5-2010

Heuristic scheduling for clinical physicians.

Dustin Banet
University of Louisville

HEURISTIC SCHEDULING FOR CLINICAL PHYSICIANS

By

Dustin Banet
B.S., University of Louisville, 2009

A Thesis
Submitted to the Faculty of the
University of Louisville
J.B. Speed School of Engineering
in Partial Fulfillment of the Requirements
for the Professional Degree

MASTER OF ENGINEERING

Department of Industrial Engineering

May 2010

HEURISTIC SCHEDULING FOR CLINICAL PHYSICIANS

Submitted by: ___________________________________
Dustin Banet

A Thesis Approved On

____
(Date)

by the Following Reading and Examination Committee:

____
Dr. Gail DePuy, Thesis Director

____
Dr. William E. Biles

____
Dr. Tim Hardin

ACKNOWLEDGEMENTS

The results of this study must be credited to the contributions of many individuals. Thank you Brian Knop, Rita Plomin, and Marjorie Baird from the University of Louisville Outpatient Center for providing the me the opportunity to work on this project and for offering your assistance and expertise. Thank you Dr. Sunderesh Heragu and Dr. Gerald Evans for your voluntary support. And finally, a very special thanks to my advisor Dr. Gail DePuy for her constructive guidance and mentoring.

ABSTRACT

Personnel scheduling is a problem faced by many organizations in the healthcare industry, particularly in rapidly developing outpatient centers. The task of creating a schedule that adequately covers patient demand while satisfying the preferences of employees, observing work regulations, and ensuring a fair distribution of work is highly complex. Even though this highly complex task directly affects measures such as patient waiting time and employee satisfaction, many organizations still resort to the traditional and cumbersome manual solution methods. A large segment of prior research on personnel scheduling in healthcare focuses on nurse rostering and the development of automated tools to aid in scheduling. The drawbacks to these methods include the lack of generality and the need for specialized software packages and training. The aim of this study is the development of an effective, low cost, and uncomplicated heuristic tool to aid schedulers in outpatient centers. Solution methodologies used by previous researchers in problems such as nurse rostering and aircrew rostering are adapted to the particular problem of physician scheduling in mixed specialty outpatient clinics. The developed heuristic tool obtains an initial feasible solution using a greedy algorithm and then uses the simulated annealing metaheuristic to improve the solution, which is a measure of physician satisfaction. The heuristic tool developed in this study was tested using eight randomly generated data sets to

model 45 unique cases. The heuristic found the optimal solution in 19 of the 45 tested cases. The average difference from the optimal physician satisfaction rating in the other 26 cases was 0.35%.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

I. INTRODUCTION

In recent years, increased focus on preventative healthcare services combined with shorter lengths of patient stays has fostered the growth of outpatient clinics. According to a study by the Center for Disease Control and Prevention, outpatient surgery visits to freestanding centers increased threefold in the ten year period from 1996 to 2006 (Cullen *et al.*, 2009). Similar to other entities within the medical field, a critical factor in the operation and success of outpatient centers is effective scheduling of physicians, equipment, rooms, and patients. Physician scheduling in particular can present a complex problem for the schedulers of outpatient centers due to conflicts of interest among physicians and the organization. As schedulers endeavor to satisfy numerous physicians with competing preferences for work schedules, they must also consider the problem of balancing the workload throughout the planning period. In the case of an outpatient center, workload can be defined as the total number of patients to be seen by all physicians assigned to a given period of time. Maintaining a balanced patient load is necessary to sustain adequate staff utilization rates and low clinic overtime. Overtime arises when the scheduled demand exceeds clinic capacity, which commonly occurs due to the overlapping preferences for timeslots amongst physicians. A balanced patient load may also ensure that employed staff members dealing directly with all patients have a fair and equal balance of duties. Clearly, the task of scheduling physicians deserves much attention since the adoption of an inferior work schedule can result in poor efficiency, dissatisfied staff, and wasted expenditures on overtime.

Current scheduling methods contrast widely due to the functional differences among outpatient centers. Clinics may employ one or many practitioners and can be general practice or specialized. This study concentrates on a mixed specialty, multi-physician clinic. It is not uncommon for schedulers of such organizations to be equipped with nothing more than the basic and standard applications available on most modern day computers such as Microsoft Office applications. The task of developing an adequate work schedule that balances workload and is satisfying to physicians becomes progressively difficult as the number of physicians considered increases. Therefore, it is sensible to consider automated tools that would aid the scheduler in the physician scheduling problem.

Several considerations will need to be accounted for during the construction of this automated tool. First, every clinic is constrained by the total amount of physical space available. Physicians obviously cannot be scheduled in a manner so that the number of required examination rooms exceeds the capacity of the facility. Fortunately, satisfying the objective of work load balancing also inadvertently reduces the likelihood of facility overload by preventing the assignment of a large number of physicians to any specific period. Second, circumstances frequently exist in which physicians must be assigned or must not be assigned to a specific time slot. Therefore, the tool must grant the user the ability to fix or prevent specific assignments. Lastly, although schedule selection can be formulated as a quantitative procedure, subjective considerations must also be accounted for. For example, certain scheduling problems will have multiple optimal solutions. In these cases, the scheduler may prefer to deliberate amongst the alternative solutions. For this reason, the tool should be capable of returning an assortment of solutions through multiple iterations. Although adequate medical scheduling software is available commercially, clinics are often unable to justify the associated costs and instead resort to the development of effective, low cost automated tools.

Since many outpatient centers do not own the comprehensive programs for building and solving large-scale linear programs optimally and quickly such as LINGO or CPLEX, this research focuses on developing a solution heuristic capable of being programmed and solved in a more ubiquitous and obtainable application. Considering that the specific outpatient center at the focus of this study currently uses Microsoft Excel for its scheduling procedures, Visual Basic for Applications (VBA) was selected as the programming language for the solution heuristic.

The physician scheduling problem is described in greater detail in Chapter II, followed by a review of all relevant literature on this topic in Chapter III. Chapter IV explores the solution methodologies and Chapter V presents the results. Finally, the conclusions and recommendations for future research are provided in Chapter VII.

II. PROBLEM DESCRIPTION

A detailed description of the physician scheduling problem is provided in this section. Definitions for the terminology used throughout this paper will be presented first and followed by a thorough review and mathematical description of the problem.

