2000

An Optimization Model Based Decision Support System for Staff Scheduling Analysis in Healthcare Facilities

Mark W. Isken
Oakland University, isken@oakland.edu

An Optimization Model Based Decision Support System for Staff Scheduling Analysis in Healthcare Facilities

Mark W. Isken, Decision and Information Sciences, Oakland University, isken@oakland.edu

Abstract

Staff scheduling is an important administrative function in healthcare facilities given the high cost of labor and the often critical nature of the service delivery process. These scheduling problems are notoriously difficult due to a number of factors including stochastic demand which varies by time of day and day of week, flexible scheduling policies such as full and part-time staff, and multiple shift lengths and start times. We describe an optimization based decision support system developed and implemented at a large tertiary care hospital to aid business analysts in solving tactical scheduling analysis problems for a wide range of hospital units including obstetrics, laboratory, transcription, surgery, and appointment scheduling clerks.

The Problem

Managers of many departments in tertiary care hospitals face very difficult resource allocation problems related to staffing and scheduling. Consider a department such as the surgical recovery room. After patients complete surgery they are transported to a surgical recovery area where they recover from the effects of anesthesia. The surgical recovery room is subject to patients arriving from many operating rooms and having undergone a wide range of procedures. Patient arrival volumes vary by time of day and day of week and the severity of the patients' conditions affect the amount of nursing care required. Recovery room nurses typically work a wide range of schedules having staggered start times, different shift lengths, and different number of hours per week in order to meet patient needs. While overstaffing leads to excess labor costs, understaffing can lead to bottlenecks which can cause patients to be "blocked" in the operating room and, more importantly, can compromise patient care. Thus it is important that effort be made to schedule staff so as to best match demand. Many other hospital units such as the laboratory, transcription, patient transport, appointment scheduling, obstetrics, and the emergency department face similar types of staffing challenges.

While there are many difficult facets to these scheduling problems, we focus here on one particular aspect, that of determining optimal or near optimal *tour schedules* for a given demand pattern and different scheduling policy scenarios. For example, in the recovery room case above, let us assume that we have used computer simulation to guide our decision to set half-hourly staffing levels for each day of the week at the 95th (say) percentile of the historical demand during each half-hour period of the week. Now, the problem faced by the manager is to determine good scheduling practices and policies that allow these staffing targets to be achieved at minimum labor cost. More precisely, the tour scheduling problem is to determine daily shift start times and days worked patterns simultaneously that minimize total labor hours (or labor cost) while meeting labor coverage requirements for each planning period, where the planning period is typically half-hourly or hourly. The staff was willing to consider eight, ten, and twelve-hour shifts. Certain shift start times were considered undesirable (those times which would result in a shift ending between 12:30AM and 6:00AM). Typical questions one might need to explore include: What is the effect on the total staff needed if only eight-hour shifts are allowed? What if part-time positions are considered? How much additional staff would be needed if each employee's start times could not change during the week? Would a wider range of allowable shift start times help to reduce labor cost, and if so by how much? These are difficult tactical questions.

A useful distinction can be made between *tactical personnel scheduling* and *operational personnel scheduling*. Operational personnel scheduling is primarily concerned with matching currently available resources to the system's coverage requirements while taking into account factors such as individual employee availability and scheduling preferences. *Tactical personnel scheduling* is the use of models in the short to medium range planning process that capture the most important aspects of the personnel scheduling process. In tactical personnel scheduling, one is not concerned with scheduling specific employees with specific availabilities, but instead with determining the number of employees needed to satisfy coverage requirements and basic personnel scheduling constraints such as allowable shift lengths, start times, and number of days worked per week. The art of this modeling process is in formulating a tactical model that takes into account the essential features of the problem at hand while neglecting the detailed aspects of operational scheduling. Tactical and operational scheduling are complementary problems. There are several commercially available computerized staff scheduling packages designed to help manage and automate the operational scheduling process. ANSOS, a widely used nurse scheduling system is described in Warner, Keller and Martel (1991).

