An Efficient Decision Support System for Academic Course Scheduling
Author(s): John J. Dinkel, John Mote, M. A. Venkataramanan
Source: *Operations Research*, Vol. 37, No. 6 (Nov. - Dec., 1989), pp. 853-864
Published by: INFORMS
Stable URL: http://www.jstor.org/stable/171469
Accessed: 15/01/2010 15:02

OR PRACTICE

AN EFFICIENT DECISION SUPPORT SYSTEM FOR ACADEMIC COURSE SCHEDULING

JOHN J. DINKEL
Texas A&M University, College Station, Texas

JOHN MOTE
The University of Texas, Austin, Texas

M. A. VENKATARAMANAN
Indiana University, Bloomington, Indiana

(Received March 1987; revisions received May 1988, January 1989; accepted June 1989)

This paper describes a network-based decision support system approach to the most general form of the academic course scheduling problem. The dimensions of faculty, subject, time, and room are considered by incorporating a penalty function into a network optimization approach. The approach, based on a network algorithm, is capable of solving very large problems. This methodology can be applied to other scheduling situations where there are competing objectives and multiple resources. Such situations include: scheduling of exams, times, and rooms in an academic setting, and scheduling of clients, times, and facilities for physicians, hospitals, dentists, counselors, and clinics. Common problems in such settings include the utilization of available space, and dissatisfaction with assigned times and locations. The proposed system results in more effective room utilization patterns, improved instructor satisfaction levels, and streamlines the tedious scheduling process. We describe the use of the model to schedule all graduate and undergraduate courses in the College of Business Administration at Texas A&M University. This involves 175 faculty, over 300 sections, 20 rooms, and 16 time slots for each semester's scheduling problem.

The poor utilization of available classrooms and the dissatisfaction of faculty with their teaching assignments are common problems encountered when attempting to schedule university classes. This paper presents a network-based decision support system (DSS) that has been successfully used to improve the utilization of classroom resources as well as to consider faculty preferences for subject, room, and time schedules.

During the period 1983–1985, when one of the authors (Dinkel) was in the Dean's Office, this model was used to schedule all undergraduate and graduate classes for the College of Business Administration at Texas A&M University. The use of this decision support system provided an effective method for dealing with a large, complex, and time consuming process in a way that allowed the decision makers, that is, the department heads, to maintain control of the process. The use of the model resulted in improved schedules with a significant reduction in the amount of time required to produce the schedule. Due to the ease of solving the model, it was possible to allow changes in priorities and preferences and to easily present alternative solutions.

The major benefits of the model, which are described in detail in Section 5, were: improved room utilization, significant reduction in unassigned courses, and a consistent approach to time period shifts. In addition, the model greatly reduced the time necessary to produce an acceptable schedule.

The model was used from 1983 to 1985. Since 1985, all the authors have moved to other universities (Mote and Venkataramanan) or have different responsibilities (Dinkel). While the model was used effectively during the rapid growth of the college in the early 1980s, it is apparently no longer used to schedule

Subject classifications: Education systems, planning: academic course scheduling. Networks/graphs, applications: network based scheduling model.

0030-364X/89/3706-0853 $01.25

courses partly due to the leaving of the authors and partly due to the fact that the data have become stable and more predictable.

With little effort, the underlying model can be modified to serve a variety of scheduling environments. The general problem of scheduling faculty, courses, time slots, and classrooms has attracted a great deal of interest. Numerous solution procedures have been proposed and tested. Each approach is designed to address certain aspects of the general scheduling problem. Andrew and Collins (1971) suggested a linear programming model; Dyer and Mulvey (1976) proposed a network model in the context of an integrated decision system. Large-scale integer programming models have been developed by Tillet (1975), Breslaw (1976), and McClure and Wells (1984). None of these optimization models considers the problem of assigning a faculty, subject, or room combination to a particular time slot.

Optimization models that address the time component of the scheduling problem have been proposed by Harwood and Lawless (1975), Shih and Sullivan (1977), and Mulvey (1982). The Harwood and Lawless approach uses a mixed integer goal programming model; the Shih and Sullivan method is based on a two-stage optimization of a zero-one integer programming model; and Mulvey (1982) uses a network model. Selim (1982, 1983) presents a linear program for constructing timetables.

Timetabling methods (Knauer 1974, Schmidt and Strohlein 1979, and De Werra 1985) have been used widely to schedule courses and exams. The overview of such methods, presented by De Werra, stresses the graph theoretic approach.

Other approaches include a heuristic procedure of Barham and Westwood (1978), and a Lagrangian relaxation procedure of Tripathy (1980). None of these approaches considers the classroom availability aspect of the scheduling problem.

The works of Dyer and Mulvey, and Mulvey (1982) placed special emphasis on the inherent network structure of the basic course scheduling problem. They developed a model that allows this embedded network structure to be fully exploited. The objective of their model is to determine an optimal matching of faculty preferences for the courses that are to be offered. While their model does not specifically address the issue of classroom and time slot availability, it serves as an important foundation for our work. Like the work of Dyer and Mulvey, and Glassey and Mizrach (1986), we have developed a DSS that allows us to exploit the underlying network structure of the scheduling problem but unlike their earlier work, we simultaneously consider the dimensions of faculty, courses, classrooms, and time.

1. THE PROBLEM

The general problem of scheduling faculty to courses, to classrooms, and to time of day is faced by every educational institution. From the faculty's point of view, the objective is to maximize their preferences including the room and time of day; from the administration's point of view, the efficient utilization of the physical facilities is a concern as well. The problem of scheduling faculty and subject assignments is well documented in the references and can be viewed as a network optimization model with the usual constraints of

- requiring that all scheduled sections be staffed;
- a maximum, and perhaps a minimum, number of assignments for a faculty member.

The classroom and time assignments introduce the additional constraints that

- a faculty member cannot be assigned more than one course per time period;
- a room cannot be assigned more than one class per time period.