The scheduling of physicians within an outpatient center can become a cumbersome task due to the wide range of specialties and variability in service times among physicians. The outpatient center studied for this project assigns physicians to morning and afternoon *slots*, which are separated by a mid-day, one hour lunch break. The schedule of a standard five day work week is made up of ten total *slots*, each being four hours in duration. The clinic personnel in charge of assigning physicians to *slots* must take several considerations into account. *Physician satisfaction* is defined as a measure of the willingness of a physician to work during any particular *slot*. One main objective for the scheduler is to schedule physicians so that the aggregated measure of *total physician satisfaction* is maximized. That is, when faced with competing slot preferences amongst physicians, the scheduler aims to make assignments that maximize the all-encompassing, *total physician satisfaction*.

In the case of an outpatient center with diverse specialties and physician service times, the number of patients scheduled per slot varies amongst physicians. For the clinic studied in this research, the number of patients that a physician will schedule during his or her assigned slot is dependent upon the physician's average service time. In essence, each physician determines his or her own workload. A competing objective for the scheduler is to keep *patient load,* the number of patients scheduled, reasonably level for all *slots*. The staff workload during any particular timeslot is directly related to the total patient load during that period, which is determined by the scheduling of physicians. Workload balancing not only contributes to improve process flow and performance but also benefits the outpatient center's staff and medical assistants in terms of a fair allotment of work. By balancing the total number of scheduled patients among the slots, a reduction in the variability of system parameters is attained.

In addition to considering the objectives of balancing *patient load* and maximizing *total physician satisfaction*, the scheduler must also take into account the space capacity of the facility. Physicians requesting a slot assignment will specify the number of examination rooms they will need. The scheduler must account for every physician's *room load* to ensure that the *room capacity*, the total number of available exam rooms in the department or facility, is not exceeded in any *slot*.

In regards to the assignment relationship between physicians and slots, individual physicians may request any number of slots. When requesting a slot, the physician must provide the scheduler with a completed informational survey such as the example shown in Figure 1 below. The completed survey provides the scheduler with all of the information necessary to formulate a suitable assignment. Some physicians may have absolutely no flexibility and only be capable of working during a specific time period or *slot*. In these cases, the scheduler may choose to grant a physician a *fixed assignment(s)*, which is predetermined and permanent.

Name ____________________

Specialty ____________________

Number of Slots Requested ____________________

Number of Patients to be Scheduled per Slot ____________________

Number of Examination Rooms Needed per Slot ____________________

Please number the time slots below in order of preference,
(1) being the most preferred and (10) being the least preferred.

	Monday	Tuesday	Wednesday	Thursday	Friday
AM					
PM					

FIGURE 1 – Scheduling Survey for Physicians

The scheduler's overall goal is to assign each physician to the number of slots he or she requests with the objectives of maximizing *total physician satisfaction* and balancing *patient load* among *slots* while satisfying the constraints in regard to space capacity. This dual objective physician-slot assignment problem is described mathematically below.

Parameters:

S_{ij} = physician i's scaled satisfaction rating for slot j

SR_i = slots requested by physician i

RR_i = examination rooms required by physician i

PL_i = patient load, or average number of patients seen per slot by physician i

RC = room capacity

I = ideal patient load

Decision Variables:

$$X_{ij} = \begin{cases} 1, & physician\ i\ is\ assigned\ to\ slot\ j \\ 0, & otherwise \end{cases}$$

$DP_j =$ difference, if positive, of actual patient load for slot j and ideal patient load

$DN_j =$ difference, if negative, of actual patient load for slot j and ideal patient load

Dual Objective Function:

Maximize Physician Satisfaction
and Balance Patient Load Minimize $\sum_i \sum_j \left(\alpha\left(S_{ij} * X_{ij}\right) + \beta\left(DP_j + DN_j\right)\right)$ (1)

Constraints:

All Physicians assigned to Requested Slots $\sum_j X_{ij} = SR_i$ $\forall i$ (2)

Space Capacity $\sum_i X_{ij} * RR_i \leq RC$ $\forall j$ (3)

Define Ideal Patient Load $I = \sum_i \sum_j \frac{PL_i * X_{ij}}{J}$ (4)

Define Difference in Actual
Patient Load and Ideal Load $DP_j - DN_j = \sum_i \left(PL_i * X_{ij} - I\right)$ $\forall j$ (5)

Binary Variables $X_{ij} \in \{0,1\}$ $\forall i, j$ (6)

Non-negativity $DP_j, DN_j \geq 0$ $\forall j$ (7)

Equation (1) in the model is the objective function, which seeks to simultaneously minimize physician dissatisfaction and patient load variability. The two user-defined weighting factors, α and β, are to be selected by the scheduler based on the priorities of the organization. Constraints (2) ensure that all physicians are assigned to the number of slots that they request, and constraints (3) ensure that the total number of examination rooms required by the assigned physicians in each slot does not exceed the total number of rooms available in the department or facility. Constraint (4) determines the ideal patient load by summing the total number of patients seen throughout an entire scheduling period and dividing by the total number of slots. Constraints (5) assign values to the variables used to represent the measure of patient load imbalance, DP_j and DN_j, for each slot. The difference of the actual number of patients seen and the ideal number of patients seen is assigned to DP_j or DN_j, depending on whether the difference is positive or negative. Finally, Constraints (6) define the assignment decision variables to be binary and constraints (7) ensure the workload balancing decision variables are non-negative.

When suitable values for the weighting factors are determined by the clinic, the linear math model described above will return an optimal solution for the physician-slot assignment problem. There are two main drawbacks to this model, however. Of the problems tested, an optimal solution could only be obtained within thirty minutes of runtime using LINGO (LINDO, 2008) for those involving a maximum of ten total physician-slot assignments. Also, the process of determining appropriate values for the weighting factors adds another dimension of complexity for the scheduler. Both of these obstacles can be overcome by making a simple adjustment to the model.

By removing the role of balancing patient load from the objective function and implementing it as a constraint, a new single objective linear mathematical model is created that can be solved optimally for medium and large sized problems. The adjusted model is shown and explained below.

