REPORT DOCUMENTATION PAGE

AFRL-SR-AR-TR-04-

0296

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	**2. REPORT TYPE** Final	**3. DATES COVERED** *(From - To)* 15 Jan 2001 - 31 December 2003

4. TITLE AND SUBTITLE Air Mobility Research for Undergraduates	**5a. CONTRACT NUMBER**
	5b. GRANT NUMBER F49620-01-1-0172
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S) Ervin Y. Rodin	**5d. PROJECT NUMBER**
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Center for Optimization and Semantic Control Campus Box 1040, Washington University St. Louis, MO 63130-4899	**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES) Air Force Office of Scientific Research 4015 Wilson Blvd Mail Room 713 Arlington, VA 22203 NM	**10. SPONSOR/MONITOR'S ACRONYM(S)** AFOSR
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Distribution Statement A. Approved for public release; distribution is unlimited.

BEST AVAILABLE COPY

20040617 066

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This report deals with the simulation of medical systems, both within the military and the civilian world, complete with scheduling problems and employment of optimal techniques in order to improve clinic performance based upon several criteria including time in clinic per patient and assets required to serve patient population. Problems of this sort fall into a category known as hybrid systems, which are characterized as problems having both discrete and continuous elements. Additionally, hybrid systems are characterized by having large numbers of interlocking subsystems, and changes made to any one of these causes rippling effects throughout all of the others. In the case of a medical system, such interactive subsystems include the doctors, nurses, technicians, and patients, in concert with record keeping, layout, other hospital clinics, lab work, medications, and tools used in the clinic such as EKG machines and portable X-ray machines.

15. SUBJECT TERMS

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON** Ervin Y. Rodin
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			**19b. TELEPHONE NUMBER** *(Include area code)*
U	U	U	UU		

Standard Form 298 (Rev. 8/98)
Prescribed by ANSI Std. Z39.18

FINAL REPORT

Submitted to
Air Force Office of Scientific Research
801 North Randolph Street
Arlington VA 22203-1977

by

Ervin Y. Rodin, P.I.
Professor and Director

Center for Optimization and Semantic Control
Department of Systems Science and Mathematics
Campus Box 1040, Washington University
One Brookings Drive
St. Louis, MO 63130-4899

in connection with
Grant AFOSR F49620-01-1-0172

AIR MOBILITY RESEARCH FOR UNDERGRADUATES

May 19, 2003

Background

The original proposal for this project, which was to involve undergraduate students in certain research aspects of the work of the Air Mobility Command and the US Transportation Command, both at Scott AFB, IL, was submitted in September 2000. This was not expected to be any unusual project for us: many of our undergraduate students have performed varieties of tasks during years past for both of these organizations.

By the time that this proposal was approved and granted, and then an appropriate group of students was found to work on it, it was the middle of 2001. We then had several preparatory meetings with individuals from the above two organizations, as was detailed in our progress reports; and several specific projects were in fact selected, all relating to military transportation problems. It was expected, however, that – as has been our past procedure – the students will have regular meetings with their "sponsors" both from AMC and from TRANSCOM.

Unfortunately, 9/11 and its aftermath made such meetings impossible: it would not have been possible to fulfill the suddenly increased requirements for security clearances for the students who were to be involved in the projects.

Fortunately, we had been approached by the USAF medical branches also to provide assistance for them, and we were ready to do so. Therefore, due to the press of circumstances, we shifted our research direction from military transportation to military medicine.

In particular, we were able to develop very successful simulations for the Internal Medicine Clinic, and later for the Emergency Services, of the 375th Medical Group at Scott AFB.

In this Final Report we are including an evaluation of our work by the Commander of the 375th, Col Andrew Colon. We are also including a report about the latter segments of this project (which, when expanded and extended, may become part of the doctoral dissertation of one of the students involved).

Finally, it may be appropriate to mention that as a result of our success with the projects at the 375th Medical Group at Scott AFB, we were approached by one of our large civilian hospitals, Missouri Baptist Medical Center, to do a similar study for their rather large Emergency Department. That work is continuing and is expected to be concluded soon. In addition, we are also assisting the Surgery Department at Duke University in their Optimal Scheduling problems. Both of these are mentioned briefly in the attached report.

DEPARTMENT OF THE AIR FORCE
HEADQUARTERS 375TH MEDICAL GROUP (AMC)

FEB 7 2004

Colonel Andrew Colon
375th Medical Group
310 W. Losey Street
Scott Air Force Base, Illinois 62225-5252

Professor Ervin Y. Rodin, Director
Center for Optimization and Semantic Control
Department of Systems Science and Mathematics
Washington University, Campus Box 1040
One Brookings Drive
St. Louis, MO 63130-4899

Dear Dr. Rodin

It has been a great privilege for the 375th Medical Group at Scott Air Force Base, IL to partner with your Washington University Center for Optimization and Semantic Control (COSC) in the conduct of optimization research into operational medicine methods and procedures. For more than 3 years, your talented graduate and undergraduate students have left their footprints throughout our Air Force hospital and its associated clinics, beginning with the Internal Medicine Clinic, the Primary Care Clinic, and, most recently, our Emergency Department. They have brought a disciplined approach and sophisticated engineering "toolkit" to bear on our processes in a manner far beyond our expertise with important results. Already, the model which they built, and the simulations run, for the Internal Medicine Clinic have resulted in important modifications of traffic flows, and resource streams, and procedural algorithms. The analysis which your students have recently performed in our Emergency Department promises to yield even more significant optimizations.

This partnership could not have happened at a better time. In pursuit of better service to our patients, and better financial return on every hard-earned taxpayer dollar, the Air Force Medical Service (AFMS) has embarked on an across-the-board attempt to optimize clinical and business processes. The 375th Medical Group is an important part of that global AFMS optimization process. The "Wash U connection" gives us access to engineering and mathematical modeling and simulation techniques far more powerful than we could ever hope to employ on our own.

The exposure of our Family Practice residents and other medical personnel to your students and faculty has been most beneficial. Reciprocally, the ability of your students to have access to a "live, operational healthcare delivery platform" we think is, likewise, beneficial. We look forward to continuing and further expanding this partnership far into the future.

Sincerely

AColon

ANDREW COLON, Colonel, USAF, BSC
Commander

I. Introduction and Background

This report deals with the simulation of medical systems, both within the military and the civilian world, complete with scheduling problems and employment of optimal techniques in order to improve clinic performance based upon several criteria including time in clinic per patient and assets required to serve patient population. Problems of this sort fall into a category known as hybrid systems, which are characterized as problems having both discrete and continuous elements. Additionally, hybrid systems are characterized by having large numbers of interlocking subsystems, and changes made to any one of these causes rippling effects throughout all of the others. In the case of a medical system, such interactive subsystems include the doctors, nurses, technicians, and patients, in concert with record keeping, layout, other hospital clinics, lab work, medications, and tools used in the clinic such as EKG machines and portable X-ray machines.

Systems of this sort completely defy traditional forms of optimization. Because they contain large numbers of random events, which over the course of time unfold as stochastic processes, any deterministic model will obviously fail. Therefore, a different approach was needed, one which can take into account both stochastics and the auto-interaction of the system. A simulative approach was chosen for its ability to capture precisely those characteristics. However, a simulative approach alone would not necessarily be sufficient to advance the understanding of such systems.

Simulation is an excellent tool for asking 'what if' style questions about alterations to a system, but must be used interactively in order to test and refine improvements. The goal presented in the paper will be to include additional techniques, both embedded into simulation and subsequent to simulation, in order to provide some automated, intelligent improvement to the systems. Simulation has been used in many capacities to study hybrid systems in the past. In addition to medical systems, simulative techniques have been used to study airfields, transportation systems such as the TSP (Traveling Salesman Problem) and its numerous derivatives. Until recently, simulation has not been coupled with true optimal techniques to improve system performance. This report will describe an extension of previous embedded optimal techniques, and a new coupling of simulation results with semantic controls.

Semantic control is a method for optimizing systems in which dynamic decision making must be made by human elements of a system. Automated elements of a system may interact with human elements, and provide decision-influencing information by offering intelligent strategies for system operation in plain language. This is then processed by human decision makers. This method has been incorporated in the past into systems such as combat aircraft, allowing an on board computer to suggest combat or evasion strategies to the human pilot. He then uses that information to engage in operations. In the case of a medical system, for example, an emergency department, a semantic control can be used to suggest to medical personnel the order of patient treatment to minimize the expected time each patient will spend in the ER. When a doctor makes an initial assessment, the information regarding necessary treatment can be entered into a pre-designed spreadsheet, which then compares all current patients and suggests a priority for each patient, essentially suggesting the order in which to treat the patients currently in the system. The doctor is then able to examine the suggested order and accept, reject or modify the semantic control.