The consideration of classroom and time assignments are important preferences for the faculty. The faculty may express preferences for certain times of day, certain days of the week, back-to-back scheduling, and to avoid certain times of day. In addition, because of pressures on room utilization, there may be a minimum size requirement; for example, a course section must have an enrollment of at least 75% of the room capacity into which it is scheduled. This avoids the problems associated with scheduling a course section of 20 into a room seating 100 in a space-constrained environment.

These last two constraints complicate the situation because they require a large-scale integer programming model in order to deal with the most general setting. In addition, we have the potentially competing objectives of faculty preference and space utilization. In a resource-constrained environment, with great pressure on the rooms, this is an important tradeoff that must be carefully analyzed.

To illustrate the problem and our approach, we use the College of Business Administration at Texas A&M University. The 7,000 full-time students represent a doubling of enrollment over the past 7 years. The college has 175 faculty and direct control over 20 classrooms of a variety of sizes in two buildings. In addition to the growth in the college, the university's enrollment has increased by almost 30% in this period. As a result, there is tremendous pressure on the available classroom space. Thus, in addition to meeting faculty preferences, it is important to utilize space controlled by the college. It is in this constrained setting that a network-based faculty, subject, classroom, and time DSS was developed.

Prior to the adoption of this approach, the administrative personnel of the College of Business Administration made the assignments manually. Each department was given priority over a subset of the classrooms in which they could schedule any of their sections, provided that they were not smaller than 75% of the classroom seating capacity. This approach led to inefficient room scheduling, particularly with respect to small graduate seminar rooms and large lecture halls. For the remaining unscheduled class sections and unused room and time slots, the assignments were made during a group bargaining session. Finally, a list of remaining unscheduled sections was provided to the university for scheduling elsewhere on campus; and unused room and time slots were used to schedule other university courses. This scheduling process was time consuming and resulted in a suboptimal utilization of resources.

2. THE MODEL

The proposed scheduling model is a capacitated, pure network flow problem with a penalty structure in the objective function. There are five classes of nodes or constraints in this network model, as illustrated in Figure 1.

The first and fifth classes are the master source and master sink nodes used to circularize the network. Between these two nodes are three levels of nodes: a departmental level ($i = 1, \ldots, I$), a faculty/subject level ($j = 1, \ldots, J$), and a classroom size ($s = 1, \ldots, S$)/time ($t = 1, \ldots, T$) level. Associated with these nodes are the following arcs.

XC Connects the master sink to the master source. The flow on this arc is the total number of sections scheduled by the model.

XD(i) Connects the master source to each department node (i). The upper and lower bounds on these arcs can be used to set the maximum and

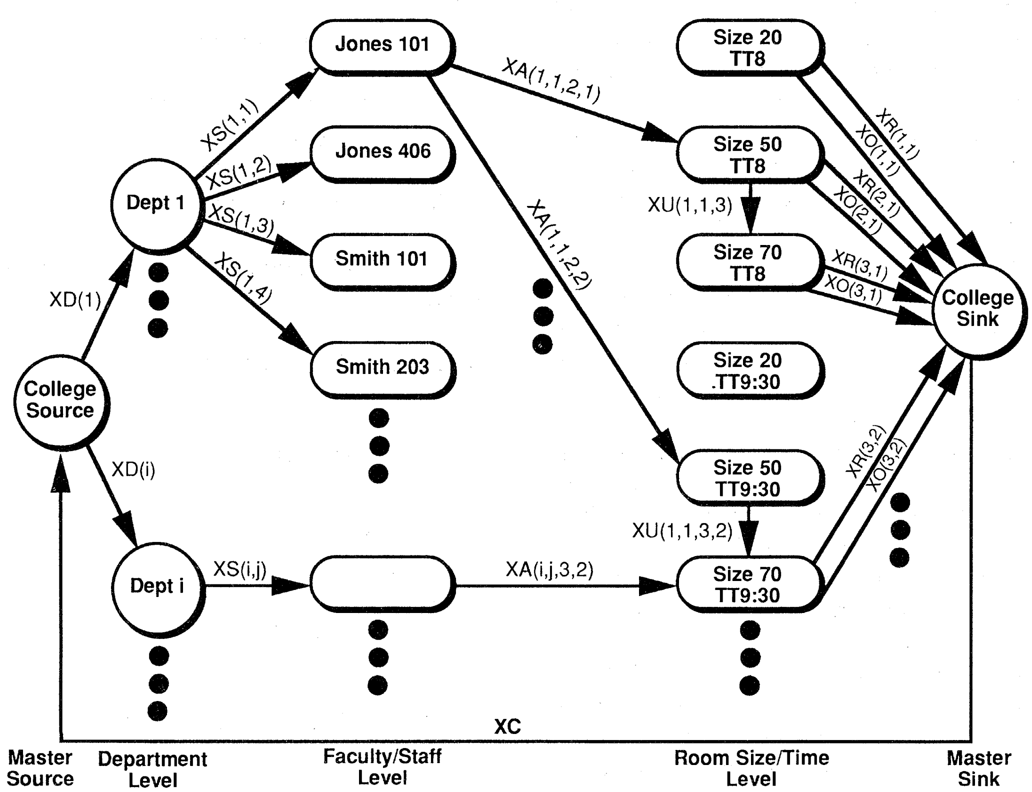

Figure 1. General network based model.

minimum number of sections to be offered by each department.

The objective function coefficients $CD(i)$ can be used to set relative preferences among the departments. For example, we can prioritize the departments by assigning weights to these arcs that reflect the order in which each department's courses should be scheduled.

$XS(i, j)$ Connects each department node (i) to each faculty/subject node (j). The model builder can control the complexity of the model at this point by which faculty and subject combinations are allowed. The most general model includes *all* possible combinations of faculty and subjects. A more realistic model restricts the combinations to all faculty and all subjects by department. In those cases where there are specializations within departments, the combinations can be restricted further to only those within specializations. Or, the faculty may be allowed to state their preferences and only those combinations are included. The upper and lower bounds can be used to define teaching loads by individual faculty.