Parameters:

S_{ij} = physician i's scaled satisfaction rating for slot j

SR_i = slots requested by physician i

RR_i = examination rooms required by physician i

PL_i = patient load, or average number of patients seen per slot by physician i

RC = room capacity

LC = patient load capacity

Decision Variables:

$$X_{ij} = \begin{cases} 1, & physician\ i\ is\ assigned\ to\ slot\ j \\ 0, & otherwise \end{cases}$$

Objective Function:

Maximize Physician Satisfaction $\quad$ Maximize $\sum_i \sum_j S_{ij} * X_{ij}$ $\quad$ (8)

Constraints:

All Physicians assigned to Requested Slots $\quad$ $\sum_j X_{ij} = SR_i$ $\quad$ $\forall i$ $\quad$ (9)

Space Capacity $\quad$ $\sum_i X_{ij} * RR_i \leq RC$ $\quad$ $\forall j$ $\quad$ (10)

Patient Load $\quad$ $\sum_i X_{ij} * PL_i \leq LC$ $\quad$ $\forall j$ $\quad$ (11)

Binary Variables $\quad$ $X_{ij} \in \{0,1\}$ $\quad$ $\forall i, j$ $\quad$ (12)

For fixed assignment(s) of physician I to slot j $\quad$ $X_{ij} = 1$ $\quad$ $\forall i, j$ $\quad$ (13)

Instead of assigning weighting values to physician satisfaction and patient load balancing, the adjusted model allows the scheduler to control the patient load capacity (LC), which is the maximum number of patients to be scheduled in any slot. The parameters of the new model provide a more explicable method of assigning priorities to satisfaction and balancing. Also, since patient load balancing is no longer part of the objective function, the main objective can now be defined more intuitively as maximizing total physician satisfaction instead of minimizing dissatisfaction. Of course, slight modifications also must be made to the numerical ordering of slot preferences in the physician survey so that higher numbers represent greater satisfaction.

The single objective, linear program described above can solve problems involving thousands of assignments optimally in an optimization software package such as LINGO within minutes. Since many clinics do not own such software packages, this research will focus on the development of a solution heuristic for the physician scheduling problem. In Chapter V, the mathematical model will be used to analyze the performance of the solution heuristic.

III. LITERATURE REVIEW

This section reviews literature on topics relevant to this study including general personnel scheduling, scheduling in healthcare, and simulated annealing.

A. General Personnel Scheduling

Employee scheduling has been thoroughly analyzed and researched over the past several decades by a wide variety of individuals including operations researchers, scientists, and managers. Although many studies have investigated topics such as optimizing the size or mix of the workforce, the majority of research in personnel scheduling concentrates on the allocation of jobs among a workforce such that costs and employee dissatisfaction are minimized, workload is distributed equitably, and all workplace constraints are satisfied. Researchers have been increasingly drawn into the study of scheduling as a result of the increasing pressures of a globally competitive environment and the shift to a more service oriented economy (Earnst *et al.*, 2004). The origins of staff scheduling can be traced as far back to when Leslie C. Eddie (1954) conducted research on the traffic delays at tollbooths. Since its inception, research on the staff scheduling problem has expanded and been applied to an assortment of application areas including but not limited to manufacturing, financial services, transportation centers, emergency services and health care systems. For a thorough and comprehensive explanation of the various applications of personnel scheduling, the author recommends Staff Scheduling and Rostering: A Review of Applications, Methods and Models (Ernst, 2004).

Scheduling in Healthcare

Because hospitals and clinics are constantly searching for ways to attract more business and retain clientele, much research has been conducted in the healthcare industry in efforts to identify value and reduce costs. Previous research has suggested that a major factor contributing to patient satisfaction during an outpatient care visit is waiting time. In a study conducted at the University of South Carolina Department of Family and Preventative Medicine, patients were more likely to be satisfied during a clinic visit if they believed themselves to be in good health, did not wait long, and had health insurance (Probst *et al.*, 1997). Of these three factors, clinics exercise the most control over patient waiting time. Low patient waiting time, low clinic overtime, and high patient throughput are a few obvious signs of efficient patient flow. Three clinic functions directly related to patient flow include patient scheduling, patient routing, and scheduling of personnel (Jun *et al.*, 1999). Although a majority of research in health care clinics has focused on patient scheduling in attempts to control demand, a sufficient amount of studies have also been conducted on personnel scheduling (Jun, 1999). Several simulation studies have shown that effective staffing strategies can help improve patient flow by reducing the inherent variability in healthcare systems (Kumar and Kapur, 1989; Lambo, 1983; Draeger, 1992). A major focus of personnel scheduling in healthcare has been in nurse scheduling, commonly referred to as nurse rostering in literature.

B. Nurse Rostering

Similar to the physician scheduling problem in this research, nurse rostering deals with obtaining a suitable schedule that covers demand while accommodating a range of employee preferences, observing work regulations, and ensuring a fair distribution of work (Ernst, 2004). In recent decades many hospital staffing problems were solved by hand, which was a very time consuming and intellectually challenging task (Burke *et al.*, 2004). Much of the research conducted in the 1970s and 1980s addressed various problem formulations and solution techniques, and focused on the development of support tools to aid in scheduling (Earnst, 2004). Numerous papers have been published on the development and use of computerized healthcare scheduling. The majority of early works focused on the use of mathematical programming methods for finding optimal solutions to linear models. In an early overview of the topic, Warner (1976) described how computer aided scheduling enables a more expeditious and complete search compared to traditional manual scheduling. A pioneering study by Warner and Prawda (1972) involved a mixed-integer quadratic program to calculate a minimum staffing requirement for nurses. One main drawback to this model is that it did not take individual preferences of nurses into consideration. Warner (1976) later presented a two-phase method of solving the nurse scheduling problem with consideration to nurse preferences. An initial feasible solution is obtained in phase I and improvements to this solution are sought in phase II. In another early study that provided a framework for future researchers to build upon, Abernathy *et al*. (1973) divided the staffing into three distinct stages: policy decisions, staff planning, and short-term scheduling. Several subsequent studies elaborated on previous formations to represent more realistic or particular situations. Aside from linear and mixed integer programming, new approaches have also been made to solve more complex rostering problems using a mix of simulation and heuristic techniques. More recently, researchers have begun to investigate methods of incorporating various meta-heuristics such as tabu search (Glover, 1990), genetic algorithms (Holland, 1973), and simulated annealing (Kirkpatrick *et al.*, 1983) in the rostering problem.

C. Simulated Annealing

The simulated annealing meta-heuristic has been used in numerous problem solving applications since its development by Kirkpatrick (1983). Similar to many other iterative improvement methods, simulated annealing steps from one solution to another, but the incorporation of temperature prevents the algorithm from becoming trapped in a local optimum by permitting uphill movements. The meta-heuristic is widely applicable due to its generality and comparatively low computational complexity. An example of the use of simulated annealing in healthcare scheduling is provided in Parr and Thompson's (2007) paper on nurse scheduling.

Simulated annealing has also proven to be useful in non-healthcare related applications, such as transportation systems. Lucic and Teodorovic (1999) used simulated annealing to develop an algorithm to solve the aircrew rostering problem. The algorithm developed was composed of two steps, similar to Warner's methodology used in nurse rostering (Warner, 1976). In the first step, a heuristic algorithm is used to generate an initial feasible solution. Then in the second step, the simulated annealing technique is used to improve the solution obtained in the first step.