The semantic controls are determined by simulative results of the system in question. The particular results for the 375th Medical Group Emergency Department at Scott Air Force Base will be detailed further on. Work incorporating similar techniques for the civilian emergency department at Missouri Baptist Hospital in St. Louis is ongoing.

II. Description of 375th Medical Group Project

The 375th Medical Group Emergency Department desired a tool that could be used to analyze the functioning of its Emergency Room (ER) in a cost free environment. The nature of large systems such as hospitals is that making any significant alteration is often extraordinarily expensive. Therefore, it is necessary before making changes to the system that it be known beforehand what the actual results of those changes will be. It is also in the nature of any hybrid system that changes to the system may have unintended consequences, and improvements to one area of the system may in fact cause deterioration of system performance overall. Because systems of these types defy traditional optimization schema such as linear or mixed integer programming, a simulative approach was determined to be the most likely to provide an avenue for performance enhancement.

Beginning in May of 2003, the author spent three months inhabiting the ER and interacting with the resources there. After an initial period of becoming familiar with the flow of patients, paperwork and ER staff, data was collected on the workings of the ER. Distributions formulated include, but are not limited to: time spent between doctors, nurses, medical technicians (MTs)[1] and patients; time required for each resource to process paperwork or computer work; distribution of phone calls, patient interarrival and ambulance interarrival; time required for EKG and X-ray. Similarly, the percentage of patients requiring EKG, X-ray, lab work and prescriptions was also collected.

III. Development of Distributions

Once the appropriate data collection was completed, statistical distributions were generated. As an example, the amount of time each doctor spent using the telephone is shown below.

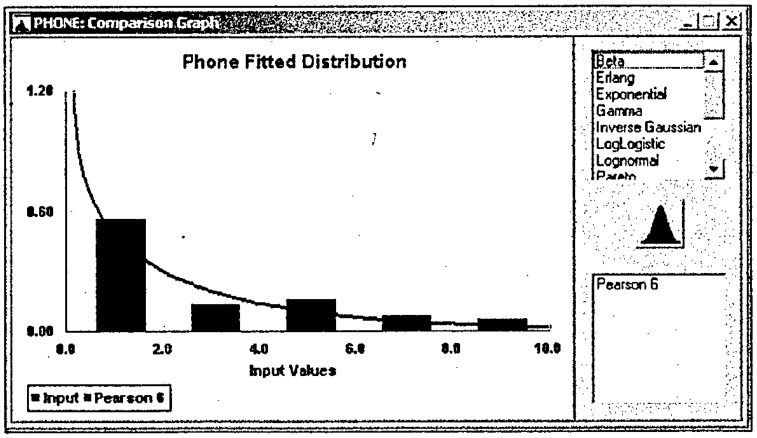


Figure 1: Distribution of Doctor's Time on Telephone

---

[1] Collectively referred to as 'resources' or 'assets'.

The fitted distribution, in this case a Pearson 6 distribution, correlates to the data collected at 97.1%. Every distribution in the simulation correlates to observed phenomena at least 95.0%. These are universally considered to be excellent co-relations for simulations of random events.

IV. Development of Simulation

Once the distributions had been calculated, design of the simulation could proceed. For this purpose, the simulation software Promodel was selected. Movement paths were laid down according to the dimensions of the ER, resources modeled, and arrival streams of patients, phone calls, and computer documents were formulated according to the collected data. Interactions between patients and resources, as well as among the resources themselves, and resource use of paperwork, were included in order that the maker of the model might more accurately capture the workings of the ER.

A typical patient encounter in the ER consists of several elements. The following description is shown in figure 2. First, the patient arrives at the entrance (or, occasionally, by ambulance) and proceeds to triage. A triage nurse evaluates the patient and places them in a queue for admittance to the ER. The nurse then creates a paper record for the patient and calls for a medical technician to bring the record to the ER desk and place it in the order of the severity of the patient's presenting problem. When available, an ER nurse or doctor will then review the records and request for a particular patient to be admitted, generally, but not always, in the order of their triage. While patients wait in the waiting room, 3.3% will leave without being seen.

Once a doctor or nurse calls for a patient to be brought to a treatment room, an MT checks the room to be sure it is clean and stocked, collects the patient and performs an initial interview and basic physical check. The record is placed in a slot corresponding to room number where it is reviewed first by a nurse, and then the ER doctor, each of whom perform their own initial assessment of the patient. The doctor will then determine what labs, if any, are needed, and whether the patient needs a radiology consult, and x-ray examination. If so, some patients can be x-rayed with the ER's portable X-ray machine, while others must be accompanied to the radiology department by an MT.

The doctor additionally will determine if any procedures (sutures, spinal tap, injections, etc.) need to be administered, and if so, by whom. An MT is capable for example of suturing, or removing sutures, or giving injections, but a nurse or doctor is required for more delicate or difficult procedures such as a spinal tap. Doctors also order EKGs, write prescriptions and decide if a patient needs to be admitted into the in-patient ward. If an admission is recommended, the ward doctor is called and consults with the patient, and the presiding ER doctor. The process of admitting a patient is by far the most time consuming common task in the ER.

Once all procedures have been completed, the patient is either discharged or admitted to the wards. It is extremely uncommon for a patient, once admitted to the ER treatment room, to leave against medical advice; only one occurrence was observed in the three month long observation period. After the patient exits the system[2] the paper record of the patient's visit must be reviewed by both a doctor and a nurse, processed by the shift leader of the medical technicians, and processed by the administrative assistant. These are then added to each patient's permanent medical record, kept at their regular clinic. Every task in the previous four paragraphs is modeled in the simulation, and all times required for each task are based upon derived statistical distributions generated from observed phenomena.

The phone system also presents a challenge. Most phone calls arrive at the desk of the shift leader (SL), who is the ranking medical technician. Some additional calls will arrive at the desk of the administrative assistant. The shift leader will process just over 70% of all phone calls directly, the other 30% being routed to other assets in the ER, such as doctors, nurses, or other MTs. By analyzing the rate at which phone calls arrive at the SL's station, and creating a Poisson distribution, the simulation is then able to estimate,

---

[2] From here on, any patient admitted to in patient wards, discharged, or leaving without being seen will be referred to as having 'exited the system'.

based on the number of active phone lines, the number of busy signals callers into the ER are likely getting. While there is no direct way to confirm these numbers, and they should therefore not be used in any scientific way, they conform to the anecdotal evidence of known assets trying to call. Additionally, these results can be used as a general guide towards the reduction of busy signals based on phone line use.