The objective function coefficients $CS(i, j)$ can be the faculty preferences for a particular subject. They also can be used to give preference to certain faculty or faculty/subject combinations over and above the stated faculty preferences. For example, we can state a preference to first schedule all full professors, then associate professors, and so on.

$XA(i, j, s, t)$ Connects the faculty and subject nodes (i, j) to the room size (s) and time (t) nodes. As with the previous set of nodes, the complexity of the model can be controlled by which combinations are included. The most general model allows the inclusion of all possible combinations of faculty and subjects with rooms and times. A more realistic model includes only those arcs that represent the assignment to rooms with sufficient capacity, and does not violate size restrictions, if any.

Departments specify an upper limit on the enrollment in each section, and since the university enforces a 75% of capacity rule, it is not in their best interest to overestimate enrollment. If a class falls below the 75% limit, it will be moved, possibly across campus, to make room for a larger class that better utilizes the space.

In general terms, all arcs can be included and those deemed unacceptable eliminated by setting the upper bound to zero. A model of more reasonable size can be constructed by including only those arcs that meet the local restrictions. We chose the latter approach in our implementation, and include only the upgrade to the next largest sized room, and denote this variable by $XU(s, t)$. This greatly reduces the size of the model and meets the restrictions on space utilization. The lower bounds are set to zero and the upper bound is set to 1 because each faculty member can teach only one class at a time.

The objective function coefficients, $CA(i, j, s, t)$, reflect the instructors' preferences for rooms and time of day. The structure of these coefficients will be discussed in Section 3. Also, as explained in that section, we can use these coefficients to control the assignment of classes to larger rooms.

$XR(s, t)$ and $XO(s, t)$ Connects each room and time node to the master sink. There is a pair of these arcs for each node (s, t) with XR representing the scheduling of courses into room and time slots; and XO representing the overflow of such slots. That is, $XO(s, t)$ represents the excess number of *required* classrooms and time combinations in order to complete the schedule.

The upper bounds on $XR(s, t)$ represent the number of classrooms of size (s) that are available at time (t).

The objective function coefficients, $CR(s, t)$, can be used to give preference to certain assignments of rooms and time. There are no upper bounds on $XO(s, t)$.

The objective function coefficients, $CO(s, t)$, are set to large numbers to minimize the number of shortages. A positive flow on an $XO(s, t)$ arc represents an assignment that cannot be met within the current space and must be scheduled within the larger pool of university resources.

The network component of the scheduling problem is given as

$$\begin{aligned}\text{minimize}\quad & \sum_i CD(i)XD(i)\\ & + \sum_i \sum_j CS(i, j)XS(i, j)\\ & + \sum_i \sum_j \sum_s \sum_t CA(i, j, s, t)XA(i, j, s, t)\\ & + \sum_s \sum_t CU(s, t)XU(s, t)\\ & + \sum_s \sum_t CR(s, t)XR(s, t)\\ & + \sum_s \sum_t CO(s, t)XO(s, t) \qquad (1)\end{aligned}$$

subject to

$$XC - \sum_i XD(i) = 0 \quad (2)$$

$$XD(i) - \sum_j XS(i, j) = 0 \quad (3)$$

$$XS(i, j) - \sum_t XA(i, j, s, t) = 0 \quad (4)$$

$$\sum_i \sum_j XA(i, j, s, t) + XU(s - 1, t) - XU(s, t) - XR(s, t) - XO(s, t) = 0 \quad (5)$$

$$\sum_s \sum_t XR(s, t) + \sum_s \sum_t XO(s, t) - XC = 0 \quad (6)$$

$$\left.\begin{array}{l} 0 \leq XC \\ LD(i) \leq XD(i) \leq UD(i) \\ LS(i, j) \leq XS(i, j) \leq US(i, j) \\ 0 \leq XA(i, j, s, t) \leq 1 \\ 0 \leq XU(s, t) \leq UU(s, t) \\ 0 \leq XR(s, t) \leq UR(s, t) \\ 0 \leq XO(s, t) \\ \text{where} \\ i = 1, \ldots, I;\ \ j = 1, \ldots, J; \\ \qquad s = 1, \ldots, S;\ \ t = 1, \ldots, T. \end{array}\right\} \quad (7)$$

Table I presents a summary of the model parameters and variables. The range of the indices will depend upon the generality of the model. For example, J will depend upon the generality of the faculty and subject assignments allowed; if there were 100 faculty and 300 sections, there can be as many as 30,000 assignments, or a much smaller number if only certain combinations are allowed (for example, only accounting faculty teach accounting courses, no teaching assistants teach graduate courses, and so on).

This is particularly true when considering the assignment of faculty and subjects to room and size nodes. In the case of 10 different room sizes and 16 possible time slots, we can have 160 possible arcs for each faculty and subject node. In reality, the number of feasible assignments may be much smaller due to considerations such as the 75% of capacity rule and the minimum number of seats based on estimated enrollment.

Constraints 2–5 are the typical conservation of flow restrictions. Since the network is presented in circularized form, all right-hand side coefficients are zero. The summations in all but constraint set 4 should be readily apparent. For (4), there is no summation over the room size index because the assignment arcs (XA[i, j, s, t]) are only constructed into the room size and time nodes of the smallest feasible size. The variables XU(s, t) represent the upgrade to a room of the next largest capacity. Constraints 7 represent the upper and lower bounds as defined in the previous paragraphs.

In the context of the example of the College of Business Administration at Texas A&M University, the model has

5 departments ($I = 5$),
150 faculty/subject assignments ($J = 150$) for each i,
16 time slots ($T = 16$),
7 room sizes ($S = 7$).

By considering only the minimum room size per course section and the upgrade to only the next largest size, we have at most 32 arcs for each faculty/subject node.

The model, (1–7), is a capacitated, pure network flow problem. Due to the unimodularity property of the conservation of flow constraints and the integrality of the lower and upper bounds, any extreme point optimal solution to the problem will possess integer decision variables. Large problems of this type can be solved efficiently; see, for example, Ali et al. (1978).