Tour scheduling problems are frequently modeled as mixed integer linear programs and are known to be difficult combinatorial problems. However, researchers have made both modeling and solution related advances which have made it practical to find optimal or near optimal solutions to such problems on today's desktop computers using commercially available optimization software. The standard formulation for staff scheduling problems is based on Dantzig's set covering approach (Dantzig 1954) and has served as a basic building block for the vast amount of personnel scheduling research that has been done in the past forty-five years. The basic model has been extended many times in order to increase its applicability and usefulness. One of the models upon which this system is based can be found in Isken (1998); details and computational results are in Isken (1995). Several excellent review articles are available (Baker 1976), (Tien and Kamiyama 1982), (Bradley and Martin 1991) and the first textbook devoted entirely to employee scheduling (Nanda and Browne 1992).

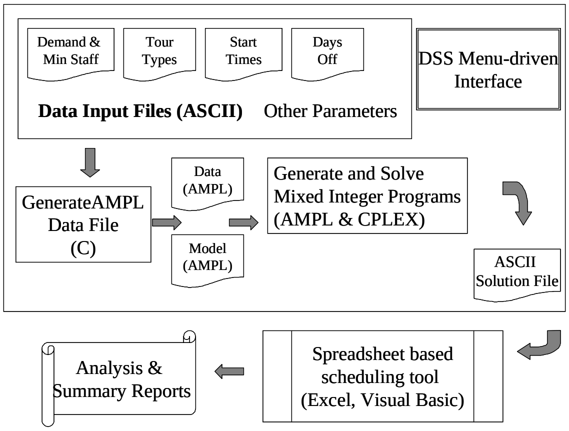
We imbedded a tactical tour scheduling model in a decision support system (DSS) to serve an operations analysis group within a large tertiary care hospital in southeastern Michigan. The operations analysis group consisted of a mixture of industrial engineers, a nurse with a master's degree in business, and other analysts with clinical backgrounds and some degree of business training. This group routinely faced tactical staff scheduling problems as part of general staffing and productivity related analyses. At the time of initial system development, I was a member of this group. The only really "new" aspect of this DSS is the actual tour scheduling formulation itself (Isken, 1995, 1998). It allowed us to solve realistic problems on a desktop computer using standard optimization algorithms that were previously far too large for such a domain. When we embarked on creating this system, we weren't consciously trying to create a DSS per se. We were trying to solve a difficult class of frequently faced problems in an environment (a non-profit hospital) facing severe resource limitations. The design and software architecture of the DSS were influenced greatly by these practical challenges. While this system is an application of a model-based DSS that has endured for eight years (it is still in use today) and has saved a considerable amount of money for the organization, from a DSS perspective, it is very much a work in progress. By describing our experience in designing, building and using this DSS, we hope that others in service industries (e.g. software support, retail sales, emergency services and call centers) in which staff scheduling is an important managerial function may benefit in their efforts to address these difficult problems. Perhaps our experience will also raise issues for further research in the area of DSS development by end-users. After discussing the system architecture we describe problem scenarios in which the DSS was used and describe how outputs of the system are integrated into the larger problem solving process. Finally, we summarize current research and development related to this DSS.

System Architecture

This DSS is best described as a "patchwork quilt" of different software technologies. Its design was influenced by a number of factors. First, other than the author, target users of the system had no more than a passing knowledge of optimization models. Thus, details of model generation and solution algorithms would have to be embedded within a software framework that essentially hid the technical details to the extent possible. Also, in order to leverage the users existing expertise, spreadsheet based tools (Lotus 1-2-3 early on and Microsoft Excel currently) were developed for data input and output manipulation. The heart of the system, the optimization model, is based on standalone versions of relatively expensive modeling (AMPL) and solver (CPLEX) software. Thus, we relied primarily on the relatively crude method of text file passing to communicate model input parameters and solution information between the user and the system. In order to "knit" the components together, we used a combination of the C programming language, a DOS batch language and Visual Basic. The overall system architecture is illustrated in Figure 1.