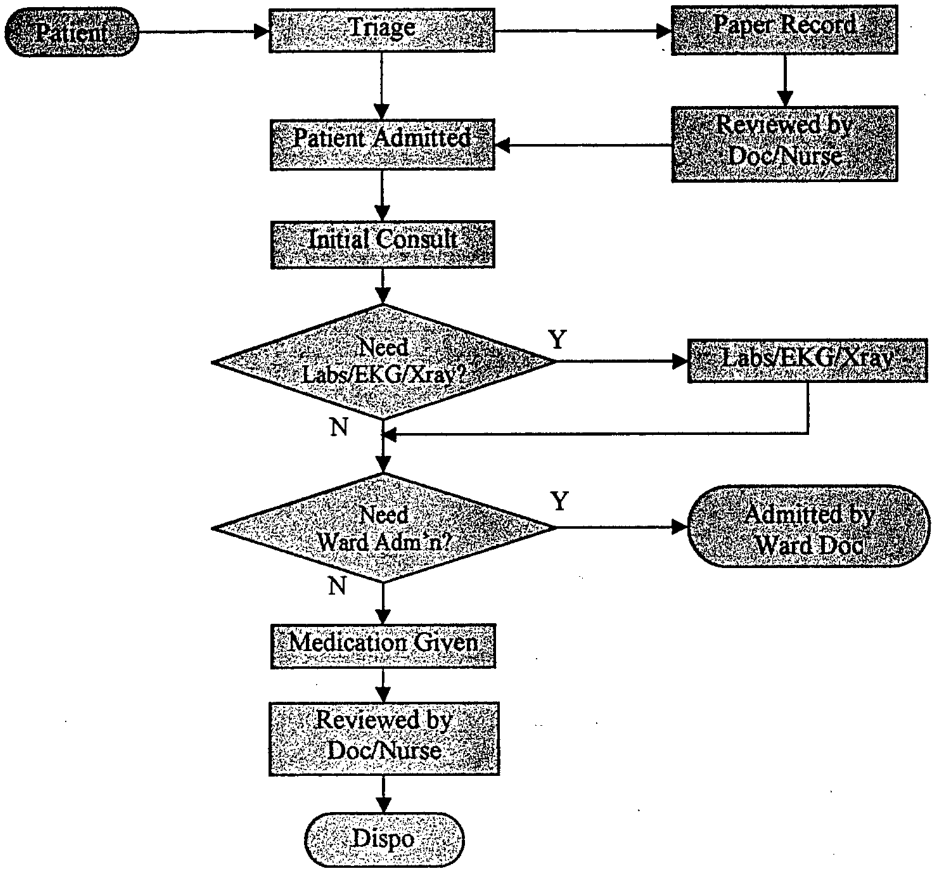


Figure 2: Flow chart of Patient activity

The output measures used to verify the capabilities of the simulation were: the patient's time in the system, and the time patients spent waiting for service from resources, as opposed to time spent in direct contact with resources. The current state of the ER is that patients spend roughly two hours and ten minutes in the ER from triage to discharge (or in-patient admission). The simulated results for current staffing levels[3] matched the observed results within 3 to 5 percent, depending on the random number stream used to generate the stochastics.

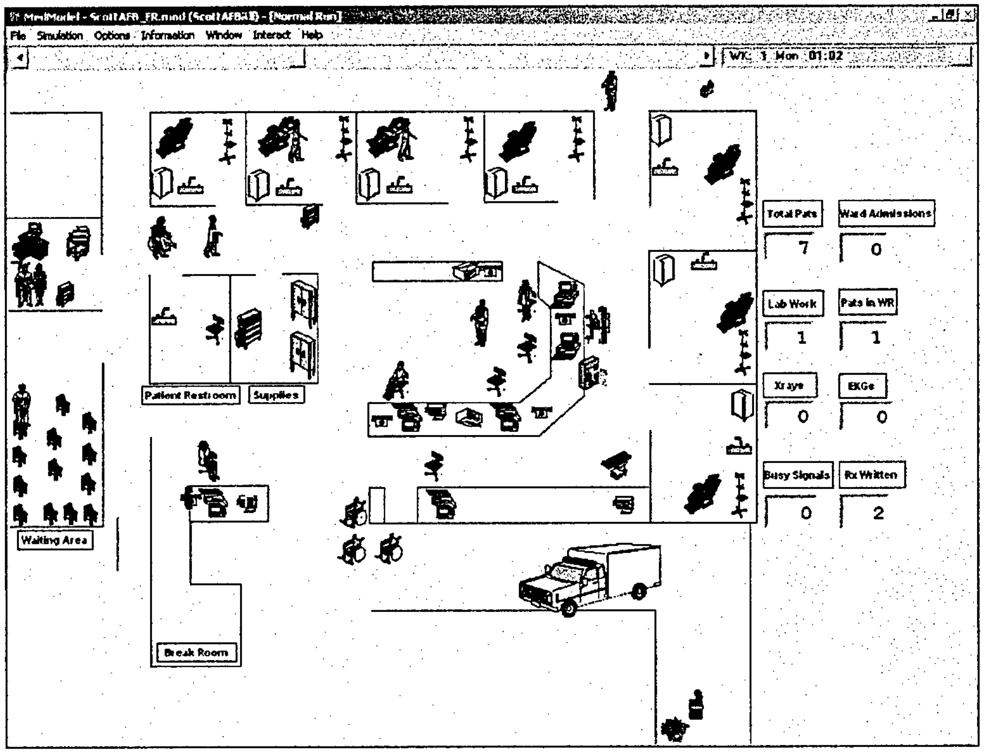
From these results, with the collaboration of emergency room staff and hospital administration, it was determined that the simulation's mimicry of the physical ER was excellent, and that the predictive abilities of the simulation to alterations in the system would be valuable. The simulation went through three periods of revision between July and October 2003, before it was concluded to be accurate. At that point, several scenarios were proposed, some by the research team and some by hospital staff, to examine scenarios involving staff additions or replacements to determine the effect they would have on ER functions. Figure 3 shows the simulation screen, which is a graphical recreation of the layout of the 375th Medical Group Emergency Department.

Complement	Patient time in ER	Patient time waiting	# exceeding 3 hours in ER
1 Doc, 1 Nurse, 3 MT, 1 AdmitDoc	127.70	61.98	272
1 Doc, 1 Nurse, 4 MT, 1 AdmitDoc	102.80	42.18	137
2 Doc, 1 Nurse, 4 MT, 2 AdmitDoc	75.01	17.25	36
1 Doc, 2 Nurse, 3 MT, 1 AdmitDoc	91.88	31.33	88
1 Doc, 2 Nurse, 4 MT, 1 AdmitDoc	83.56	25.65	101
1 Doc, 1 Nurse, 4 MT, 2 AdmitDoc	94.24	35.45	141
1 Doc, 2 Nurse, 4 MT, 2 AdmitDoc	78.76	21.10	77
3 Doc, 3 Nurse, 6 MT, 2 AdmitDoc	63.04	7.43	20

Table 1: Scenario Results

---

[3] Generally one doctor, one nurse, and three or four MTs, one of whom is the shift leader. An additional triage nurse and one doctor are available to admit patients to the in-patient wards. An administrative assistant handles paperwork and disposition of patient records.



Total Pats 7
Ward Admissions 0
Lab Work 1
Pats in WR 1
Xrays 0
EKGs 0
Busy Signals 0
Rx Written 2

Figure 3: Simulation Screen of 375th Medical Group Emergency Room.

V. Results of Scenarios

The primary focus of the initial scenarios simulated was to determine the benefits achieved by adding or subtracting various assets from the ER. The personnel currently working in the ER were of unanimous opinion that an additional nurse was both required and would provide the greatest benefit to the department. Because the hospital is not profit based, the principal output measures used to determine success or failure or a proposed scenario were the amount of time a patient spent in the emergency room, and of that time, how much was spent waiting for services as opposed to being treated. Results of the various scenarios can be seen in the table below. As can be seen, consensus opinion regarding the value of an additional nurse was not wrong with regard to the improvement of the patients' waiting time. However, nearly as dramatic results are achieved through adding a single medical technician, especially if an additional doctor is available to perform patient admissions. Because only one doctor from the inpatient wards is available at any one time, patients needing admission frequently spend a long period of time occupying a room in the ER. As a result, fewer patients can be seen.

Expediting these patients is discussed in the next section. It should be noted that the final line of table 1 was constructed for the purpose of finding a functional minimum for time patients spend in the ER. It is not proposed as a reasonable solution cost-wise, but merely to determine a benchmark against which reasonable solutions can be measured. It is known (by whom, and how-ed.) that it is functionally impossible to reduce patient waiting time under normal patient influx below one hour.

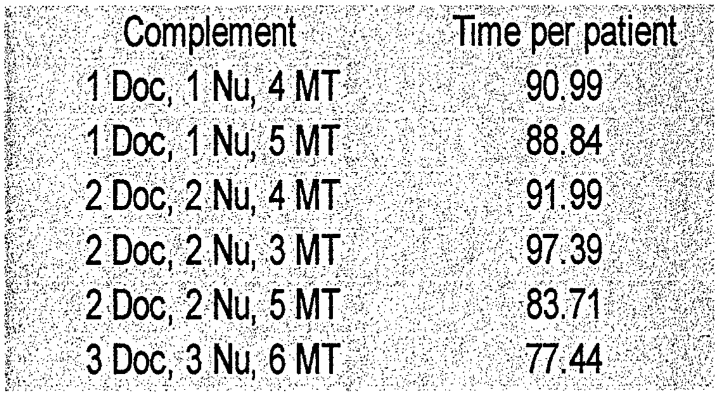
What the above analysis reveals is in some ways fairly obvious. Both time for patients in the system, and the number of failures, can be reduced by adding staff. However, a measure exists now for how good a solution is, not merely in its improvement over the current situation, but also in its deviation from what we now know is essentially the theoretical minimum time possible. This acts, essentially, as a 'duality gap'. When solving a linear program, distance from optimality can be determined by the difference between a particular solution and the corresponding solution to the dual problem. In the simulation, we can use the gap between the current real world situation and the ideal scenario to evaluate the diminishing returns of additional assets.