Unfortunately, the basic model is not yet complete. Specifically, conflict problems arise when the model elects to assign a single instructor multiple classes at the same time, or to assign multiple classes to the same room at the same time. These conflicts are not prohibited by any of the constraints. To prevent this situation, the model should have the following set of conflict avoidance constraints

$$\sum_s \sum_t XA(i, j, s, t) \leq 1 \quad \text{for each } i, j. \quad (8)$$

There is one such generalized upper bound or multiple choice constraint for each combination of department/instructor/subject (i and j) and room size/time (s and t). The summation is defined over the different subjects that a given instructor may teach. These constraints destroy the inherent integrality property of the solution. While the initial formulation, (1)–(7), is a capacitated network flow problem that can be solved routinely by a variety of optimization algorithms (Ali et al., Glover and Klingman 1975, Mulvey 1978), the expanded formulation, (1)–(8), is a constrained network flow or a large-scale integer programming problem that only can be solved by a specialized

Table I
Description of Decision Variables

Decision Variable	Description	Flows	Bounds	Objective Function Coefficient
XC	Circularization arc	Total number of sections	—	0
XD(i)	Connect master source to each department (i)	Number of sections by department	Maximum or minimum number of sections to be offered by department	Relative preferences among departments
XS(i, j)	Connect department (i) with faculty/subject (j)	Number of sections taught by faculty	Number of sections to be taught	Preferences for faculty/subject combination
XA(i, j, s, t)	Connect faculty/subject (i, j) with room (s) and time (t)	Assigns section (i, j) to room (s), time (t)	Upper bound of 1	Preferences for room, time assignments
XU(s, t)	Connect room (s) to room (s + 1) at time (t)	Number of upgrades from room size (s) to (s + 1) at time (t)	Number of rooms of size (s + 1) available at time (t)	Preferences for allowing room upgrades
XR(s, t)	Connect room (s), time (t) to sink	Number of rooms of size (s) used at time (t)	Number of rooms available of size (s) at time (t)	Relative preference among size, time assignments
XO(s, t)	Assignments that cannot be made for room (s), time (t)	Number of classes of size (s) that cannot be scheduled at time (t)	None	Arbitrary (large)

algorithm. The next section describes how we deal with constraints of the form (8) in the context of the network flow model, (1)–(7).

3. PENALTY STRUCTURES

The cost coefficients associated with the faculty/subject to room/time arcs and the room upgrade and overflow arcs are important in the construction and analysis of the model. The structure of the coefficients provides the mechanism to incorporate certain preferences and to avoid certain scheduling conflicts.

3.1. Faculty/Subject to Room/Time Arcs

Faculty, in addition to picking a most preferred time slot, may desire a shorter span of the teaching day and want to avoid 5-day teaching schedules. They may also want to avoid particular time slots. These criteria are incorporated into the model through the time shift cost coefficients, CA(i, j, s, t). These coefficients allow the consideration of a variety of such preferences while maintaining the flexibility to allow movement of sections to achieve effective space utilization. In addition, these cost coefficients can be used to assist in avoiding the multiple assignments of faculty to time slots, but they will not guarantee the avoidance of multiple assignments. The example of Figures 2 and 3 will show such a use of these coefficients.

Each arc connecting a faculty/subject node to a room/time node is given a cost coefficient that indicates the preference for a room/time allocation. For example

$$\mathrm{CA}(i, j, s, t) = \begin{cases} 0 & \text{for most preferred time} \\ F^1 & \text{for next most preferred time} \\ F^2 & \text{for next most preferred time} \\ \vdots & \\ \infty & \text{for infeasible assignments} \end{cases}$$

where the relationship between F^1, F^2, ... can be linear or nonlinear for each person. The choice of the relationship among the levels of preference will affect the ease with which courses are moved to other time slots.

For the example, we can use a linear, V-shaped function. That is, the preferred time as stated by the faculty member is given a cost of zero, and each time period is given a coefficient of 1 for each period away from the preferred time.

Unacceptable times are assigned a value of 99, and the mechanism for dealing with back-to-back requests is given in the following example. The Smith/

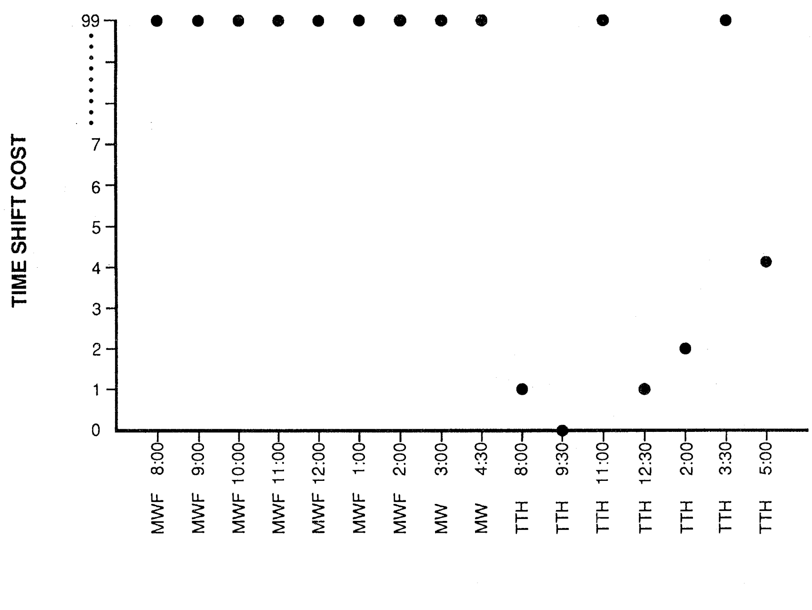

Figure 2. Time shift cost coefficient for Smith/ACC 101, TTh 9:30.