With this approach, we were able to use very high quality software components at the expense of having to integrate the components. At the time we developed this system (1992), there was no integrated environment that had all of the capabilities we needed. While modeling environments have improved tremendously since then (for example, AIMMS 3 and ILOG OPL Studio), so to have the capabilities and ease with which tools such as Visual Basic, spreadsheets, databases and modeling/solver software can be integrated. Furthermore, the trend for high end analytical applications such as algebraic modeling languages, optimization packages and computer simulation languages has been to facilitate creation of DSS applications by exposing their functionality via constructs such as callable libraries and ActiveX

Figure 1. System Architecture



technology. Since our system, like many DSS development efforts, was planned to be evolutionary in nature (Turban and Aronson, 1998), we believed that a component based approach was more amenable to unforeseen system modifications and enhancements. The designers and developers of a complex, model-based DSS face difficult decisions with respect to selection of software development tools. The DSS software arena is particularly overwhelming and confusing for end-users today due to the very inclusive nature of the term "decision support" and with the commercial emphasis on DSS focusing largely on data warehousing, OLAP and data mining.

Another factor affecting our development approach was that, we as end-users, were also the system developers and worked entirely outside of the established information systems group in the organization. While this contributed to very quick prototype and functional system development, the resulting system is far from a polished, professional information system. Error diagnosis and recovery is minimal as is scenario management. It was also apparent that there was, and still is, a significant chasm between the organization's information systems professionals and business analysis professionals. It is clear that the organization as a whole would have benefited from information system people with a greater understanding of business analysis needs and potential value, and from business analysis people with a better understanding of information systems development and maintenance. Our anecdotal experience suggests that this phenomenon is relatively widespread in healthcare.

Model Inputs

The user interacts with the system first via the creation of several data input files containing problem specific parameter values. In essence, the user models the problem by setting various parameter values, not by changing the optimization model itself. We developed macro driven spreadsheet templates to semi-automate the creation of the input files. A spreadsheet environment is well suited to this task for a number of reasons. The files are tabular in nature. Often the analyst will create several versions of the input files to model different scenarios. Standard spreadsheet editing features such as copying and pasting allow the analyst great flexibility in quickly creating these different scenarios. Each file is exported from the spreadsheet to an ASCII file for use in the model generation process. However, scenario management functionality is lacking. Since a complex project may result in a large number of data input files, this shortcoming results in less input file re-use than desired. Scenario management would likely be better suited to a database application. The five main data input files are described briefly in Table 1.

In addition to the input files there are several input parameters that must be specified by the analyst. The **Start Window Width** is a single integer specifying the number of half hours within which the shift start times for an individual tour can vary from day to day. The **Budget** parameter provides an upper bound on the total amount of staff to be scheduled. This parameter is often used when the objective of the problem is to find a schedule that minimizes the amount of understaffing for a fixed staff size. The **Part Time Fraction** parameter is the maximum fraction of scheduled labor hours provided by part-time tours. Since part-time tours can be quite useful in matching the staffing requirements, practical limits must be set to prevent our "dumb" optimization models from suggesting solutions with unobtainable amounts of part-time staff. This raises a user education issue that we faced. While users were far from experts in the area of mixed integer programming, it was necessary for us to have several in-house training seminars in order to discuss the basic ideas underlying mathematical programming. This was needed so that the analysts could become effective modelers in the sense of knowing how different parameter settings would likely affect the optimization results. We had to diffuse the notion that such models are smart in any meaningful way with respect to the problem domain. Ideally, our DSS could be made more intelligent by capturing some of the modeling expertise of the developers and making it available to the analysts either as a searchable knowledge database or even a rule based modeling expert system. The domain for which the modeling expertise is needed is quite well defined and not prohibitively large.