VI. Mass Casualty Simulation

In addition to examining the ordinary, day to day operations of the ER, it was deemed valuable to use the simulation to study the possibilities of a medium to large mass casualty situation. Because the Scott Air Force Base ER is relatively small, particularly in comparison to other nearby hospitals such as Barnes Jewish (BJC) hospital in downtown St. Louis, it was imagined that 32 patients would be a mass casualty scenario appropriate to the circumstances. This would be comparable to a bus or light rail accident. Considering the proximity of the base to the Metro-Link and to I-64, both are real life possibilities. Additionally, the number of casualties simulated is in proportion with a small scale terror attack such as a suicide bomber. It is entirely plausible that victims of an accident of this size might be treated, or at least stabilized for transport, at the Air Force Base,. This might be far better for their condition than to be transported directly to BJC, which is roughly thirty minutes away by car under good conditions.

Unfortunately, one can not count on good traffic conditions as there is frequently severe congestion on the bridges from Illinois to Missouri.

What both the hospital staff and the experimenter were most interested in examining was how rapidly the entire set of casualties can be treated. There are several assumptions that must be made when dealing with a large number of patients. First, it is assumed that patients retain their probabilities of being lightly, moderately or severely injured. Second, it is assumed that phone calls, computer work, and other ordinary activities are suspended during the extreme situation. Paperwork for the patients is rudimentary, but not abandoned. Third, it is presumed that doctors will require the assistance of either a nurse or an MT at all times while performing procedures on a patient. Lastly, during the crisis, one medical technician acts as an additional triage agent, so that severe cases can be processed as rapidly as possible.



Complement	Time per patient
1 Doc, 1 Nu, 4 MT	90.99
1 Doc, 1 Nu, 5 MT	88.84
2 Doc, 2 Nu, 4 MT	91.99
2 Doc, 2 Nu, 3 MT	97.39
2 Doc, 2 Nu, 5 MT	83.71
3 Doc, 3 Nu, 6 MT	77.44

Table 2: Patient Times for Mass Casualty Scenarios

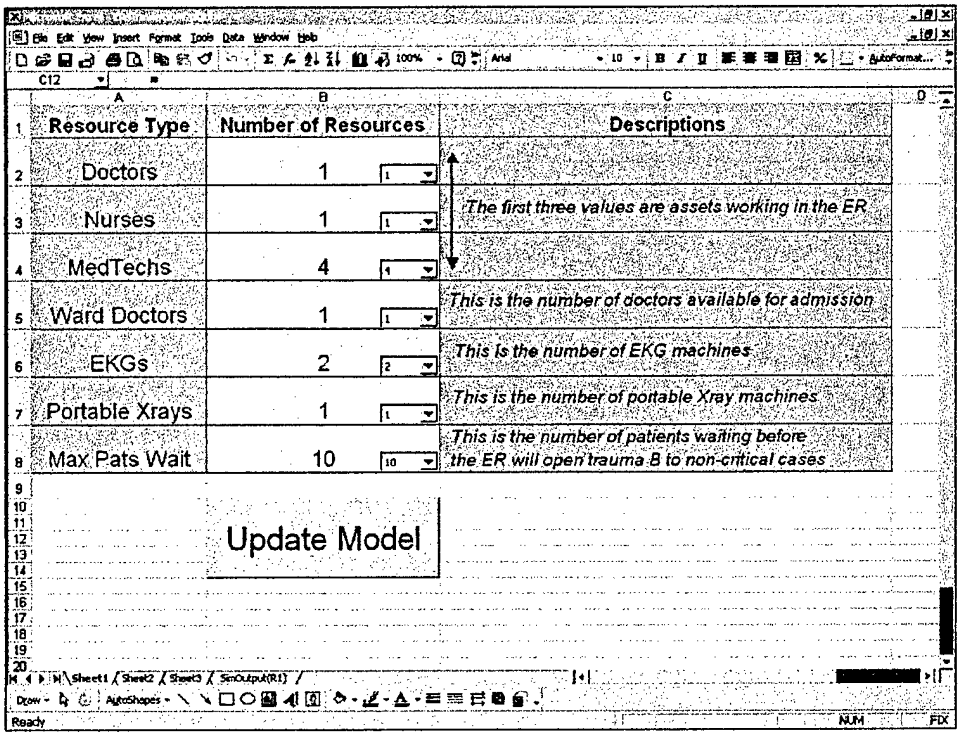
Clearly one fundamental question to be answered is can the Emergency Room use the assets normally in the hospital but assigned to other duties. When the 375th Medical Administrator was consulted on this point, he indicated that this was reasonable. Therefore, we can consider several scenarios with varying levels of staff. What we find is interesting from both a medical and systems engineering point of view. The entire system is generally resistant to significant changes. The time per patient is somewhat inelastic, as the difference between the ordinary staff levels and the scenario with what is considered to be maximum available assets is only 14.8%. This is due to several factors.

First, there are only seven rooms available, thus many patients spend a great deal of time in the waiting room while the ER rooms are filled. We make the assumption that no patient can be treated in the hall, but must be admitted to a room in order to receive medical attention.

The second item we discover is that the addition of doctors and nurses is actually detrimental to patient time, although not greatly so. The reason for this is that doctors and nurses are forced to wait for MTs to continue their work. Without a large enough complement of medical technicians, patients get bogged down in waiting for their doctor or nurse to get access to an MT. Fundamental insights learned from this exercise are that the 375th Emergency Department is equipped to assist in a mass casualty situation, so long as a sufficient technician to provider ratio is maintained.

VII. Creating Individualized Scenarios

In order for the simulation to be truly useful for persons other than the developers, it must be possible to run and interpret myriad scenarios in an efficient and straightforward manner. To this end, an Excel spreadsheet was created so that new scenarios can be easily created and simulated. As can be seen in Figure 4, a spreadsheet titled 'ERSimulationOptions.xls' is included with the model. Using the drop down boxes, the number of each type of asset and tool can be selected. When the desired numbers have been entered, clicking the 'Update Model' button will send the correct parameters to the model, and spawn the model, which can then be run from the MedModel toolbar. The standard run is for a five week long simulation, but this can be changed under the simulation options in MedModel.



Resource Type	Number of Resources	Descriptions
Doctors	1	The first three values are assets working in the ER
Nurses	1	
MedTechs	4	
Ward Doctors	1	This is the number of doctors available for admission
EKGs	2	This is the number of EKG machines
Portable Xrays	1	This is the number of portable Xray machines
Max Pats Wait	10	This is the number of patients waiting before the ER will open trauma B to non-critical cases

Figure 4:Excel Interface for Simulation.

Once the model has been run, MedModel produces an output screen containing information about the entire simulation. Additionally, the SimOutput sheet in ERSimulationOptions.xls contains detailed information for each individual patient seen during the simulation, including the time required for service, and the number of services required by each patient.

VIII. Implementing Optimal Techniques within the Simulation

The next fundamental question to approach with the simulation is: can improvements be made to the workings of the ER without expenditure on additional staff or other resources (X-ray machine, etc.). In examining this prospect, consultations with hospital administration indicated that the primary goal of such attempts should be to reduce patient encounter failures. That is, our optimal goal is to minimize the number of patients spending at least three hours in the Emergency Room. A secondary goal is to reduce overall waiting time, but this is subordinate to minimizing outliers. The reason for this is that frequently patients who would not otherwise require in-patient admission are in fact admitted in order to free a bed in the ER, which has limited space. The in-patient wards are much larger, but it is nevertheless wasteful to fill those beds with persons not requiring overnight observation or longer treatments.

In order to reduce the number of outliers, one must first understand the causes of long ER visits. To effect this, the simulation captured each individual patient's time in the system, together with a Boolean vector containing information regarding each treatment required by the patient. These columns were correlated for a five week run of the simulation. The results, in table 3, once again seem to match intuition; patients needing admission, doctor's procedures and x-ray are most highly correlated with time in the clinic.

	Time
Time	1
Severity	0.135127
LabWork	0.043511
Xray	0.166042
EKG	0.062777
TechProc	0.135235
NurseProc	0.086359
DocProc	0.245805
Admit	0.599439
LWOBS	-0.21223

Table 3: Correlation of Patient Requirements to Time in ER Prior to Optimization.