This research pertains to a physician scheduling problem in an outpatient clinic with a structure similar to that of the nurse rostering problem. The solution methodology used in this study includes the simulated annealing metaheuristic and is similar to the two phase methods used earlier in similar applications. Additional information is provided on the solution methodology in the following chapter.

IV. SOLUTION METHODOLOGY

A. Greedy Assignment Algorithm

A greedy algorithm finds a solution by iteratively making the local optimal decision at each stage. Although greedy algorithms are not guaranteed to obtain the optimal solution, they have an expeditious execution time and often provide a satisfactory solution. For the physician-slot assignment heuristic, a greedy algorithm was selected to be used in the first phase of the model to determine an initial feasible solution. In this greedy algorithm, iterative physician assignment occurs as a function of the patient loads of physicians and patient load capacities of slots. The objective is to balance the total patient load among all slots so that feasible solutions are derived when the user-defined patient load capacity parameter is constricting. That is, when the user sets the patient load capacity to a relatively low value so that the variability of patient load amongst slots is minimal. This is fairly similar to the bin packing problem (Berkey and Wang, 1987), in which objects of various sizes must be placed into a finite number of fixed capacity bins with the objective of minimizing the total number of bins used. The physician-slot assignment problem is different in that the number of bins (slots) available is already known, and the objective is attempting to fill every bin to the same level.

The greedy assignment algorithm used in the physician-slot assignment problem first sorts the physicians by decreasing patient load, and then the assignment procedure proceeds as follows. The unassigned physician with the largest patient load is selected and assigned to the slot with the largest available patient load capacity, without any regard to physician satisfaction. Once a physician has been assigned to a slot, he or she is removed from further consideration.

That is, the heuristic only allows each physician to be assigned to one slot. The patient load and space capacities of the slots are updated after each assignment, and the procedure is repeated until all physicians have been assigned. If a physician(s) requests more than one slot assignment, a duplicate physician entity for each additional slot must be created in the program. In the case that a feasible physician assignment is prevented due to a violation of slot capacity constraints, a message informing the user of the hindrance is presented and the algorithm is terminated. Figure 2 presents the pseudocode for the greedy assignment algorithm.

```
Do Until each physician is assigned to one slot
        Find unassigned physician with largest unassigned patient load P1
        Find slot with largest available patient load capacity S1
                If S1 ≥ P1
                        Assign P1 to S1
                        Update patient load capacity and room capacity for S1
                Else If S1 < P1
                        Display informative message and terminate algorithm
                End If
Loop
```

FIGURE 2 – Pseudocode for Greedy Algorithm

The greedy algorithm described in this section is used in the first phase of the physician-slot assignment heuristic. The solution it returns is highly favorable in terms of the patient load balancing objective, although it is likely to be mediocre or poor in terms of the physician satisfaction metric. The second phase of the heuristic incorporates the use of simulated annealing (Kirkpatrick *et al.*, 1983), which attempts to reassign and swap physician assignments in order to increase the total physician satisfaction while remaining within the limits imposed by space and the user-defined patient load capacity. The trade-offs between patient load balancing and physician satisfaction become apparent in the second phase. If the user defines the patient load capacity to be very stringent, the opportunities for satisfying physicians decrease. The scheduler is capable of generating several competing schedules for evaluation by controlling and adjusting the patient load capacity.

B. Simulated Annealing

The simulated annealing metaheuristic operates in a very similar manner to the actual process of annealing in metallurgy (Kirkpatrick *et al.*, 1983). When heated metals are cooled at a gradual and controlled rate rather than quenched, the atoms within have more time to redistribute into lower energy configurations. Metallurgists use the annealing process to improve the homogeneity of metals, making them more ductile and workable.

Comparable to the physical process, the simulated annealing metaheuristic can be used to improve a predetermined solution to a math problem by incorporating randomness and providing the possibility of departure from a local optimal region. The algorithm first calculates the difference of the objective function values of the present solution and a nearby solution, δ. If the difference is favorable, the present solution is discarded and the new solution is accepted. If the difference is unfavorable, the new solution is still accepted with a probability of $e^{\frac{-\delta}{T}}$. The temperature variable T is set to a relatively large value initially, then gradually reduced so that the probability of accepting an inferior solution decreases as the algorithm proceeds to termination. The initial high value of T allows the algorithm to depart from the current solution and explore other regions, similar to the physical reorganization of atoms during the annealing of metals (Heragu, 2008).

The user has limited control over the performance of the simulated annealing algorithm with the assignment of four parameters: *T, R, ITEMP, and NOVER*. *T* is the initial temperature and as mentioned above, is typically set to a relatively large value to allow for departure from a local optimum. The cooling factor *R,* a multiplier with positive values less than or equal to one, specifies the rate at which temperature *T* will be decreased. Naturally, larger values for *R* provide for a more gradual reduction in *T*. *ITEMP* specifies the number of times that the initial temperature *T* is to be decreased, and *NOVER* defines the maximum number of new solutions to be evaluated at each temperature. These four parameters must be selected with a consideration of the trade-off between solution quality and run-time (Heragu, 2008).

In the first phase of the physician-slot assignment model, an initial feasible solution is determined using the *greedy algorithm* described in the previous section. The simulated annealing metaheuristic is then used to explore other regions of the solution space in efforts to improve the objective function value. In order to generate nearby solutions, the algorithm chooses a random physician and a random slot and moves the physician from their currently assigned slot to the new slot if space capacity and patient load capacity constraints will not be violated. If constraints prevent this move from taking place, the algorithm will randomly select a second physician and attempt to swap the slot assignments of the two randomly selected physicians. Each time a new solution is created, the difference (δ) between the present solution and the new solution is calculated and the acceptance procedure mentioned previously is executed. The extent to which this process is repeated and the resulting runtime of the algorithm are dependent upon the selected values for the four parameters mentioned above. Figure 3 below shows the pseudocode for the simulated annealing heuristic.

```
Set temperature T equal to objective function value of current solution OFV1
Do Until T < 0.1
 T=TR
        Do NOVER times
         Solution 2 (X2) = Solution 1(X1)
        OFV2= 0
        Choose random physicians P1 and P2 and random slot S1
                If S1 ≥ P1, move P1 to S1 and update X2
                        Calculate OFV2 and set δ = OFV1 – OFV2
                        Set X2=X1 if δ < 0 or with a probability of e^(−δ/T)
                        Update slot capacities
                Else, if feasible, swap assignments of P1 and P2
                        Caclulate OFV2 and set δ = OFV1 – OFV2
                        Set X2=X1 if δ < 0 or with a probability of e^(−δ/T)
                        Update slot capacities
                End If
        Loop
Loop
```