Accounting 101 assignment has a requirement for 125 seats and a preferred time of TTh 9:30; Smith/Accounting 410 has a requirement for 50 seats and a preferred time of TTh 11; Smith does not want to teach in the 3:30 time slot and wants to teach only on Tuesday and Thursday. Figures 2 and 3 illustrate the assignment of costs for arcs from these two nodes to different times. Note that Smith/Accounting 101, TTh 11:00 arc is assigned a cost of 99 and Smith/Accounting 410, TTh 9:00 is assigned a cost of 99, so that if one of the desired times is assigned, conflicts regarding multiple assignments to that time choice constraint slot are avoided.

The penalty scheme is a simple linear one, with a unit increase in the penalty for each time slot away from the desired time. This helps to schedule classes

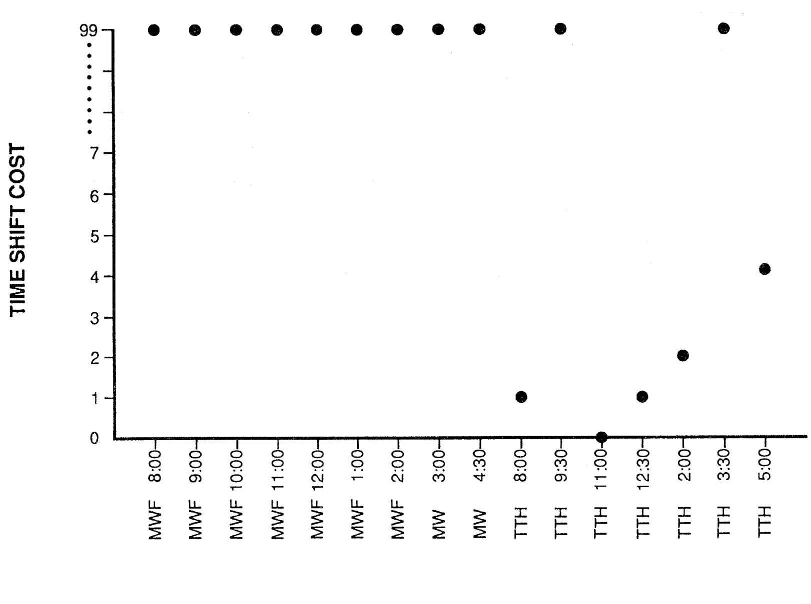

Figure 3. Time shift cost coefficient for Smith/ACC 101, TTh 11:00.

back to back, when one of the desired time slots is not available. It also tends to reduce the span of a teaching day. Assignment of 99 to the TTh 3:30 time avoids scheduling classes for Smith during that time slot. Also, by assigning all MW and MWF times a cost of 99, a 5-day schedule is avoided.

One of the attractive features of this penalty structure is the ability to generate and change automatically such a structure for a large number of faculty. For example, all values are initially set at 99, the most preferred time is reset to a value of 0 and the V-shaped costs are generated automatically with the exception of any nonpreferred times. Also, back-to-back considerations are incorporated by resetting certain costs to 99.

3.2. Room Upgrade Coefficients

A scheme similar to that used for time preferences can be used for the room upgrades coefficients CU. The form of the cost function can reflect the cost of the upgrade. In the case of no prohibition on upgrades, it might reflect unused seats (capacity-enrollment); in the case of certain limitations, a high cost can be assigned to those upgrades, which allows upgrades but only at a high cost. In those cases where there are strict limitations on upgrades, certain arcs can be eliminated or given an upper bound on the flow of zero.

3.3. Room Overflow Coefficients

Room overflow refers to the situation where a particular class cannot be scheduled with the existing resources. Since we want to schedule as much as we can in space we control, and that space may not be sufficient to meet all requirements, we want to make use of these coefficients. We assign a positive value to these coefficients and use them in conjunction with the weighting scheme of the next section. We can use the coefficients to express certain preferences. For example, we can set to 1 the cost of overflows for teaching assistants and a cost of 99 for a certain professor. In this way, we can force the overflows to occur in those areas of less preferred scheduling.

3.4. Weighting Scheme

The three cost components of: faculty preference (CA), room upgrade (CU), and room overflow (CO) can be viewed as competing objectives. This is in addition to their use to express preferences among individuals, for specific assignments, and so on. It is clear that we can significantly alter the assignments by weighting these objectives differently. For example, if we weight the faculty preferences at a zero level, we can improve room utilization but at the expense of moving a lot of assignments to different times and locations.

In order to deal with these objectives, we impose relative weights on the various objective function components. Let M^k, $k = 1, 2, \ldots, 6$ be the relative weight assigned to the objective function components. This results in an objective function of the form

$$
\begin{aligned}
& M^1 \sum_i \mathrm{CD}(i)\mathrm{XD}(i) + M^2 \sum_i \sum_j \mathrm{CS}(i, j)\mathrm{XS}(i, j) \\
& \quad + M^3 \sum_i \sum_j \sum_s \sum_t \mathrm{CA}(i, j, s, t)\mathrm{XA}(i, j, s, t) \\
& \quad + M^4 \sum_s \sum_t \mathrm{CU}(s, t)\mathrm{XU}(s, t) \\
& \quad + M^5 \sum_s \sum_t \mathrm{CR}(s, t)\mathrm{XR}(s, t) \\
& \quad + M^6 \sum_s \sum_t \mathrm{CO}(s, t)\mathrm{XO}(s, t) \qquad (9)
\end{aligned}
$$

where the M^k are chosen to reflect the preferences of the decision maker for the various objectives.

Our completed model is a capacitated network where the objective function coefficients reflect various preferences. In addition, the various portions of the objective function are weighted to give additional consideration to the various objectives.

4. DECISION SUPPORT SYSTEM

The network-based DSS developed for the course scheduling process is made up of three primary software components, as illustrated in Figure 4. The first component is the *Problem Generator* which constructs an MPS-formatted problem file (IBM 1979) based on information extracted from three primary data sets: a room availability file, a faculty/subject to room/time preference file, and a faculty/subject request file. The room availability file contains the classroom inventory by size and time.

The faculty/subject to room/time preference file contains the cost coefficient structure detailed in the previous section. It is treated as a data file rather than as specific programming statements in order to facilitate the analysis of alternative structures.