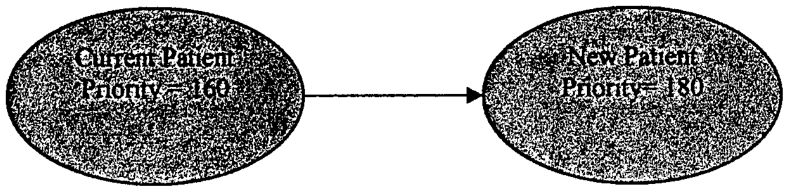
Knowing the co-relations, it is possible to address the problem without necessarily adding staff. The proposal is to reprioritize patients according to the procedures they require, while nevertheless not neglecting patients who are terribly severe and need urgent attention. For example, a patient with a deep but clean cut will require service immediately, even though they are not likely to need admission or any procedures more sophisticated than sutures and a tetanus shot. A patient that is non-emergent but will have to be admitted, say an elderly patient with shortness of breath but a good blood oxygen level, is frequently put aside under current procedure, as the condition is not immediately threatening. However, such a patient will generally require many tests and x-ray and EKG before they can be admitted. Realizing this, starting on such a patient quickly, so long as truly urgent patients are not neglected, can dramatically improve system performance.

The proposal is to prioritize the patients according to a function of their severity, and the number of procedural requirements they have. The function is simple, with a weight applied to make it more meaningful to the MedModel preemption process.

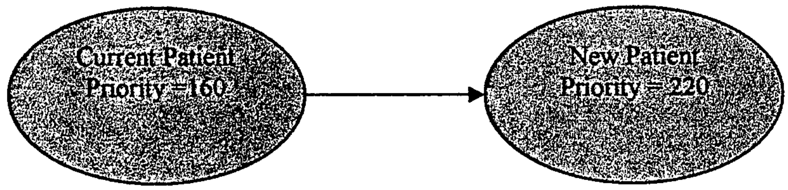
$$\text{Priority} = 20*(\text{Sev}+\text{Lab}+\text{EKG}+\text{Xray}+\text{TechProc}+\text{NurseProc}+2*\text{DocProc}+3*\text{Admit})$$

'Sev' is the severity of the patient from 1-10. All other values are either one or zero depending on whether the patient requires that service. The weights given to doctor's procedures and necessity of admission are to encourage added importance to patients requiring these time consuming services. The initial weight of 20 is used to spread the values assigned to patient priority over a wide range, but not so wide as to cause undue preemption.

The simulation uses priorities as follows. Resources search for patients according to their assigned priority as defined above. If more than one patient is waiting the resource attends the one with the highest priority. If a new patient arrives during the consultation with a higher priority, the resource will only preempt the current patient if the new patient's priority has crossed the threshold into a new multiple of 100. Shown Below:



No Preemption



Preempt Patient

Figure 5: Example of Preemption Process

This method of preemption ensures that patients of lower priority are not arbitrarily left alone for long periods of time, as a patient must be significantly more urgent to require preemption. Thus, patients are given a priority between 20 and 400. Realistically, any patient with a priority of 400 would presumably die, however no incidents of patient death were observed in the ER, and the simulation does not specifically include them.

The results of making these changes are dramatic. Without any increase in resources, the incidence of patient encounter failure fell from 12.9% to 2.86%. Additionally, the average number of patients in the waiting room declined correspondingly. The data presented is for a five week run of the simulation, each with identical resources[4]. The new co-relations between time in clinic and the various requirements of the patient are as seen below in table 4.

---

[4] One doctor, one nurse, three MT's, one wards doctor, and a shiftleader. The standard complement observed in the ER as currently staffed.

	Time
Time	1
Severity	0.129713
LabWork	0.083801
Xray	0.317152
EKG	0.101581
TechProc	0.110994
NurseProc	0.153703
DocProc	0.133635
Admit	0.47567
LWOBS	-0.32171

Table 4: Co-relations with Priority Implementation

As can be seen, the co-relations with time of procedures, severity, lab work, and EKG are smooth between 0.08 and 0.15, indicating that these all contribute fairly equally to patient time in clinic. The highest co-relations now are, as before, X-ray and Admit. The X-ray correlation has grown, as is understandable, given that the time required to leave the clinic to be seen in radiology and return,, often in a wheelchair, cannot be diminished much by prioritizing it. The admission correlation, though, while still high again due to its nature of requiring services from elsewhere in the hospital and extensive consultations with other assets, has been decreased from 0.599 to 0.476. This indicates that admitted patients are no longer taking up so much of the time of the ER resources, despite the fact that they occur at precisely the same rate per patient as previously.

This approach has several significant advantages. The first is that it does not require the introduction of any new personnel or equipment. Therefore, it does not increase the cost of running the emergency room. Second, it greatly reduces the number of patient encounter failures, and reorganizes the time spent in the ER into more meaningful interactions with the assets, rather than waiting for services to begin. And finally, the prioritizing method used is simple to implement. It would require little more than a doctor, after his initial assessment, making a single calculation and then informing the rest of the ER staff of the priority of each patient. This would necessarily include an instruction to interrupt other patients during non-vital procedures to attend to a higher priority patient. Some diplomacy would certainly be required here, in order to ensure certain lower priority patients don't become offended, but could be handled with a nurse or an MT simply saying 'I need you for a moment, doctor.'

IX. Continuing Work and Conclusions to the 375th MedGroup Problem

Extensions to this study include the reaction of an ER to a chemical or biological event. Currently, the 375th ER is constructing a decontamination chamber. Simulation of such an event would be straightforward, but would require additional data and knowledge of system flow. The donning of chemical suits, and the decreased mobility affiliated with them would have to be modeled. One could properly assume that such an event would coincide with a mass casualty situation, at least in the case of a chemical attack. Modeling mortality rates and alternative procedures used to treat such victims would require access to providers with training or experience in such matters due to the fact that observing a statistically significant number of events to model from real life would be impossible and undesirable.

Additionally, the inclusion of progressive refinement within the simulation of the coefficients for the semantic output will add a significant improvement to the novelty of this project. The simulation will be used as an optimal tool in and of itself. Instead of requiring multiple runs of a simulation with human interaction to find better values by trial and error, a single run of the simulation will find optimal values of the coefficients based on the stochastics inherent in the system. Therefore, the results of that particular simulation run would be best interpreted not as accurate based upon patient times and results, but as an engine for determining coefficients to assign semantic states.

The ER simulation is a sophisticated but relatively straightforward tool for examining the workings of the emergency department of the 375th Medical Group at Scott Air Force Base. A simulative approach to hybrid systems such as an emergency department, which contains many different elements and types of both patients and resources, has been shown in the past to provide interesting insights. The current approach, that in which ER resources are able, dynamically within the simulation, to organize patients into an intelligent order, updating each priority as each patient arrives, provides for improvement over the current actual ER performance. The inclusion of dynamic optimal techniques, based on the known co-relations of patient activity to time in clinic, shows remarkable promise.

X. Simulation of the Missouri Baptist Emergency Department

Much of the description of the simulation of the Missouri Baptist Emergency Department would be redundant given its similarity to the preceding information. Additionally, the simulation of the Missouri Baptist (MoBap) facility is still a work in progress. Begun in January of 2004, an identical observation and data acquisition process was implemented in the new facility. The MoBap ER is significantly larger than the one of the 375th MedGroup. There are fourteen exam rooms, one of which is useable only for ophthalmology and otolaryngology. Additionally, there are seven hall beds available for use if all exam rooms are full. In the event that there is an extreme overload in request for services, there is an additional holding room for non-critical cases which can hold four beds. Generally this area will not be used unless the hospital has already gone to 'divert status', meaning that it will no longer accept ambulances. It is a critical situation that happened only once in more than three months of observation.

The proper use of hall beds presents a delicate problem. The reason for this is that there is no established method for their employment in the actual hospital. It is idiosyncratic from doctor to doctor. Some doctors choose not to fill the hall beds until all treatment rooms are full. Other doctors will wait even longer, until there is a backlog in the waiting room. However, some doctors *prefer* to use hall beds even when one or perhaps many treatment rooms are available, Despite the fact that patients prefer the privacy of rooms. The reasons they give are that the patients put in the hall beds are non-critical and generally do not need to disrobe (although there are screens available in the event that an EKG needs to be done. Additionally, using a hall bed early allows for the arrival of a patient who cannot use the hall beds.

