83	124	129	108	80
139	98	85	93	120	112	65
120	61	78	73	101	92	98
110	87	76	78	85	87	89

Table A-2. Raw pharmacy order requests for a second week

Time Period	Weekdays	Weekend
0	67.9	70.25
1	59.2	57.5
2	47.2	45.25
3	44.5	50.5
4	54.4	33.25
5	48.2	43.75
6	51.4	53
7	66.5	59
8	133.5	119
9	166.6	130
10	187.6	193.5
11	218.8	216
12	203.7	222.25
13	187.8	182.75
14	183.4	181.75
15	200.7	179
16	208	206
17	185.7	137
18	175.6	113.5
19	127.1	92.75
20	110.2	84.25
21	108.4	86.5
22	88.2	94.75
23	84	86.25

Table A-3. The average pages of requested orders broken down into weekdays and weekends.

Time Period	Sunday	Sunday	Sunday	Sunday	Avg	VAR
0	65	43	48	55	52.75	90.91667
1	50	44	53	76	55.75	196.25
2	34	40	58	41	43.25	106.25
3	44	44	48	30	41.5	62.33333
4	25	48	38	35	36.5	89.66667
5	44	54	40	25	40.75	144.9167
6	57	56	38	30	45.25	179.5833
7	50	66	63	46	56.25	94.91667
8	82	116	113	132	110.75	436.9167
9	111	151	117	136	128.75	333.5833
10	140	185	185	168	169.5	451
11	164	253	204	210	207.75	1326.917
12	217	249	203	224	223.25	370.9167
13	193	121	210	195	179.75	1591.583
14	195	128	183	176	170.5	864.3333
15	139	124	197	198	164.5	1489.667
16	126	87	181	219	153.25	3408.25
17	118	117	160	89	121	856.6667
18	85	83	93	82	85.75	24.91667
19	80	50	60	56	61.5	169
20	64	93	70	64	72.75	190.25
21	77	62	78	57	68.5	112.3333
22	63	55	53	74	61.25	90.91667
23	57	66	60	74	64.25	56.25

Table A-4. Here is the variance of for samples of a Sunday

APPENDIX B
THE DATA WITH THE OPTIMIZATION PROBLEM

Weekend		0	1	2	3	4	5	6	7	8	9	10	11
	8 Hr	0	0	0	2	0	0	0	1	0	2	2	1
	10 Hr	0	0	0	0	0	0	1	0	0	0	0	0

	12	13	14	15	16	17	18	19	20	21	22	23
8 Hr	0	0	0	0	0	0	1	0	0	0	0	0
10 Hr	0	0	0	0	2	0	0	0	0	0	0	0

Table B-1. The input setup for the weekend solution with the results of the optimizer as the answers.

Weekday	8 Hr	0	1	2	3	4	5	6	7	8	9	10	11
		0	0	1	0	0	0	0	0	0	1	0	2
	10 Hr	0	1	2	3	4	5	6	7	8	9	10	11
		0	0	1	0	1	0	0	0	3	0	1	0

12	13	14	15	16	17	18	19	20	21	22	23
0	0	0	1	0	0	2	0	0	0	0	0
12	13	14	15	16	17	18	19	20	21	22	23
0	0	0	0	0	0	1	0	0	0	0	0

Table B-2. The input setup for the weekday solution with the results of the optimizer as the answers.