Table 1: Data Input Files

Demand	A seven row, forty-eight column table specifying target staffing levels for each half-hour period of the week.
Minimum Staff Levels	Structured like the DMD file but specifies absolute minimum staffing levels which must be maintained. This file is relevant for situations in which understaffing is allowed but penalized in the objective function.
Tour Type Mix	This file specifies the tour types to be considered and any constraints providing upper or lower bounds on the number of positions scheduled for each tour type. For example, the number of (8,3) tours (eight hours per day, three days worked per week) might be limited based on department policy or current staff complement.
Allowable Shift Start Times	For each shift length, this file contains a seven row, forty-eight column table consisting of zeroes and ones. A one indicates that a shift of that length can start on that day and at that time; a zero means it cannot.
Allowable Days Off Patterns	This file contains specification of allowable days off patterns for each tour type. Each pattern is specified with a zero or one for each day of the week; a one indicating the day is worked, a zero that it is a day off.

The objective function of the model can be specified to consider only labor cost or a sum of labor cost and understaffing cost. It is part of the art of modeling to specify understaffing cost such that the solution minimizes labor cost while at the same time allows a level of understaffing which is appropriate for the problem being considered.

Model Generation and Solution

The scheduling model itself is written in the algebraic mathematical programming language, AMPL (Fourer, Gay, and Kernighan, 1993). An attractive feature of AMPL is that it enforces the separation of the model from the data. The input data is used to create an *instance* of the model. The structure of the model itself does not change, just the values of its many parameters. Using AMPL allows us to quickly develop specially tailored models based on the nuances of the particular problem if our standard models are insufficient. Often we are able to use these customized models in other projects and have accumulated a relatively unorganized collection of models which we reuse or modify as needed.

To generate an instance of the model, an AMPL data file must first be created. The analysts do not know AMPL nor do they need to do so. A custom "front end" was developed which allows specification of the scenario name, the input files to be used and the values of the other parameters. Originally the front end was created using an extended DOS batch language and the C programming language. It has been recently rewritten in Visual Basic. A menu option initiates a custom C program which reads the input files and parameter values and creates the AMPL data file. Next, the front end controls the solution process by running AMPL in a DOS shell to create the model instance and then passing the problem on to the integer programming solver, CPLEX (ILOG, 1999). CPLEX is one of the leading optimization packages available. It provides a reliable, technically sound basis upon which to build optimization based solutions. It is available as a standalone solver or as a callable library on a wide variety of computing platforms. A number of optimization packages are available for a range of prices and with varying functionality (Fourer, 1999).

Figure 3. Partial Coverage Report

	Target	Sched	+/-	Target	Sched	+/-	Target	Sched	+/-
6:00 AM	7	7		9	10	1	9	10	1
6:30 AM	7	7		9	10	1	9	10	1
7:00 AM	8	8		11	12	1	11	11	
7:30 AM	8	8		11	12	1	11	11	
8:00 AM	8	8		10	10		10	10	
8:30 AM	8	8		10	10		10	10	
9:00 AM	8	8		10	10		10	10	
9:30 AM	8	8		10	10		10	10	
10:00 AM	8	9	1	10	11	1	10	11	1
10:30 AM	8	9	1	10	11	1	10	11	1
11:00 AM	8	8		10	10		10	10	
11:30 AM	8	8		10	10		10	10	
12:00 PM	8	8		9	10	1	9	10	1

After either an optimal solution is found or the analyst specified time limit is exceeded, the solution is written out to a specially formatted text file. It is important to note that the details of model generation and solution are hidden from the analyst. Instead of worrying about technical optimization details, they worry about modeling issues specific to the problem at hand. We should point out that the user does see the progress of the branch and bound process as the problem is being solved. While they may not fully understand all of the technical details, they do quickly learn what a successful search looks like and can quickly tell when something as gone awry. Typical problems are on the order of a few thousand variables and a few thousand constraints and usually solve in anywhere from seconds to several minutes on a 300Mhz Pentium machine with 64Mb of RAM. A much needed improvement to this phase is better automatic handling of events such as the specification of an infeasible model or other problems during the solution phase.