Although they are a relatively small percentage[5], certain patients cannot be put into hall beds. These include patients with intestinal conditions such as diarrhea, patients who disrupt others due to psychiatric problems or inebriation, female patients with gynecological complaints, or patients requiring chest X-rays. For safety reasons, X-rays cannot be done in the hall. This provides an additional complication for the constructor on the model as occasionally a patient requiring a portable X-ray will be seen in the hall,, Then he might be moved briefly into a vacated treatment

---

[5] Perhaps 15-25% depending on the situation in the ER, as some patients are 'swing patients' that can be put in the hall if necessitated by extreme conditions, but will not be put in hall beds if there is a treatment room available.

room just long enough to have the Portable X-ray employed to aid in his diagnosis. Rooms are vacated briefly when patients must go to the Radiology Department for studies that can not be done with the portable equipment. Portable X-ray radiology generally required less than five minutes to complete. Then the patient would be returned to a hall bed.

Therefore, given the idiosyncratic nature of the doctors' decision making, and the intractable nature of hall beds, we attempt to produce a simulation which accurately mimics ordinary behavior and produces identical output. When the total patient time in clinic, for example, is within a very small range of the actual system over long simulated runs[6], perhaps between 98% and 102%, and is repeatable, we can say that we have captured the system accurately enough to make conclusions. We can then implement strategies that have desirable results within the simulation.

Currently, the simulation of the MoBap ER is in its final stages. The patient flow, phone system, treatment rooms and record keeping systems have all been built. Remaining is the proper function of hall beds and the overflow holding room. It is believed that results not dissimilar from the successes of the previous simulation will be achieved, and may, due to the larger system and financial drive of a private sector institution, be more valuable.

XI. Scheduling of Residents at Duke University Hospitals

The final project of the last semester has been a study of scheduling the resident workload at a group of hospitals associated with the Duke University Medical System. The problem can be generally described as the desire to satisfy all work requirements, given a pool of residents of varying skill levels, and concurrently given the constraint that no resident may average more than 80 hours of work each week. Any shortfall in available resident work hours must be filled by fellows, that is, full physicians. The specific problem description is given below:

Statement of the Problem:

---

[6] Simulations of this sort are sometimes not accurate for small periods of time such as a few hours or a day, as their reliance on large numbers of stochastic processes can sometimes produce outliers which disrupt the system for a short period of time. This happens in the actual system as well, of course, but it is difficult to draw conclusions from such instances. A 'long run' of the system is considered to be roughly twelve weeks. This allows the Strong Law of Large Numbers to take effect, and for outliers on both sides of the probability curves to average out.

1. Residents are divided into 5 clinical years, labeled R1 through R5. R5s are the most senior.
2. There are a total of 25 R1, 10 R2, 7 R3, 7 R4, 7 R5 and 1 R6 residents.
3. The residents are allowed to work a total of 80 hours per week averaged over 4 weeks.
4. Residents must have off 1 complete 24 hour day out of 7 days, averaged over 4 weeks.
5. There must be at least a 10 hour period of rest between duty periods.
6. Residents cannot work more than 24 hours straight, although an additional 6 hours of time may be used for learning or outpatient clinic activities.
7. The resident work day begins at 6:00 AM

The residents rotate through 4 hospitals, Duke, DRH, VA1 and VA2.

At Duke, the primary services (n=5) are: GI, Vascular, Surg Onc, Trauma, and Transplant. On some services, there are additional caregivers, Fellows, who can cover the residents' uncovered hours. These Fellows are not subject to work hour restrictions. Our first priority is to cover the services at Duke.

1. GI, Vascular, Surg Onc, and Trauma must have at least 1 R5, 1 R3 or R4, and 1 R1 or R2.
2. Vascular has an R6 resident. GI and Surg Onc each have a Fellow.
3. Transplant must have an R3 and R2. Transplant has a Fellow.
4. VA1 must have 1 R5, 1 R3 or R4, and 1 R1 or R2.
5. VA2 must have 1R5, 1 R3 or R4, and 2 R1 or R2s (or a combination thereof)
6. DRH must have 1 R3 or R4, and 1 R1 or R2.

From this point onward, the manpower requirements at Duke are listed. The other hospitals will configure their own schedule, once the residents are assigned.

7. During the hours of 6AM to 6PM, Mon through Fri, each service at Duke must have a R4, R5, R6 or Fellow on duty.
8. During the hours of 6AM to 6PM, Mon through Fri, each service at Duke must have a R1, R2 or R3 on duty.
9. During the hours of 6 PM to 6AM, Mon through Fri, all of Duke will have a single R4, R3 or R5 on duty.
10. During the hours of 6 PM to 6AM, Mon through Fri, all of Duke will have 2 or 3 R1s or R2s on duty.
11. On Sat and Sun, from 6AM to 10AM, each service at Duke must have an R4, R3 or R5 on duty.
12. On Sat and Sun, from 6AM to 10AM, each service at Duke must have an R1 or R2 on duty.
13. On Sat and Sun, from 10AM to 6AM, all of Duke must have 2 or 3 R1s or R2s on duty.
14. At any one time, there are 2 R2s assigned to Critical Care and they rotate 24 hours on, followed by 24 hours off.
15. Each resident changes services once per month.
16. Each resident is allowed 4 weeks of vacation per year. This can be taken as a lump or spread through out the year in week-long blocks.

The questions are:

1. Can these manpower needs be accommodated within the total number residents available and the 80 hour work week requirements? If so, what is the minimum number of R1 through R5 residents required? What if we also wish to minimize the Fellows' time to 80 hours per week.
2. If there are excess residents, how many are there and what are the total number of excess hours available? We would like to know this so we can rotate residents through elective courses.
3. Can this solution be configured in a general format since the numbers of residents change from year-to-year? The manpower requirements do not.

Formulation of the Model

The model was designed to satisfy all constraints, and was initially designed as a linear program. However, the linear program was determined to be insufficient for certain, nonlinear aspects of the model requirements. The constraint that all residents must have 10 hours off after a shift of 24 hours, for example, required an if-then type constraint, which is nonlinear. It was therefore decided to take a constraint satisfaction approach to the problem, employing the software suite ILOG, which is designed to solve problems of this kind, with both linear and nonlinear constraints.

The complete formulation is presented below, in the Optimization Programming Language used by ILOG for solving nonlinear mixed integer programming problems (NLMIP).