0 Shift	0																	0	1	0	0	0	0	0	3	>=	3
	0															0	2	0	0	0	0	0	0	0			
1 Shift	0	0																	1	0	0	0	0	0	3	>=	3
	0	0															2	0	0	0	0	0	0	0			
2 Shift	0	0	0																	0	0	0	0	0	2	>=	2
	0	0	0															2	0	0	0	0	0	0			
3 Shift	0	0	0	2																	0	0	0	0	2	>=	2
	0	0	0	0															0	0	0	0	0	0			
4 Shift	0	0	0	2	0																	0	0	0	2	>=	2
	0	0	0	0	0															0	0	0	0	0			
5 Shift	0	0	0	2	0	0																	0	0	2	>=	2
	0	0	0	0	0	0															0	0	0	0			
6 Shift	0	0	0	2	0	0	0																	0	3	>=	3
	0	0	0	0	0	0	1															0	0	0			
7 Shift	0	0	0	2	0	0	0	1																	4	>=	3
	0	0	0	0	0	0	1	0															0	0			
8 Shift		0	0	2	0	0	0	1	0																4	>=	4
	0	0	0	0	0	0	1	0	0															0			
9 Shift			0	2	0	0	0	1	0	2															6	>=	5
	0	0	0	0	0	0	1	0	0	0																	
10 Shift				2	0	0	0	1	0	2	2														8	>=	7
		0	0	0	0	0	1	0	0	0	0																
11 Shift					0	0	0	1	0	2	2	1													7	>=	7
			0	0	0	0	1	0	0	0	0	0															
12 Shift						0	0	1	0	2	2	1	0												7	>=	7
				0	0	0	1	0	0	0	0	0	0														
13 Shift							0	1	0	2	2	1	0	0											7	>=	7
					0	0	1	0	0	0	0	0	0	0													
14 Shift								1	0	2	2	1	0	0	0										7	>=	7

Table B-3. Weekend constraints

	0	1	0	0	0	0	0	0	0	0												
15 Shift				0	2	2	1	0	0	0	0									6	>=	6
		1	0	0	0	0	0	0	0	0	0											
16 Shift					2	2	1	0	0	0	0	0								7	>=	7
			0	0	0	0	0	0	0	0	0	2										
17 Shift						2	1	0	0	0	0	0	0							5	>=	5
				0	0	0	0	0	0	0	0	2	0									
18 Shift							1	0	0	0	0	0	0	1						4	>=	4
					0	0	0	0	0	0	0	2	0	0								
19 Shift								0	0	0	0	0	0	1	0					3	>=	3
						0	0	0	0	0	0	2	0	0	0							
20 Shift									0	0	0	0	0	1	0	0				3	>=	3
							0	0	0	0	0	2	0	0	0	0						
21 Shift										0	0	0	0	1	0	0	0			3	>=	3
								0	0	0	0	2	0	0	0	0	0					
22 Shift											0	0	0	1	0	0	0	0		3	>=	3
									0	0	0	2	0	0	0	0	0	0				
23 Shift												0	0	1	0	0	0	0	0	3	>=	3
										0	0	2	0	0	0	0	0	0	0			

Table B-3. Continued

0 Shift	0																	0	2	0	0	0	0	0	3	>=	3
	0															0	0	0	1	0	0	0	0	0			
1 Shift	0	0																	2	0	0	0	0	0	3	>=	3
	0	0															0	0	1	0	0	0	0	0			
2 Shift	0	0	1																	0	0	0	0	0	3	>=	3
	0	0	1															0	1	0	0	0	0	0			
3 Shift	0	0	1	0																	0	0	0	0	3	>=	3
	0	0	1	0															1	0	0	0	0	0			
4 Shift	0	0	1	0	0																	0	0	0	3	>=	3
	0	0	1	0	1															0	0	0	0	0			
5 Shift	0	0	1	0	0	0																	0	0	3	>=	3
	0	0	1	0	1	0															0	0	0	0			
6 Shift	0	0	1	0	0	0	0																	0	3	>=	3
	0	0	1	0	1	0	0															0	0	0			
7 Shift	0	0	1	0	0	0	0	0																	3	>=	3
	0	0	1	0	1	0	0	0															0	0			
8 Shift		0	1	0	0	0	0	0	0																6	>=	5
	0	0	1	0	1	0	0	0	3															0			
9 Shift			1	0	0	0	0	0	0	1															7	>=	7
	0	0	1	0	1	0	0	0	3	0																	
10 Shift				0	0	0	0	0	0	1	0														7	>=	7
		0	1	0	1	0	0	0	3	0	1																
11 Shift					0	0	0	0	0	1	0	2													9	>=	9
			1	0	1	0	0	0	3	0	1	0															
12 Shift						0	0	0	0	1	0	2	0												8	>=	8
				0	1	0	0	0	3	0	1	0	0														
13 Shift							0	0	0	1	0	2	0	0											8	>=	7
					1	0	0	0	3	0	1	0	0	0													
14 Shift								0	0	1	0	2	0	0	0										7	>=	7