FIGURE 3 – Pseudocode for Simulated Annealing Metaheuristic

Empirical studies were carried out to determine suitable values for the four performance parameters. The data sets used in the results section (Chapter V) of this report were run using several combinations of values to determine which combination provided the most favorable solution results within a reasonable run-time. As a result, the initial temperature *T* was set to be equivalent to the objective function value of the initial greedy heuristic and the cooling factor was set at 0.95 so that *T* is reduced gradually. A value of 1000 was selected for *NOVER* so that a considerable amount of new solutions would be generated at each temperature. Instead of selecting a numeric value for *ITEMP*¸ the algorithm is set to run until the value of Temperature is reduced to less than 0.1. This adjustment ensures that algorithm will have a consistent and effective termination point for problems of various sizes.

V. RESULTS

Eight data sets, made up of different combinations of slots and physicians and modeled to represent realaistic situations, were generated to compare the results of the solution heuristic to the optimal. A sample data set is provided in Figure 6 at the end of this chapter. The optimal results for the data sets were obtained by using LINGO (LINDO, 2008), an optimization modeling software, to solve the linear mathematical model described previously in Chapter II. Recall that by converting patient load balancing from a part of the objective function to a definite limiting constraint, the mathematical model becomes solvable for relatively large sized problems involving hundreds of assignments within seconds. The solution heuristic was executed five times for each problem and the best solution was documented.

Before presenting the information relevant to this section, it is necessary to describe the quantitative descriptor that is used to characterize problem parameters. Although the patient load capacity is defined by the scheduler, the constraining effect of this parameter is obviously dependent upon the sum of the patient loads of all physicians to be scheduled. The Patient Load Percentage (PLP) is used to represent the constraining effect as a percentage of the ideal, completely level load. To understand how the PLP is calculated, consider a situation in which the sum of the patient loads for all physicians requesting a slot at a clinic is 100. The clinic operates on a five-day weekly schedule of ten slots, two slots per day. The ideal and completely level load could be accomplished if physicians are capable of being scheduled so that exactly 10 patients are seen in each slot. To achieve this balance using the solution heuristic or math model, the patient load capacity parameter must be set to 10 patients. If a feasible solution does exist, the associated PLP would be 0% since there is no excess capacity available. However, if the scheduler decides to allow more variability in patient load among slots in an effort to increase physician satisfaction, the patient load capacity could be increased to 12 or 13 patients, resulting in PLP's of 20% and 30% respectively. The PLP represents the additional percentage of the ideal patient load that is allowed in each slot, and it will be used to represent the magnitude of patient load balancing in each problem tested.

The results analysis begins with the generation and testing of two small data sets at various levels of PLP. The first data set consisting of twenty physicians and five slots could realistically represent an outpatient center that operates five days per week and assigns each physician to a specific day. The results are shown in the table below.

TABLE I

RESULTS FOR SMALL DATA SETS

Problem Size (Physicians, Slots)	Patient Load Percentage (PLP)	Optimal	Heuristic	% Difference
20, 5	31	98	98	0.00%
20, 5	16	97	97	0.00%
20, 5	10	97	97	0.00%
20, 5	3	95	95	0.00%
30, 10	30	294	294	0.00%
30, 10	16	293	293	0.00%
30, 10	10	290	289	0.34%
30, 10	4	286	284	0.70%
30, 10	1	278	278	0.00%

The heuristic was able to find an optimal solution in seven out of the nine tested cases. In the two cases that the heuristic did not find the optimal solution, the greatest percent difference from optimal was 0.7%. Considering all nine cases together, the average percent difference from optimal is 0.116%. The average runtime for the heuristic was 20.4 seconds.Computer runtimes for any program are clearly dependent upon the specifications of the machine used. Information pertaining to runtimes in this section was obtained using a Dell Workstation PWS370 PC with an Intel 3.4 GHz processor and 1 GB of RAM.

In order to gain a better understanding of what factors affect performance, solutions for medium sized data sets were generated and tested next. The data sets with eight slots are intended to represent outpatient centers operating on a four day work week with two slots per day. Table II shows the results.

TABLE II

RESULTS FOR MEDIUM DATA SETS

Problem Size (Physicians, Slots)	Patient Load Percentage (PLP)	Optimal	Heuristic	% Difference
75, 10	31	743	743	0.00%
75, 10	16	738	737	0.14%
75, 10	10	736	733	0.41%
75, 10	4	729	726	0.41%
100, 10	30	995	990	0.50%
100, 10	16	990	987	0.30%
100, 10	7	983	978	0.51%
100, 10	1	975	958	1.74%
75, 8	30	597	597	0.00%
75, 8	16	595	595	0.00%
75, 8	10	592	591	0.17%
75, 8	4	589	588	0.17%
100, 8	30	800	800	0.00%
100, 8	16	797	797	0.00%
100, 8	10	796	796	0.00%
100, 8	2	792	791	0.13%

When tested with medium data sets, the heuristic found the optimal solution in six of 16 cases. The trending of the percent differences from optimal in Table II hints to the fact that the patient load balancing may have an effect on the overall performance of the heuristic. In many cases, decreasing values of PLP result in an increase in the solution difference from optimal. Also, in the case of an extremely balanced patient load with PLP equal to 1%, the percent difference from optimal exceeds three times that of all other tested cases. The heuristic required an average of 100.9 seconds to solve the medium sized data sets. To better understand the effects of problem size on performance, data sets were generated and tested for larger problems. The results are shown in Table III below.

TABLE III

RESULTS FOR LARGE DATA SETS

Problem Size (Physicians, Slots)	Patient Load Percentage (PLP)	Optimal	Heuristic	% Difference
200, 10	30	1991	1987	0.20%
200, 10	16	1981	1977	0.20%
200, 10	8	1970	1961	0.46%
200, 10	1	1956	1940	0.82%
300, 8	30	2398	2398	0.00%
300, 8	16	2390	2389	0.04%
300, 8	10	2385	2381	0.17%
300, 8	2	2374	2361	0.55%

With an average percent difference from optimal of 0.31%, the overall performance of the heuristic appears to be consistent across data sets of all sizes. The runtime for large data sets increased significantly, having an average of 353.3 seconds. The number of physicians, or assignments, seems to be the factor having the largest affect on runtime. Figure 4 below represents the average trend in computation time as a function of the number of physicians to be scheduled.