Mathematical Model for Duke Hospital System:

```
int NumRes=58;
int NumShifts=168;
int NumServ=11;

range Shift 1..NumShifts;
range Resident 1..NumRes;
range Serv 1..NumServ;


var int X[Resident, Serv, Shift] in 0..1;
var int Z[Resident, Serv] in 0..1;

maximize (sum(i in Resident : i<58, j in Serv : j<>11, k in Shift) X[i,j,k])

subject to
{

//Housekeeping
forall(i in Resident){forall(s in Shift){sum(j in 1..10) (X[i,j,s]+X[i,11,s])=1}};
forall(i in Resident){forall(j in 1..10){forall(s in 2..167) X[i,j,s]=1 => (X[i,j,s-1]=1 ∨
X[i,j,s+1]=1)}};
forall(i in Resident){forall(s in 2..167) X[i,11,s]=0 => (X[i,11,s-1]=0 ∨ X[i,11,s+1]=0)};

//alpha
forall(i in 1..56){sum(j in Serv) Z[i,j]<=1};
```

```
//beta
forall(i in 1..56){forall(j in 1..10){sum(k in Shift) X[i,j,k]-80*Z[i,j]<=0}};

//delta
forall(i in Resident){forall(k in Shift){sum(j in Serv) X[i,j,k]<=1}};

//epsilon
forall(j in 1..10){sum(i in 1..60, k in Shift) X[i,j,k]>=1};

//A
forall(i in Resident){forall(k in 1..165){ 10*X[i,11,k]-
10*X[i,11,k+1]+5*X[i,11,k+2]+5*X[i,11,k+3]>=0}};

//B
X[i,11,k]+X[i,11,k+1]+X[i,11,k+2]+X[i,11,k+3]+X[i,11,k+4]+X[i,11,k+5]+X[i,11,k+6]+X[i,11,
k+7]>=0}};
forall(i in 1..57){forall(k in 1..161){sum(n in 1..7) X[i,11,k+n]>=1}};


//C
forall(j in 1..4){sum(i in 1..35) Z[i,j]>=1};
forall(j in 1..4){sum(i in 36..56) Z[i,j]>=1};

//D
sum(i in 26..35) Z[i,5]>=1;
sum(i in 36..42) Z[i,5]>=1;

//E
sum(i in 1..35) Z[i,6]>=1;
sum(i in 36..49) Z[i,6]>=1;
sum(i in 50..56) Z[i,6]>=1;

//F
sum(i in 1..35) Z[i,7]>=2;
sum(i in 36..49) Z[i,7]>=1;
sum(i in 50..56) Z[i,7]>=1;


//G
sum(i in 1..35) Z[i,8]>=1;
sum(i in 36..49) Z[i,8]>=1;

//G*
sum(i in 26..35) Z[i,9]=2;


//H
forall(j in 1..5){forall(k in 1..3){sum(i in 43..56) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 7..9){sum(i in 43..56) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 13..15){sum(i in 43..56) X[i,j,k]>=1}};
```

```
forall(j in 1..5){forall(k in 19..21){sum(i in 43..56) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 25..27){sum(i in 43..56) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 31..33){sum(i in 43..56) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 37..39){sum(i in 43..56) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 43..45){sum(i in 43..56) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 49..51){sum(i in 43..56) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 55..57){sum(i in 43..56) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 61..63){sum(i in 43..56) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 67..69){sum(i in 43..56) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 73..75){sum(i in 43..56) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 79..81){sum(i in 43..56) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 85..87){sum(i in 43..56) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 91..93){sum(i in 43..56) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 97..99){sum(i in 43..56) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 103..105){sum(i in 43..56) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 109..111){sum(i in 43..56) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 115..117){sum(i in 43..56) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 121..123){sum(i in 43..56) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 127..129){sum(i in 43..56) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 133..135){sum(i in 43..56) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 139..141){sum(i in 43..56) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 145..147){sum(i in 43..56) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 151..153){sum(i in 43..56) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 157..159){sum(i in 43..56) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 163..165){sum(i in 43..56) X[i,j,k]>=1}};


//I
forall(j in 1..5){forall(k in 1..3){sum(i in 1..42) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 7..9){sum(i in 1..42) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 13..15){sum(i in 1..42) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 19..21){sum(i in 1..42) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 25..27){sum(i in 1..42) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 31..33){sum(i in 1..42) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 37..39){sum(i in 1..42) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 43..45){sum(i in 1..42) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 49..51){sum(i in 1..42) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 55..57){sum(i in 1..42) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 61..63){sum(i in 1..42) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 67..69){sum(i in 1..42) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 73..75){sum(i in 1..42) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 79..81){sum(i in 1..42) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 85..87){sum(i in 1..42) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 91..93){sum(i in 1..42) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 97..99){sum(i in 1..42) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 103..105){sum(i in 1..42) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 109..111){sum(i in 1..42) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 115..117){sum(i in 1..42) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 121..123){sum(i in 1..42) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 127..129){sum(i in 1..42) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 133..135){sum(i in 1..42) X[i,j,k]>=1}};
```

```
forall(j in 1..5){forall(k in 139..141){sum(i in 1..42) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 145..147){sum(i in 1..42) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 151..153){sum(i in 1..42) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 157..159){sum(i in 1..42) X[i,j,k]>=1}};
forall(j in 1..5){forall(k in 163..165){sum(i in 1..42) X[i,j,k]>=1}};

//J
forall(k in 4..6){sum(i in 36..56) X[i,10,k]>=1};
forall(k in 10..12){sum(i in 36..56) X[i,10,k]>=1};
forall(k in 16..18){sum(i in 36..56) X[i,10,k]>=1};
forall(k in 22..24){sum(i in 36..56) X[i,10,k]>=1};
forall(k in 28..30){sum(i in 36..56) X[i,10,k]>=1};
forall(k in 34..36){sum(i in 36..56) X[i,10,k]>=1};
forall(k in 40..42){sum(i in 36..56) X[i,10,k]>=1};
forall(k in 46..48){sum(i in 36..56) X[i,10,k]>=1};
forall(k in 52..54){sum(i in 36..56) X[i,10,k]>=1};
forall(k in 58..60){sum(i in 36..56) X[i,10,k]>=1};
forall(k in 64..66){sum(i in 36..56) X[i,10,k]>=1};
forall(k in 70..72){sum(i in 36..56) X[i,10,k]>=1};
forall(k in 76..78){sum(i in 36..56) X[i,10,k]>=1};
forall(k in 82..84){sum(i in 36..56) X[i,10,k]>=1};
forall(k in 88..90){sum(i in 36..56) X[i,10,k]>=1};
forall(k in 94..96){sum(i in 36..56) X[i,10,k]>=1};
forall(k in 100..102){sum(i in 36..56) X[i,10,k]>=1};
forall(k in 106..108){sum(i in 36..56) X[i,10,k]>=1};
forall(k in 112..114){sum(i in 36..56) X[i,10,k]>=1};
forall(k in 118..120){sum(i in 36..56) X[i,10,k]>=1};
forall(k in 124..126){sum(i in 36..56) X[i,10,k]>=1};
forall(k in 130..132){sum(i in 36..56) X[i,10,k]>=1};
forall(k in 136..138){sum(i in 36..56) X[i,10,k]>=1};
forall(k in 142..144){sum(i in 36..56) X[i,10,k]>=1};
forall(k in 148..150){sum(i in 36..56) X[i,10,k]>=1};
forall(k in 154..156){sum(i in 36..56) X[i,10,k]>=1};
forall(k in 160..162){sum(i in 36..56) X[i,10,k]>=1};
forall(k in 166..168){sum(i in 36..56) X[i,10,k]>=1};


//K
forall(k in 4..6){sum(i in 1..35) X[i,10,k]>=2};
forall(k in 10..12){sum(i in 1..35) X[i,10,k]>=2};
forall(k in 16..18){sum(i in 1..35) X[i,10,k]>=2};
forall(k in 22..24){sum(i in 1..35) X[i,10,k]>=2};
forall(k in 28..30){sum(i in 1..35) X[i,10,k]>=2};
forall(k in 34..36){sum(i in 1..35) X[i,10,k]>=2};
forall(k in 40..42){sum(i in 1..35) X[i,10,k]>=2};
forall(k in 46..48){sum(i in 1..35) X[i,10,k]>=2};
forall(k in 52..54){sum(i in 1..35) X[i,10,k]>=2};
forall(k in 58..60){sum(i in 1..35) X[i,10,k]>=2};
forall(k in 64..66){sum(i in 1..35) X[i,10,k]>=2};
forall(k in 70..72){sum(i in 1..35) X[i,10,k]>=2};
forall(k in 76..78){sum(i in 1..35) X[i,10,k]>=2};
```

```
forall(k in 82..84){sum(i in 1..35) X[i,10,k]>=2};
forall(k in 88..90){sum(i in 1..35) X[i,10,k]>=2};
forall(k in 94..96){sum(i in 1..35) X[i,10,k]>=2};
forall(k in 100..102){sum(i in 1..35) X[i,10,k]>=2};
forall(k in 106..108){sum(i in 1..35) X[i,10,k]>=2};
forall(k in 112..114){sum(i in 1..35) X[i,10,k]>=2};
forall(k in 118..120){sum(i in 1..35) X[i,10,k]>=2};
forall(k in 124..126){sum(i in 1..35) X[i,10,k]>=2};
forall(k in 130..132){sum(i in 1..35) X[i,10,k]>=2};
forall(k in 136..138){sum(i in 1..35) X[i,10,k]>=2};
forall(k in 142..144){sum(i in 1..35) X[i,10,k]>=2};
forall(k in 148..150){sum(i in 1..35) X[i,10,k]>=2};
forall(k in 154..156){sum(i in 1..35) X[i,10,k]>=2};
forall(k in 160..162){sum(i in 1..35) X[i,10,k]>=2};
forall(k in 166..168){sum(i in 1..35) X[i,10,k]>=2};


forall(k in 4..6){sum(i in 1..35) X[i,10,k]<=3};
forall(k in 10..12){sum(i in 1..35) X[i,10,k]<=3};
forall(k in 16..18){sum(i in 1..35) X[i,10,k]<=3};
forall(k in 22..24){sum(i in 1..35) X[i,10,k]<=3};
forall(k in 28..30){sum(i in 1..35) X[i,10,k]<=3};
forall(k in 34..36){sum(i in 1..35) X[i,10,k]<=3};
forall(k in 40..42){sum(i in 1..35) X[i,10,k]<=3};
forall(k in 46..48){sum(i in 1..35) X[i,10,k]<=3};
forall(k in 52..54){sum(i in 1..35) X[i,10,k]<=3};
forall(k in 58..60){sum(i in 1..35) X[i,10,k]<=3};
forall(k in 64..66){sum(i in 1..35) X[i,10,k]<=3};
forall(k in 70..72){sum(i in 1..35) X[i,10,k]<=3};
forall(k in 76..78){sum(i in 1..35) X[i,10,k]<=3};
forall(k in 82..84){sum(i in 1..35) X[i,10,k]<=3};
forall(k in 88..90){sum(i in 1..35) X[i,10,k]<=3};
forall(k in 94..96){sum(i in 1..35) X[i,10,k]<=3};
forall(k in 100..102){sum(i in 1..35) X[i,10,k]<=3};
forall(k in 106..108){sum(i in 1..35) X[i,10,k]<=3};
forall(k in 112..114){sum(i in 1..35) X[i,10,k]<=3};
forall(k in 118..120){sum(i in 1..35) X[i,10,k]<=3};
forall(k in 124..126){sum(i in 1..35) X[i,10,k]<=3};
forall(k in 130..132){sum(i in 1..35) X[i,10,k]<=3};
forall(k in 136..138){sum(i in 1..35) X[i,10,k]<=3};
forall(k in 142..144){sum(i in 1..35) X[i,10,k]<=3};
forall(k in 148..150){sum(i in 1..35) X[i,10,k]<=3};
forall(k in 154..156){sum(i in 1..35) X[i,10,k]<=3};
forall(k in 160..162){sum(i in 1..35) X[i,10,k]<=3};
forall(k in 166..168){sum(i in 1..35) X[i,10,k]<=3};

//L
forall(j in 1..5){sum(i in 36..56) X[i,j,31]>=1};
forall(j in 1..5){sum(i in 36..56) X[i,j,73]>=1};
forall(j in 1..5){sum(i in 36..56) X[i,j,115]>=1};
forall(j in 1..5){sum(i in 36..56) X[i,j,157]>=1};
```

```
forall(j in 1..5){sum(i in 36..56) X[i,j,37]>=1};
forall(j in 1..5){sum(i in 36..56) X[i,j,79]>=1};
forall(j in 1..5){sum(i in 36..56) X[i,j,121]>=1};
forall(j in 1..5){sum(i in 36..56) X[i,j,163]>=1};


//M
forall(j in 1..5){sum(i in 1..35) X[i,j,31]>=1};
forall(j in 1..5){sum(i in 1..35) X[i,j,73]>=1};
forall(j in 1..5){sum(i in 1..35) X[i,j,115]>=1};
forall(j in 1..5){sum(i in 1..35) X[i,j,157]>=1};
forall(j in 1..5){sum(i in 1..35) X[i,j,37]>=1};
forall(j in 1..5){sum(i in 1..35) X[i,j,79]>=1};
forall(j in 1..5){sum(i in 1..35) X[i,j,121]>=1};
forall(j in 1..5){sum(i in 1..35) X[i,j,163]>=1};

//N
forall(k in 32..36){sum(i in 1..35) X[i,10,k]>=2};
forall(k in 32..36){sum(i in 1..35) X[i,10,k]<=3};
forall(k in 38..42){sum(i in 1..35) X[i,10,k]>=2};
forall(k in 38..42){sum(i in 1..35) X[i,10,k]<=3};
forall(k in 74..78){sum(i in 1..35) X[i,10,k]>=2};
forall(k in 74..78){sum(i in 1..35) X[i,10,k]<=3};
forall(k in 80..84){sum(i in 1..35) X[i,10,k]>=2};
forall(k in 80..84){sum(i in 1..35) X[i,10,k]<=3};

forall(k in 116..120){sum(i in 1..35) X[i,10,k]>=2};
forall(k in 116..120){sum(i in 1..35) X[i,10,k]<=3};
forall(k in 122..126){sum(i in 1..35) X[i,10,k]>=2};
forall(k in 122..126){sum(i in 1..35) X[i,10,k]<=3};
forall(k in 158..160){sum(i in 1..35) X[i,10,k]>=2};
forall(k in 158..160){sum(i in 1..35) X[i,10,k]<=3};
forall(k in 162..166){sum(i in 1..35) X[i,10,k]>=2};
forall(k in 162..166){sum(i in 1..35) X[i,10,k]<=3};



};

search{

    generateSize(X);
    generateSize(Z);
};


display (i in Resident, j in Serv : Z[i,j]) Z[i,j];
display (i in Resident, j in Serv, k in 1..68 : X[i,j,k]=1) X[i,j,k];
```