Table B-4. Weekday constraints

	0	0	0	3	0	1	0	0	0	0												
15 Shift				0	1	0	2	0	0	0	1									8	>=	8
		0	0	3	0	1	0	0	0	0	0											
16 Shift					1	0	2	0	0	0	1	0								8	>=	8
			0	3	0	1	0	0	0	0	0	0										
17 Shift						0	2	0	0	0	1	0	0							7	>=	7
				3	0	1	0	0	0	0	0	0	0									
18 Shift							2	0	0	0	1	0	0	2						7	>=	7
					0	1	0	0	0	0	0	0	0	1								
19 Shift								0	0	0	1	0	0	2	0					5	>=	5
						1	0	0	0	0	0	0	0	1	0							
20 Shift									0	0	1	0	0	2	0	0				4	>=	4
							0	0	0	0	0	0	0	1	0	0						
21 Shift										0	1	0	0	2	0	0	0			4	>=	4
								0	0	0	0	0	0	1	0	0	0					
22 Shift											1	0	0	2	0	0	0	0		4	>=	4
									0	0	0	0	0	1	0	0	0	0				
23 Shift												0	0	2	0	0	0	0	0	3	>=	3
										0	0	0	0	1	0	0	0	0	0			

Table B-4. Continued

S =	10	Overall CVaR Constraint	C
ς =	-19.94218809		
α =	0.9	18.71247 <=	20

Loss minus ς	1	2	3	4	5	6	7	8	9	10
	-396.387	-306.387	-191.3874856	-0.38749	-318.3874856	-113.387	-199.3874856	-244.387	-110.3874856	-157.3874856
	<	<	<=	<=	<=	<	<=	<=	<=	<=
	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10
ts	22.55508	1.739802	1.409276532	4.393267	1.757499393	1.243737	1.42542377	1.54461	1.237800083	1.348162704
ts Max	22.55508	1.739802	1.409276532	4.393267	1.757499393	1.243737	1.42542377	1.54461	1.237800083	1.348162704

Table B-5. Weekday CVaR constraints

S =	10	Overall CVaR Constraint	C
ς =	-7.483129122		
α =	0.9	19.73049 <=	20

Loss minus ς	1	2	3	4	5	6	7	8	9	10
	-275.405	-16.4047	-265.4046653	-193.405	-35.40466533	-87.4047	-54.40466533	-148.405	-173.4046653	-94.40466533
	<=	<=	<=	<=	<=	<=	<=	<=	<=	<=
	t1	t2	t3	t4	t5	t6	t7	t8	t9	t10
ts	0.516578	1.781172	0.519341936	0.522869	0.523697179	0.517978	1.208647597	20.50992	0.53083104	0.58258303
ts Max	0.516578	1.781172	0.519341936	0.522869	0.523697179	0.517978	1.208647597	20.50992	0.53083104	0.58258303

Table B-6. Weekend CVaR constraints

LIST OF REFERENCES

[1] "Computer Entry for Prescribers' Orders", *Pharmacy Department Policy and Procedure*, Wellstar Health System, DPP #II-01, August 2008

[2] "Pharmacy Department Overview", *Pharmacy Department Policy and Procedure*, Wellstar Health System, Aug 2008, DPP# I-01

[3] D. C. Montgomery, *Design and Analysis of Experiments*, 7th edition, John Wiley & Sons, Inc., New Jersey, USA, 2009

[4] D. J. Patil, *Hospital And Clinical Pharmacy*, 2nd edition, Nirali Prakashan Inc, Gultekdi, India, 2008

[5] D.P. Ahlfeld, P.M. Barlow, and A. E. Mulligan, *GWM--a ground-water management process for the U.S. Geological Survey modular ground-water model {MODFLOW-2000}, U.S. Geological Survey Open-File Report* 2005-1072, 124p.

[6] F. Glover and C. McMillan, "The General Employee Scheduling Problem: An Integration of MS and AI," *Comput. & Ops. Res.*, Vol. 13, No.5, pp. 563-573, 1986

[7] J.K. Karlof, *Integer Programming Theory and Practice*, Taylor & Francis Group, Boca Raton, FL, USA, 2006.