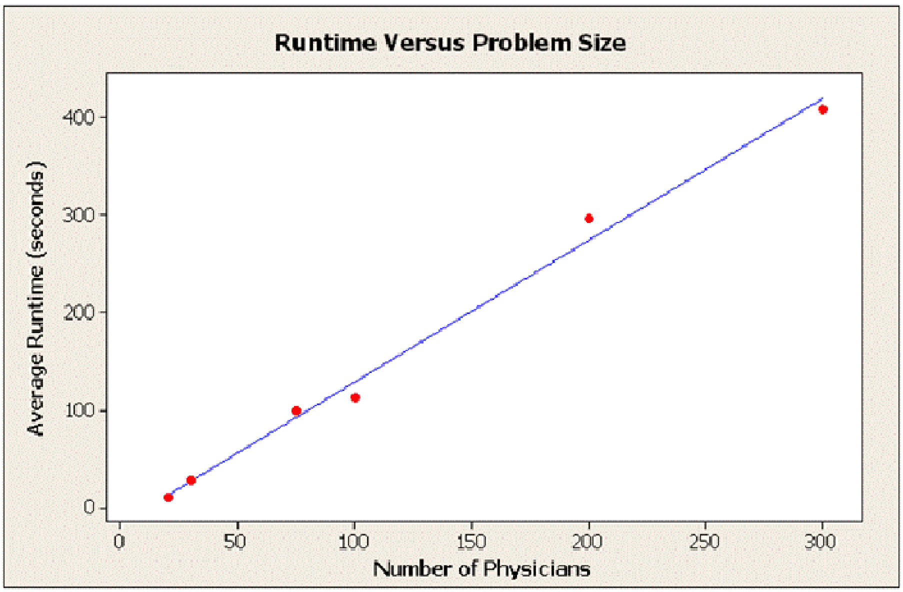


FIGURE 4 – Plot of Runtime Versus Problem Size

A linear relationship exists between the number of physicians and average heuristic runtime. This can be attributed to selection of performance parameters for the heuristic since parameters were selected so that the runtime is a function of the problem size. Because the algorithms within the heuristic undergo more iterations as the problem size increases, consistent solution quality is maintained at the expense of escalating execution time. LINGO was able to find the optimal solution for all data sets within two seconds.

As mentioned previously, the physician-slot assignment heuristic is modeled to handle fixed assignments. That is, the scheduler is capable of fixing any number of physicians to specific slots provided that space or patient balancing constraints are not violated. To assess the performance of the heuristic with the use of varying amounts of fixed assignments, a medium sized problem was generated and executed at various combinations of PLP. Table IV shows the results for increasing numbers of fixed assignments.

TABLE IV

RESULTS FOR MEDIUM DATA SET WITH FIXED ASSIGNMENTS

Number of Fixed Assignments	Problem Size (Physicians, Slots)	Patient Load Percentage (PLP)	Optimal	Heuristic	% Difference
5	75, 10	31	742	742	0.00%
5	75, 10	16	736	735	0.14%
5	75, 10	10	730	729	0.14%
10	75, 10	31	742	741	0.13%
10	75, 10	16	736	735	0.14%
10	75, 10	10	730	728	0.27%
20	75, 10	31	742	742	0.00%
20	75, 10	16	736	735	0.14%
20	75, 10	10	729	729	0.00%
40	75, 10	31	741	741	0.00%
40	75, 10	16	733	733	0.00%
40	75, 10	10	727	726	0.14%

The number of fixed assignments does not have a noticeable relation to heuristic performance as Table IV shows that the solution obtained by the heuristic was within 0.27% of the optimal in all tested cases. As the number of fixed assignments increases, the heuristic actually performs better in the trials selected. This may be ascribed to the fact that the heuristic has fewer alternative solutions to consider as the number of fixed assignments increase. The average runtime for these problems was 106.8 seconds. As expected, the optimal value for physician satisfaction decreases as patient load is balanced, or with a decrease in PLP.

Overall, the solution heuristic provided adequate solutions for nearly all data sets tested, with the exception of one case with a PLP of 1% in which the difference from optimal was still within 2%. Of the 45 scenarios tested, the heuristic was able to find 19 optimal solutions within five replications. Also, solution quality does not appear to be significantly affected by problem size or number of fixed assignments. Although execution time does increase with problem size, the largest data sets tested were considered realistic extremes and solved within seven minutes.

Figure 5 below provides a graph of the solution heuristic's highest objective function value obtained at each temperature of simulated annealing for a medium sized data set containing 75 assignments. As the temperature is gradually reduced, fewer unfavorable solutions are accepted and the objective function value (total physician satisfaction) improves. Figures 6 and 7, shown on the following pages, provide screen shots of the heuristic tool's input and output screens. The example problem used in the screen shot involves the assignment of 20 physicians to 5 slots, with one fixed assignment of physician 1 to slot 2.