Discussion of Formulation

The lettered constraints correspond directly to the numbered constraints in the problem formulation. The 'housekeeping' constraints ensure that no resident works for more than eighty hours in a week, that any twenty four hour shift is followed by at least twelve hours off[7]. Additionally, these constraints, which include the only nonlinear constraints in the model, require that any shift is at least eight hours long, and any break between shifts is at least eight hours long. These were included due to the fact that previous attempts resulted in several residents being assigned to several consecutive instances of a four hour shift followed by a four hour break. While this technically satisfied the problem statements constraints, it was not deemed feasible.

The above model, while it does in fact capture the problem accurately, has prohibitive run times to solve. Due to the fact that it is a NLMIP, certain portions of the solution space must be searched 'brute force'. In a problem containing tens of thousands of variables and constraints, it became impossible to achieve a solution. Therefore, the next step is to pare down the model to a one week solution rather than a four week solution. This eliminates three quarters of the variables and almost that same percentage of the constraints. Then, to add some constraints so that the first day of the week is dependant not only on the second, but on the last as well. This will enable us to 'stitch together' four one week solutions into a single month long schedule. It is hoped that this diminution of the model will allow for run-times that are acceptable.

The anticipated answers to the problem formulation questions are as follows:

1. Because earlier relaxed version of the model provided excellent but slightly unfeasible results, it is believed that the current manpower available is more than sufficient to cover the requirements. The minimum number of residents required is unknown at this time. Determining that number would require multiple runs of the model while gradually decreasing the number of residents of various types available. It may be that it is possible to remove several residents of a particular level, but none of another.

---

[7] Although the strict requirement is only for ten hours off following a twenty four hour shift, the problem was slightly relaxed to accommodate computational difficulties. This allows the model to function in four hour shifts rather than two hour shifts, halving in the number of variables required. This modification was made with the approval of the Duke University Administrator.

2. All requirements were covered with the average resident working between 64 and 80 hours per week. Therefore there most likely will be extra hours available for elective courses. It is now known whether Fellows' schedules can be accommodated to an eighty hour week as we do not have the data for when they are required and at which facilities, nor how many Fellow are employed at Duke.

3. Providing a year long model would be virtually impossible. While such a model could be easily (but tediously) assembled, it is doubtful that it could be solved using currently available computational hardware. It is recommended that as a one week model could be extended to provide a month long schedule, so perhaps could a month long schedule be extended to 52 weeks. As the number of residents changes over time, new solutions could be obtained using the number of residents available at that time.

XII Conclusions and Continuations

The three projects described above have been the focus of work for the last year, and specifically from January 2004 until present. The 375th Medical Group project is completed with respect to the simulation of the Emergency Department and verification of its validity and capacity to capture the activities in that system. Development of the semantic states for system interaction with medical personnel is ongoing. Additionally, a progressive refinement engine for the coefficients of the priority equation is presently being implemented. It is believed that this will allow the simulation to act itself as an optimizing algorithm, an approach which is to be included in the author's forthcoming doctoral thesis as a novel approach to Interactive Semantic Hybrid Systems. These approaches will be included into the Missouri Baptist Emergency Department Simulation as well. This will comprise the bulk of the dissertation, portions of which are included in this report as descriptive material.

The Duke University Hospital Resident Scheduling Model is a new constraint satisfaction approach to nonlinear mixed integer programming assignment problems. The model, agreed to be accurate and complete, is nevertheless prohibitively complex for ordinary computational solutions, and must therefore be decimated in order to provide valuable results. Previous relaxed versions of the model provided good but slightly infeasible results, and so it is believed that the complete model will function once the appropriate revisions are identified and implemented.