The faculty/subject file contains the requests from the departments for specific faculty and subject assignments. This file contains a faculty identifier, the department, the course, the class size, and the preferred time. The time preferences will be incorporated into the previous file.

The generated MPS-formatted problem file is passed to the second component of the support system—*The Network Optimizer*. The model that we solve as part of this DSS has an objective function (9) and is subject

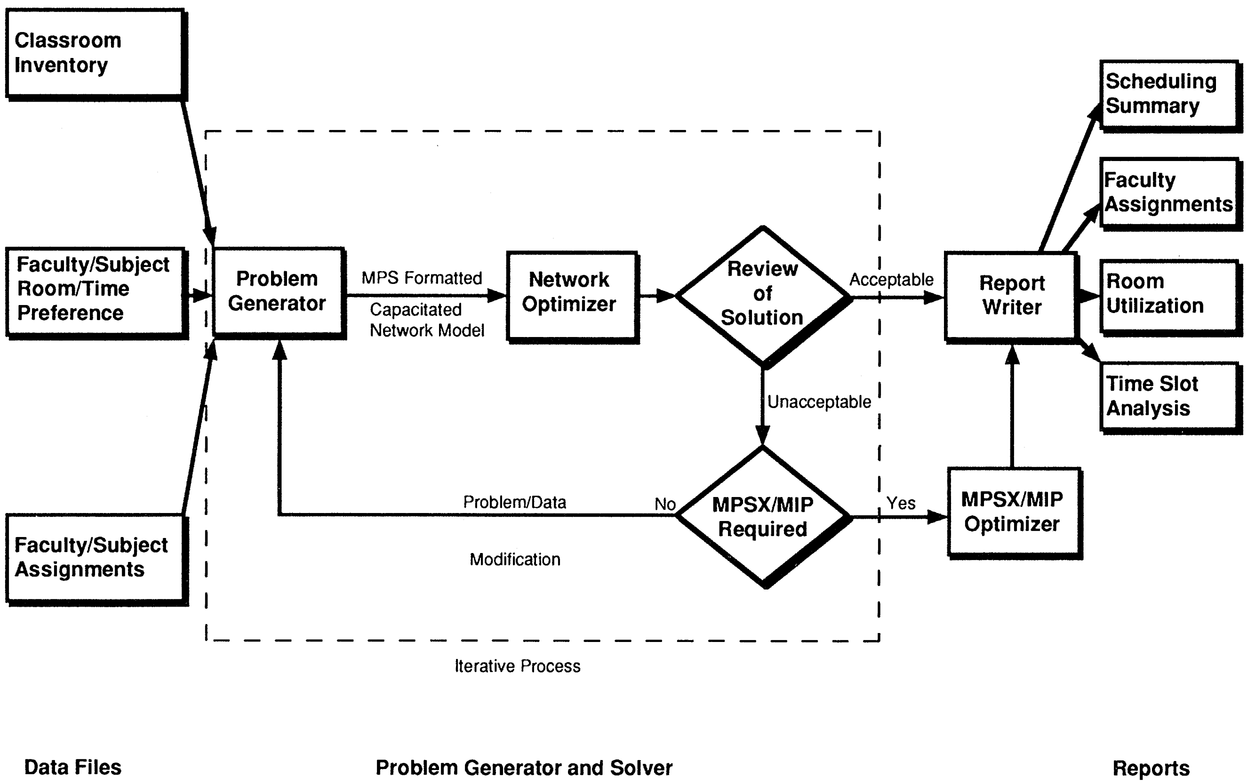

Figure 4. Course scheduling decision support system.

to constraints 2–7. Since we do *not* include the nonnetwork side constraints of the form (8), the network optimizer can be any of the standard network optimization algorithms (Ali et al., Glover and Klingman, Mulvey 1978).

Since there is no guarantee that constraints of the form (8) must be imposed to generate a feasible solution, we initially solve the model without such constraints. If the resulting solution does not contain any conflicts, then we are done. If there are conflicts that require constraints of the form (8), we use the generated solution as an advanced starting point for the constrained problem.

Such a solution methodology does not penalize the user. The incorporation of the nonnetwork side constraints dictates the use of a constrained network algorithm or a general linear and integer programming package such as MPSX/MIP (IBM 1979). Such algorithms are computationally expensive for problems of this size. A widely used implementation strategy is to solve the unconstrained version and use the solution as an advanced start to a constrained algorithm, if required.

One of the main advantages of using a standard network optimization algorithm is the ease of generating solutions. Thus, the decision maker has the opportunity to make changes in the preference structure, the penalty structure, and so on, and can easily generate solutions that reflect these changes.

The third component is the *Report Writer* which takes the output from the optimizer and presents it in usable form. A report is generated for each department giving the teaching schedules for the faculty; a series of summary reports are generated detailing room utilization, time slot utilization, and an overall summary of the schedule.

Prior to the implementation of a final schedule, the summary and room utilization reports are reviewed in order to assess the quality of the solution. The summary report indicates the total number of assigned sections for each department, the total number of section time shifts, the total number of room upgrades, and the total number of room overflows. Based on these composite figures, the decision maker may elect to modify the relative weights assigned to the individual objective functions, or alter the data in the instructor/section request file, or lock-in certain class assignments and then generate a new solution. When the decision maker is comfortable with the generated assignment schedule, the Report Writer can be used to construct the individual faculty assignment reports for each department.

5. COMPUTATIONAL RESULTS

In order to evaluate this model, we use data from the College of Business Administration at Texas A&M University. Over the past several semesters, the scheduling involved about 120 full-time faculty, 40–60 teaching assistants and part-time faculty, and over 300 course sections. There are a total of 16 time slots such as MWF8, MWF9, TTH8, TTH9:30, and so on. The room inventory file is

Size	Number
20	1
36	3
48	6
64	4
78	2
120	3
280	1

It is important to note that our model has a reduced arc set. For example, only room upgrades to the next largest size are allowed, there is no crossing of department lines in faculty and subject assignments, and the room and time arcs are included only for feasible faculty and subject combinations.