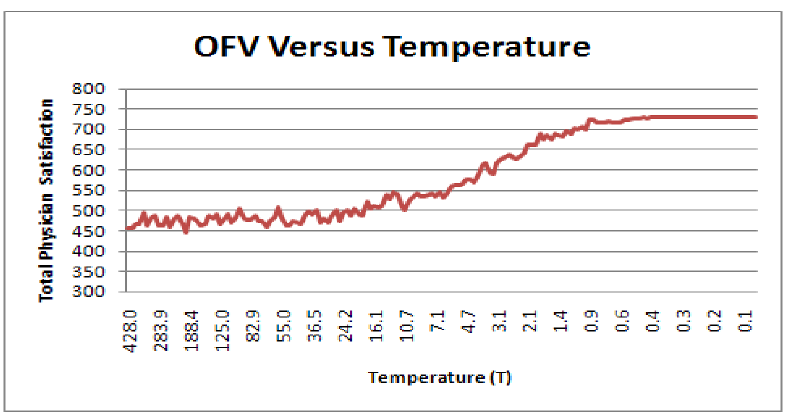
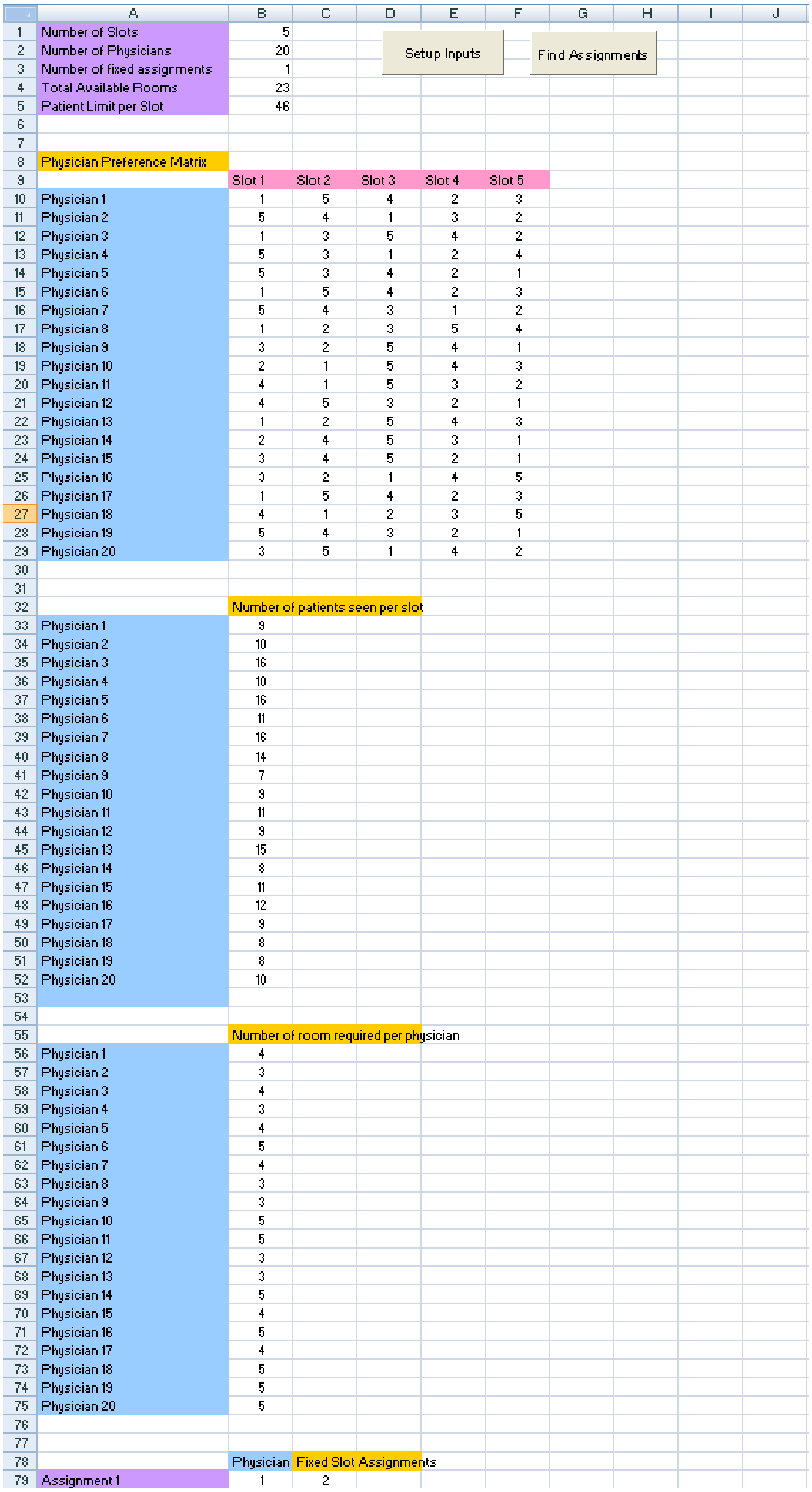


FIGURE 5 –Solution Improvement During Simulated Annealing



Setup Inputs

Find Assignments

	A	B	C	D	E	F
1	Number of Slots	5				
2	Number of Physicians	20				
3	Number of fixed assignments	1				
4	Total Available Rooms	23				
5	Patient Limit per Slot	46				
6						
7						
8	Physician Preference Matrix					
9		Slot 1	Slot 2	Slot 3	Slot 4	Slot 5
10	Physician 1	1	5	4	2	3
11	Physician 2	5	4	1	3	2
12	Physician 3	1	3	5	4	2
13	Physician 4	5	3	1	2	4
14	Physician 5	5	3	4	2	1
15	Physician 6	1	5	4	2	3
16	Physician 7	5	4	3	1	2
17	Physician 8	1	2	3	5	4
18	Physician 9	3	2	5	4	1
19	Physician 10	2	1	5	4	3
20	Physician 11	4	1	5	3	2
21	Physician 12	4	5	3	2	1
22	Physician 13	1	2	5	4	3
23	Physician 14	2	4	5	3	1
24	Physician 15	3	4	5	2	1
25	Physician 16	3	2	1	4	5
26	Physician 17	1	5	4	2	3
27	Physician 18	4	1	2	3	5
28	Physician 19	5	4	3	2	1
29	Physician 20	3	5	1	4	2
30						
31						
32		Number of patients seen per slot				
33	Physician 1	9				
34	Physician 2	10				
35	Physician 3	16				
36	Physician 4	10				
37	Physician 5	16				
38	Physician 6	11				
39	Physician 7	16				
40	Physician 8	14				
41	Physician 9	7				
42	Physician 10	9				
43	Physician 11	11				
44	Physician 12	9				
45	Physician 13	15				
46	Physician 14	8				
47	Physician 15	11				
48	Physician 16	12				
49	Physician 17	9				
50	Physician 18	8				
51	Physician 19	8				
52	Physician 20	10				
53						
54						
55		Number of room required per physician				
56	Physician 1	4				
57	Physician 2	3				
58	Physician 3	4				
59	Physician 4	3				
60	Physician 5	4				
61	Physician 6	5				
62	Physician 7	4				
63	Physician 8	3				
64	Physician 9	3				
65	Physician 10	5				
66	Physician 11	5				
67	Physician 12	3				
68	Physician 13	3				
69	Physician 14	5				
70	Physician 15	4				
71	Physician 16	5				
72	Physician 17	4				
73	Physician 18	5				
74	Physician 19	5				
75	Physician 20	5				
76						
77						
78		Physician	Fixed Slot Assignments			
79	Assignment 1	1	2			