Managing Model Outputs

The output of the solution process is a text file listing the tours. For each tour, the file contains the start time and shift length of each shift worked during the week. Mathematical models such as these seldom provide complete answers to real problems; instead they provide the analyst with good ideas, partial solutions, and a greater understanding of the problem and the qualities of a good solution. It is important that tools be available for analysts to refine the solutions suggested by the scheduling model and to quickly assess the effect on time of day coverage levels of modifying tours. In order to facilitate further analysis, a Visual Basic for Applications driven Excel spreadsheet tool has been developed, which we have called EUREKA. It allows the analyst to import the solution file and the demand file and to calculate the scheduled staffing level, or coverage level, in each half hour of the one week planning cycle. Tours can be altered manually with coverage levels being recalculated with the push of a button. A large number of audit calculations are automatically done as the user makes manual changes to the schedule. The audit calculations make it easy to spot undesirable schedule features that are introduced during manual schedule manipulation. For example, the output of the current generation of models is a one week tour schedule. A heuristic process is used to create a two week schedule from the one week solution. The audit calculations highlight schedule characteristics such as the greatest number of consecutive work days scheduled or total number of weekend days scheduled over two weeks.

Figure 2. Partial One Week Schedule

	Week 1 - Hospital X - Flexible Scheduling Practices			
Tour Type	**Sunday**	**Monday**	**Tuesday**	**Wednesday**
12hr FT	Off	6:00a - 6:00p	Off	6:00a - 6:00p
12hr FT	Off	6:00a - 6:00p	Off	Off
12hr FT	6:00a - 6:00p	Off	6:00a - 6:00p	6:00a - 6:00p
12hr FT	Off	6:00a - 6:00p	6:00a - 6:00p	Off
10 hr FT	6:00a - 4:00p	6:00a - 4:00p	6:00a - 4:00p	Off
10 hr FT	Off	Off	6:00a - 4:00p	6:00a - 4:00p

Figure 4. Start Time Summary

# Tours	Tour Type	Description		
	1	2	3	8
Start	12hr FT	12hr FT	12hr FT	10 hr FT
6:00a	2	2		2
7:00a	1	1	8	
7:00p	1	1	6	
8:00a	1	1		
10:00a				1
6:00p	2	2		
8:00p	2	2		
Grand Total	9	9	14	3

Other summary reports such as the total number of staff scheduled by tour type and start time, by full or part-time status, and by day of week and start time are also created automatically using Excel's Pivot Table functionality. Examples of a partial one week schedule, a coverage report, and a start time summary produced by EUREKA are illustrated in Figures 2, 3 and 4. These reports are the primary means by which the analysts communicate their findings to those involved in the project. The summary report provides a means of comparing the different scenarios based on aggregate measures such as total labor hours scheduled, total positions scheduled and total amount of understaffing. The potential cost of different scheduling policies such as a reluctance to use multiple shift lengths becomes apparent upon examination of the summary. One week schedules are reviewed for impractical or undesirable features that were not originally eliminated via the input files and scheduling parameters. Coverage reports are used to identify magnitude and timing of under and overstaffing occurrences. Staffing surpluses or shortages can often be moved to different times during the day by adjusting shift start times without impacting the overall level of understaffing. Similarly, the impact of a staffing reduction on understaffing can be evaluated using the coverage report. It must be emphasized that these are analysis tools; they are not operational schedules, though they do provide the basis for development of operational schedules.

Uses of the DSS

The DSS was (and continues to be) used in a large number of staffing related projects with scheduling analysis components. Hospital units face the fundamental problem of determining how many people are required to meet workload demand and how they should be scheduled to meet hourly and daily fluctuations. Service level criteria such as patient waiting times or turnaround times for test results, have a direct impact on staffing needs. Scheduling analysis using the DSS can aid in assessing the impact on labor costs of meeting targeted service level goals or suggest ways to better schedule staff to move closer to meeting such goals. Changes in the operating environment such as extending hours of service, physicians joining the staff, or acquisition of new technology can tax staffing resources. Scheduling analysis can suggest ways to meet increased demand with current staff or to minimize staffing additions. "What if?" analysis can be done to answer operation questions related to the appropriate mix of different shift lengths, the mix of full and part-time employees and the benefits of flexibility in shift start times. While political factors may well be the deciding factor in a staffing related decision, at least the decision maker will have had a quantitative estimate of the implications of the decision.