The initial form of the model employed a weighting scheme in (9) of

$$\begin{aligned} M^3 &= 1 \quad \text{(Time Shift)} \\ M^4 &= 100 \quad \text{(Room Upgrade)} \\ M^6 &= 10000 \quad \text{(Room Overflow)} \\ M^1 &= M^2 = M^5 = 0. \end{aligned} \tag{10}$$

This form of the objective function puts the emphasis on avoiding room overflows by allowing room upgrades and time shifts; time shifts are preferred to room upgrades. Recall that the V-shaped function within the individual coefficients will control unreasonable changes in a faculty member's schedule.

Table II contains the computational results for the network model, (1)–(7) with objective function, (9)–(10). The table also presents a comparison of the solution times for a primal simplex network algorithm (Glover and Klingman) and the general purpose linear programming algorithm MPSX (IBM). These results are not unexpected, as the efficiency of network algorithms is well documented (Ali et al., Glover and Klingman, and Mulvey 1978). The results are presented for the purpose of comparison. While one could use a general purpose algorithm to solve these models, it is clear that as the size of the model increases and repeated solutions are required, only the use of a special purpose network algorithm is realistic. The cost of the run is presented because it combines all aspects of running a job: memory, I/O, disk, and execution time and converts these to a single number. In that sense, the cost is a measure of all resources required to run a job.

Table II
Solution Results for Penalized Model, (1)–(7)

Item	Semester Spring 1985	Fall 1985	Spring 1986
Problem Description			
Number of rows	330	338	311
Number of columns	2231	2658	2009
A matrix nonzeros	6383	6643	5742
Density	0.75	0.74	0.80
Number of instructors	161	180	154
Number of sections	312	319	287
MPSX Results[a]			
Input seconds	0.6	0.6	0.6
Output seconds	1.2	1.2	1.2
Optimization seconds	60.6	105.6	49.8
Number crash iterations	1703	1700	1510
Number phase I iterations	1953	2058	1763
Total iterations	3168	3249	2845
Cost ($)[b]	109.82	141.53	91.01
Network Results[a]			
Input seconds	1.2	1.33	1.14
Output seconds	1.13	1.23	1.08
Optimization seconds	0.494	0.585	0.411
Number of pivots	3045	3381	2603
Cost ($)[b]	4.02	4.15	3.62

[a] The computer was an Amdahl 5860 with 32MB of memory, with an MVS operating system.
[b] The cost represents total processing using the Texas A&M University charging algorithm, which includes I/O, memory, and CPU utilization. These costs do not include printing.

Table II shows that the approach presented here, when combined with an efficient network optimizer, can be used to solve large, complex assignment models. Once the data sets are created, repeated solution of a model can be accomplished effectively.

It is important to note that while we initially anticipated that the optimal solution would have multiple assignments of faculty to the same time slot, this did not occur in any of the solutions. This is due to the structure of the V-shaped function that prevents such multiple assignments.

While the computational results are impressive, we are more interested in the quality of the solution in terms of the teaching assignments. In order to compare the quality of the solutions, we must clearly define the processes used.

5.1. Manual Process

1. The faculty submits course and time preferences to the department head.

2. The department head prepares a preliminary schedule using rooms assigned to the department. The department head also may exchange rooms with other departments. Some time and location tradeoffs and room upgrades may take place.
3. All departments meet with unassigned rooms and courses and attempt to assign the maximum possible. This will result in time and location tradeoffs as well as room upgrades. The outcome of this step is referred to as the best manual solution.
4. A list of unassigned rooms and courses is sent to the university for scheduling.

This process typically takes place over a 7–10 day period and involves about 6 hours of time on the part of each department head or their designee.

5.2. Automated Process

1. The departments submit the course and time preferences to the Dean's Office. This may include preferences for back-to-back assignments, avoiding certain times, specific teaching schedules, and so on.
2. The model is solved and checked for any infeasibilities, for example, two sections in the same room at the same time, and so on. It is this schedule with infeasibilities corrected that is referred to as the best network solution. This may require making certain assignments and resolving the model.
3. The results are returned to the department for review and any further alteration in assignments.

The input process typically takes about an hour once the data base has been built. A comparison of the solutions is given in Table III.

Based on the cost structure in (10), the network model significantly reduces the number of unassigned courses and rooms. The number of unassigned rooms is reduced by at least 30 per semester. This results in a similar reduction in unassigned courses. The algorithm achieves this by moving course and time assignments, indicated by Total Time Shifts in Table III. This requires anywhere from 43 to 54 changes in a semester schedule. The new assignments are presented to the departments for their evaluation. The number of upgrades appears to increase significantly; however, the actual change is smaller because the manual solution process includes such changes as the department heads bargaining for rooms among themselves.

Note the increase in the number of unassigned rooms in the Spring 1986 semester. This is due to increased enrollment which has pushed many courses beyond the smaller room capacities. The majority of the unassigned rooms are in the smallest room sizes.

Table III
Comparison of Manual Solution to Network Model

Best Solution	Spring 1985	Fall 1985	Spring 1986
Manual Process			
Unassigned room/periods	94	85	108
Unassigned courses	46	42	35
Room upgrades	15	14	11
Network Model			
Unassigned room/periods	53	54	78
Unassigned courses	5	11	5
Room upgrades	25	36	3
Total time shifts	54	47	43
1 period	38	35	32
2 periods	14	9	11
3 periods	2	3	0

The network model can be used to evaluate tradeoffs among various objectives; for example, room utilization versus room upgrade, time shift versus faculty preferences, and so on. Because of the efficient solution procedure, we can allow the decision maker to evaluate such tradeoffs by changing the penalty structure, (9)–(10) and resolving the model. Table IV presents a series of such analyses for various penalty structures. It is clear from these results that as the penalty on faculty preferences increases, the number of time shifts decreases.

The problems that were solved for this paper did not require the solution of models with the additional nonnetwork constraints (8). That is, there were no multiple assignments of professors to courses, or courses to rooms; thus, there was no need to adjoin the additional constraints.