FIGURE 6 – Screen Shot of Heuristic Tool Input Screen

	A	B	C	D	E	F	G	H
1	Run Time (s)	7.890625						
2	Total Satisfaction	93						
3								
4								
5		Slot 1	Slot 2	Slot 3	Slot 4	Slot 5	Phys Sat	
6	Physician 1	0	1	0	0	0	5	
7	Physician 2	1	0	0	0	0	5	
8	Physician 3	0	0	0	1	0	4	
9	Physician 4	0	0	0	0	1	4	
10	Physician 5	1	0	0	0	0	5	
11	Physician 6	0	1	0	0	0	5	
12	Physician 7	0	1	0	0	0	4	
13	Physician 8	0	0	0	1	0	5	
14	Physician 9	0	0	1	0	0	5	
15	Physician 10	0	0	1	0	0	5	
16	Physician 11	0	0	1	0	0	5	
17	Physician 12	1	0	0	0	0	4	
18	Physician 13	0	0	0	1	0	4	
19	Physician 14	0	0	1	0	0	5	
20	Physician 15	0	0	1	0	0	5	
21	Physician 16	0	0	0	0	1	5	
22	Physician 17	0	0	0	0	1	3	
23	Physician 18	0	0	0	0	1	5	
24	Physician 19	1	0	0	0	0	5	
25	Physician 20	0	1	0	0	0	5	
26	Total Patients	43	46	46	45	39		
27	Total Rooms	15	18	22	10	17		
28								

FIGURE 7 – Screen Shot of Heuristic Tool Output Screen

VI. CONCLUSIONS AND FUTURE RESEARCH

This research was conducted for the purposes of developing a method to assist in the scheduling of physicians in an outpatient center. Provided that many organizations do not already own and are not inclined to purchase optimization software capable of solving this model, this research focused on the development of a heuristic solution procedure solvable in more familiar and available programs, such as Microsoft Excel. A dual objective mathematical model was initially constructed so that the performance of the heuristic could be evaluated. Because this model was only capable of solving small problems within a reasonable amount of time, adjustments were made that converted the model into a single objective optimization solvable for large problems. This math model was used to analyze the performance of the solution heuristic, which uses an initial greedy assignment algorithm followed by the Simulated Annealing meta-heuristic. The performance of the heuristic was tested by solving a variety of randomly generated data sets and comparing the results to the optimal values obtained by solving the mathematical model in LINGO.

The results provided in Chapter V show that the heuristic was capable of finding an optimal solution in 42.2% of the tested cases. When an optimal solution was not obtained, the average difference from optimal was 0.35%. The results also indicate that an increasing problem size and number of fixed assignments does not have a notable negative impact on solution quality. Although runtime does increase linearly with problem size, data sets created to represent realistic scheduling conditions were solved within seven minutes.

The satisfactory results of the heuristic suggest that it could be beneficially used as a supplemental tool to schedulers of outpatient centers at virtually no cost. For instance, an organization currently utilizing a five day work schedule with excess capacity may find by experimentation that a four day work schedule could provide improved work load balance and physician satisfaction. Because this research focused on the operational characteristics of a particular outpatient center, the strategies used may not be applicable to other clinics functioning differently. Also, this research does not consider the assignments of full-time medical assistants to physicians, a critical aspect of many outpatient centers. It is recommended for future research to be conducted in order to explore the possibility of integrating medical assistant workload balancing as an additional consideration to the physician scheduling problem.

LIST OF REFERENCES

Abernathy, W.J. Baloff, N., Hershey, J.C., and Wandel, S. 1973. A three-stage manpower planning and scheduling model-A service-sector example. *Operations Research*. 22: 693-711.

Berkey, J., Wang, P. 1987. Two-dimensional finite bin-packing algorithms. *Journal of the Operational Research Society.* 38(5): 423-429.

Burke, E. Causmaecker, P., Vanden Berghe, G., Van Landeghem, H., 2004. The state of the art of nurse rostering. *Journal of Scheduling.* 4: 441-499.

Cullen, K., Hall, M. Golosinskiy, A., 2009. Ambulatory surgery in the United States - Revised, *National Health Statistics Reports*.

Draeger, M.A. 1992. An emergency department simulation model used to evaluate alternative nurse staffing and patient population scenarios. *Porceedings of the 1992 Winter Simulation Conference.* IEEE. Arlington, VA. 13-16 December. 1057-1064.

Earnst, A.T., Jiang, H., Krishnamoorthy, M., Sier, D. 2004. Staff scheduling and rostering: A review of applications, methods, and models. *European Journal of Operations Research.* 153: 3-27.

Edie, L. 1954. Traffic delays at toll booths. *Journal Operations Research Society of America.* 2(2): 10–137.

Glover, F. 1990. Tabu search – A tutorial. *Interfaces.* 20(4): 74-94.

Heragu, Sunderesh. 2008. *Facilities Design-3rd Edition.* Boca Raton: CRC Press

Holland. J. 1973. Genetic algorithms and the optimal allocation of trials. *SIAM J. Comput.* (2)2: 88-105.

Jun, J.B., Jacobson, S.H., Swisher, J.R. 1999. Application of discrete-event simulation in health care clinics: a survey. *The Journal of the Operational Research Society.* 50(2): 109-123.

Kirkpatrick, S., Gelatt, C.D., Vecchi, M.P. 1983. Optimization by Simulated Annealing. *Science.* 220: 671-680.

Kumar, A.P. and Kapur, R. 1989. Discrete simulation application-scheduling staff for the emergency room. *Proceedings of the 1989 Winter Simulation Conference*. IEEE. Washington, DC. 4-6 December. 1112-1120.

Lambo, E. 1983. An optimization-simulation model of a rural health center in Nigeria. *Interfaces*. 13: 29-35.

LINDO Systems, 2008. LINGO (Version 11) [Computer Software].

Lucic, P., Teodorovic, D. 1999. Simulated annealing for the multi-objective aircrew rostering problem. *Transportation Research Part A – Policy and Practice.* 33(1): 19-45.

Parr, D., Thompson, J.M. 2007. Solving the multi-objective nurse scheduling problem with a weighted cost function. *Operations Research.* 155: 279-288.

Probst, J.C., Greenhouse, D.L., Selassie, A.W. 1997. Patient and physician satisfaction with an outpatient care visit. *Journal of Family Practice*. 45(5): 418–425.

Warner, M. and Prawda, J. 1972. A mathematical programming model for scheduling Nursing personnel in a hospital. *Management Science*. 19: 411-422

Warner, M., 1976. Scheduling Nursing Personnel According to Nursing Preference: A mathematical programming approach. *Operations Research.* 24: 842-856.

Warner, M. 1976. Nurse staffing, scheduling, and reallocation in hospitals. *Hosptial & Health Services Administration.* 21(3): 77-90.

VITA

Dustin Jacob Banet

Djbane01@louisville.edu

EDUCATION

Masters of Engineering in Industrial Engineering……………………………………May 2010
University of Louisville, Louisville, KY

Bachelors of Science in Industrial Engineering………………………………………May 2009
University of Louisville, Louisville, KY