While the details of scheduling analysis projects differ based on the objectives of the study, most have similar components. Most projects involve meeting with department management and scheduling committee. Information is gathered by surveying department managers and staff regarding the scheduling policies, practices, and preferences. Analysis is done to determine time of day staffing targets and consequences of understaffing. This portion of the process is also supported with a variety of computer simulation and queueing based capacity planning models. The process proceeds iteratively with the analyst performing the analysis, sharing the results with the department, and then refining the analysis based on their feedback. Sample schedules, coverage reports and labor cost summaries are the primary tools used to convey the results. The analysis often concludes with the analyst helping to document, standardize, improve and perhaps automate the department's ongoing operational staff scheduling process. A recommendation to purchase a specific computerized staff scheduling package may also be made. It is important to note that our DSS is just one part of the overall process of solving staffing problems.

In addition to the projects at the study hospital, the DSS has been used to analyze the impact of scheduling practices on staffing needs in inpatient obstetrical units in nine different hospitals throughout the country. For these, the DSS was used to quantify the difference in staffing needs between a scenario with limited scheduling flexibility and one with significant, yet realistic, flexibility. Differences ranged from 2%-15% with an overall average of about 8%. Thus, the DSS allowed us to put an explicit monetary cost on lack of scheduling flexibility. This information was then used as part of the managerial decision making process with respect to scheduling policies and staff sizing.

Future R&D

As mentioned, this DSS is by no means a polished, fully integrated system. Instead, it has been cobbled together with a variety of software tools including spreadsheets, Visual Basic, C, text editors, an algebraic modeling language and a mixed integer programming solver. Nevertheless, it has been used to support resource allocation decisions related to staffing and scheduling in a wide range of departments at a large tertiary care hospital and in a number of different healthcare facilities throughout the country. With its value firmly established through extensive experience, we continue with research and development. Current efforts center on theoretical extensions to the tour scheduling modeling and solution techniques, better scenario management, better model management, automated problem specific side constraint generation, improvements to the user interface and the smoother integration of the different software components making up the DSS.

References

Baker, K. 1976, “Workforce Allocation in Cyclical Scheduling Problems: A Survey”, *Operations Research Quarterly*, Vol. 27 No. 1,ii, pp. 155-167.

Bradley, D.J., and J.B. Martin 1991, “Continuous Personnel Scheduling Algorithms: A Literature Review”, *Journal of the Society for Health Systems*, Vol. 2, pp. 8-23.

Dantzig, G. 1954, “A Comment on Edie's Traffic Delays at Toll Booths”, *Operations Research*, Vol. 2, pp. 339-341.

Fourer, R., Gay, D.M., and Kernighan, B.W. 1993, *AMPL: A Modeling Language For Mathematical Programming*, The Scientific Press, South San Francisco, CA.

Fourer, R. 1999, “Linear Programming”, *OR/MS Today*, Vol. 26, No. 4.

Isken, M.W. 1995, “Personnel Scheduling Models for Hospital Ancillary Units”, Unpublished Ph.D. Dissertation, University of Michigan.

Isken, M.W. and W.M. Hancock, “Tactical Scheduling Analysis for Hospital Ancillary Units,” 1998, *Journal of the Society for Health Systems*, Vol. 5, No. 4, pp. 11-23.

ILOG, 1999, ILOG CPLEX 6.5 User’s Manual.

Nanda, R., and J. Browne 1992, *Introduction to Employee Scheduling*, Van Nostrand Reinhold, New York.

Tien, J.M., and A. Kamiyama 1982, “On Manpower Scheduling Algorithms”, *SIAM Review*, Vol. 24, pp. 275-287.

Turban, E. and J.E. Aronson 1998, “*Decision Support Systems and Intelligent Systems”*, Prentice Hall, New Jersey.

Warner, M., Keller, B.J. and S.H. Martel 1991, “Automated Nurse Scheduling”, *Journal of the Society for Health Systems*, Vol. 2, pp. 66-80.