Had there been multiple assignments, we adjoin the appropriate constraints (8), use the generated solution as an advanced starting point, and solve the model using MPSX/MIP or a similar general purpose algorithm.

Table IV
Tradeoff Analysis for Spring 1985 Assignments

	$M^3 = 1$, $M^4 = 10$, $M^6 = 10000$	$M^3 = 10$, $M^4 = 1$, $M^6 = 10000^b$	$M^3 = 10000^a$, $M^4 = 10$, $M^6 = 1$
Unassigned courses	5^c	5	52
Unassigned rooms	53	53	100
Room upgrades	25	43	0
Time shifts	54	43	0
1 period	38	36	0
2 periods	14	7	0
3 periods	2	0	0

[a] Emphasis on meeting faculty timing preferences.
[b] Emphasis on assigning all courses.
[c] From Table III.

The solution of large-scale network models with nonnetwork side constraints has been the subject of recent research. Most approaches based on partitioning the problem constraints into network and nonnetwork bases that use a single surrogate constraint or a general purpose algorithm can be accelerated by an advanced starting solution (Venkataramanan 1987).

The network model presented here provides a reliable approach for the assignment of courses to instructors, to rooms, and to time slots. The model and the solution method are much more robust than previous models in both the detail and size of the problem that can be solved effectively. Based on the example data, the quality of the solution generated by the model is as good as that generated by the manual process. The ease with which the model can be changed and resolved makes it possible to use the model to evaluate alternative scenarios.

Once the data base for a particular setting is created, it is likely that it can be modified easily for future applications. For example, it is likely that the Fall, Spring, and Summer schedules will remain somewhat the same from year to year. Thus, rather than re-create the data base, it can be modified to reflect current needs and preferences.

5.3. Some Comments on Implementation

Since each instructor may state his or her preferred time for each course section and time, there was no incentive to try to trick the system. There is no advantage, and in fact it is a disadvantage, to state anything but the most preferred time for each course section. The tactic most observed was the overstatement of enrollment to attempt to capture a larger or more preferred room. The strict enforcement of the 75% occupancy rule by the university quickly discouraged this behavior.

In the face of less predictable enrollments than those presented here, the decision maker may want to examine the impact of different penalty structures on room upgrades and overflows. Since we are solving a capacitated network model, we can easily and quickly resolve the model under different assumptions.

For example, if there are wide variations in enrollment, the penalty structure on the room overflow can be relaxed. This allows assignment to larger rooms at the expense of other preferences. Given the ease with which this model can be set up and solved, such repeated analysis is feasible.

REFERENCES

ALI, A. I., R. V. HELGASON, J. L. KENNINGTON AND H. S. LALL. 1978. Primal Simplex Network Codes: State of the Art Implementation Technology. *Networks* **8**, 315–339.

ANDREW, G. M., AND R. COLLINS. 1971. Matching Faculty to Courses. *College and University* **46**, 2, 83–89 (Winter).

BARHAM, A. M., AND J. B. WESTWOOD. 1978. A Simple Heuristic to Facilitate Course Timetabling. *J. Opnl. Res. Soc.* **29**, 1055–1060.

BRESLAW, J. A. 1976. A Linear Programming Solution to the Faculty Assignment Problem. *Socio-Economic Planning Sciences* **10**, 227–230.

DE WERRA, D. 1985. An Introduction to Timetabling. *Eur. J. Opnl. Res.* **19**, 151–162.

DYER, J. S., AND J. M. MULVEY. 1976. An Integrated Optimization/Information System for Academic Departmental Planning. *Mgmt. Sci.* **22**, 1332–1341.

GLASSEY, C. R., AND M. MIZRACH. 1986. A Decision Support System for Assigning Classes to Rooms. *Interfaces* **16**, 5, 92–100.

GLOVER, F., AND D. KLINGMAN. 1975. Real World Applications of Network Related Problems and Breakthroughs in Solving Them Efficiently. *ACM Trans. Math. Software* **1**, 47–55.

HARWOOD, G. B., AND R. W. LAWLESS. 1975. Optimizing Organizational Goals in Assigning Faculty Teaching Schedules. *Dec. Sci.* **6**, 513–524.

IBM Mathematical Programming System Extended/370, Program 5740-XM3. 1979. IBM Corporation, White Plains, NY.

KNAUER, B. 1974. Solution of a Timetable Problem. *Comput. Opns. Res.* **1**, 363–375.

MCCLURE, R. H., AND C. E. WELLS. 1984. A Mathematical Programming Model for Faculty Course Assignments. *Dec. Sci.* **15**, 409–420 (Summer).

MULVEY, J. M. 1978. Testing of Large-Scale Network Optimization Program. *Math. Prog.* **15**, 291–315

MULVEY, J. M. 1982. A Classroom/Time Assignment Model. *Eur. J. Opnl. Res.* **9**, 64–70.

SCHMIDT, G., AND T. STROHLEIN. 1979. Timetable Construction - An Annotated Bibliography. *Comput. J.* **23**, 307–316.

SELIM, S. M. 1982. An Algorithm for Constructing a University Faculty Timetable. *Comput. Educ.* **6**, 4, 323–332.

SELIM, S. M. 1983. An Algorithm for Producing Course and Lecturer Timetables. *Comput. Educ.* **7**, 2, 101–107.

SHIH, W., AND J. A. SULLIVAN. 1977. Dynamic Course Scheduling for College Faculty Via Zero-One Programming. *Dec. Sci.* **8**, 711–721.

TILLET, P. I. 1975. An Operations Research Approach to the Assignment of Teachers to Courses. *Socio-Econ. Plan. Sci.* **9**, 314, 101–104.

TRIPATHY, A. 1980. A Lagrangian Relaxation Approach to Course Timetabling. *J. Opnl. Res. Soc.* **31**, 599–603.

VENKATARAMANAN, M. 1987. Constrained Network Algorithms. Ph.D. Thesis, Texas A&M University, College Station, Texas.