[8] M. S. Bazaraa, J.J. Jarvis, and H. D. Sherali, *Linear Programming and Network Flows*, 4th edition, John Wiley & Sons, Inc, New Jersey, USA, 2010

[9] M.I. Andreica, R. Andreica, and A. Andreica, "Minimum Dissatisfaction Personnel Scheduling," *Proceedings of the 32nd Congress of the American Romanian Academy of Arts and Sciences (ARA Congress)*, pp. 459-463, Boston, USA, 2008

[10] Office of the Professions, www.op.nysed.gov/prof/pharm/, taken March 2010

[11] P.D. Causmaecker, P. Demeester, G. V. Berghe, and B. Verbeke, "Analysis of real-world personnel scheduling problems", *Information Technology*, Gent, Belgium, 2008

[12] R. A. Wurbs, *Computer Models for Water-Resources Planning and Management*, IWR Report 94-NDS-7, Texas A&M, USA, July 1994

[13] R. Adams, "Optimiser, a linear-programming spreadsheet", *InfoWorld*, vol. 4, no. 20, pp. 34-35, May 1982

[14] R.T. Rockafellar. and S. Uryasev, "Optimization of Conditional Value-At-Risk," *The Journal of Risk*, Vol. 2, No. 3, 2000, 21-41

[15] S. Agne, "The Impact of Pesticide Taxation on Pesticide Use and Income in Costa Rica's Coffee Production", Publication Series Special Issue No. 2, Pesticide Policy Project, IICA Biblioteca Venezuela, March 2000

[16] S. Uryasev and P. M. Pardalos, *Stochastic Optimization: Algorithms and Applications,* Kluwer Academic Publishers, Dordrecht, The Netherlands, 2001

[17] S. Uryasev, "Conditional Value-at-Risk: Optimization Algorithms and Applications", *Financial Engineering News*, vol. 14, February 2000.

[18] V. L. Boginski, C. W. Commander, and T. Turko, "Polynomial-time Identification of Robust Network Flows Under Uncertain Arc Failures," Optimization Letters, Vol. 3, pp. 461-473, 2009

[19] W. L. Winston and M. Venkataramanan, *Operations Research Volume One Introduction To Mathematical Programming,* 4th edition, Thompson Learning, California, USA, 2003

BIOGRAPHICAL SKETCH

John Robert Stripling IV (Robert) was born in New Orleans, LA in 1979 to Catherine Ann Stripling and Dr. John Robert Stripling III. Robert grew up with a half-sister, Monica Davison Hertzbach, 14 years his elder and a younger brother, Nathan George Stripling, 18 months his junior. In 1994 Robert and his brother moved up to Atlanta, GA for the remainder of their pre-college education. Robert and his brother attended one of the top college preparation schools in the Atlanta area, Greater Atlanta Christian School. Robert became one of the top students in high school while also lettering in football, tennis, and wrestling. He became student body president his senior year and was set to begin college after high school graduation in 1998. Once acceptance letters were received, Robert selected Vanderbilt University in Nashville, TN as the school of choice for his higher education.

In August 1998 Robert began his undergraduate degree of undecided. In the second semester of his sophomore year he changed his degree to computer engineering. During these initial years, Robert was active outside of the class room. He became a cheerleader for the university. In the second semester of his sophomore year, Robert joined the Sigma Alpha Epsilon fraternity, the largest fraternity in the country at the time

Robert graduated in May 2002 with a Bachelor of Engineering and entered a country deep in recession from the “dot com” bubble bursting. He found work at Vanderbilt’s Owen Business school library doing information technology support and web design. After six months of looking for an engineering job, Robert decided to join the Air Force. By July, 2003 he had started basic officer training. After basic officer training, he was assigned to Lackland AFB in San Antonio, TX. Robert's first job was developing network operations capabilities working under the Air Force Information Warfare Center. After three years, his next assignment was to Eglin AFB in Fort Walton Beach, FL. There he managed two offices of government electronic warfare engineers supporting the combat Air Force. It is at Eglin AFB that Robert began his master's degree at the University of Florida pursuing a degree in industrial and systems engineering which began June 2008.