INTELLIGENT SCHEDULING SUPPORT
FOR THE U. S. COAST GUARD

Ken Darby-Dowman*
Cormac Lucas*
Gautam Mitra*
Raymond Fink**
Leonard Kingsley***
J. Walter Smith***

*Brunel, The University of West London
**Idaho National Engineering Laboratory
***United States Coast Guard R & D Center

CONF 921127-1

TABLE OF CONTENTS

1. BACKGROUND
2. AN OVERVIEW OF THE INTEGRATED SYSTEM
3. DISCRETE OPTIMIZATION MODEL FOR SCHEDULING
4. SCHEDULE DIAGNOSIS AND ANALYSIS
5. DISCUSSIONS
6. REFERENCES

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

MASTER

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

for

1. BACKGROUND

The United States Coast Guard (USCG) undertakes several important missions in the maritime regions of this country. These responsibilities are classified as:

- (i) Search and rescue.
- (ii) Law enforcement.
- (iii) Response to (marine) environmental incidents.
- (iv) Fishery and custom regulation.
- (v) Vessel safety.

A number of platforms, that is, vessels (cutters) and aircraft of different types are normally deployed to carry out these missions as appropriate. In recent times requests for these platforms have increased tremendously. For instance, the growing usage of small boats by persons without adequate nautical training has increased the search and rescue missions substantially. Law enforcement requirements have also expanded to combat drug related activities and to check the rivalries of different nations in the use of natural resources at sea. These growing demands call for better usage of already scarce sea and air platforms which are at the disposal of the USCG. Thus, the USCG operational staff are required to make efficient scheduling decisions under severe time, fiscal and legal constraints.

The long-term planning and scheduling of diversified Coast Guard platforms, engaged in a variety of missions, is a complicated process. Thus the Coast Guard has assigned the scheduling offices so platforms having similar operating profiles are grouped together to simplify the processes. Platforms are further separated on the east and west coasts under Commands which are called LANTAREA (Atlantic Area) and PACAREA (Pacific Area). Moreover, cutters and aircraft are scheduled by different offices within the Area commands. Each Area is further subdivided into several districts. The District commanders manage all operational and administrative matters arising in their districts. The Area Commander, however, regulates all operational missions that are "multi-district" in geographical region, scope, or resources.

Periodically, the Area Commander's staff prepares and formally issues operational schedules for the platforms under their regional control. These schedules serve as the operational orders to the Commanding Officers of the platforms, specifying the duties that the platform is required to perform at a given time, duration and location. Moreover, the schedule has a large impact on morale,

informing the crews when they can expect to be home with their families, and when and for how long they will be away.

Typical schedules include assigned durations for maintenance, patrols, public affairs events, and training and other exercises. Moreover, specific doctrines impose restrictions on the number of days a particular platform may be away from homeport and the minimum number of days that a platform can be assigned to a particular patrol location. In addition, the intervals between training and maintenance actions are fixed by Coast Guard policy. However, contracting maintenance facilities for engine overhauls and technology upgrades result in uncertainties in the starting and ending dates of a given assignment producing volatile schedules. Further, unscheduled maintenance and emerging mission requirements may cause adjustments to be made to an existing schedule as platforms are reassigned to unscheduled activities. Much greater effects can be caused by such extremes as a military action (that requires Coast Guard assets) or a sudden loss in the budget for operating funds.

Individual platforms, scheduled by separate offices, frequently must work with one another on particular missions. For example, some cutters are equipped with a flight deck for carrying helicopters on patrol. A conflict can occur if the helicopter is unavailable during a cutter/helicopter patrol assignment. Likewise, various training exercises might require platforms which have been scheduled separately by other authorities, that is, U. S. Navy assets. So, even though schedules are developed independently, the appropriate scheduling office must operate within an environment that contains a high degree of cooperation. This poses a burden on managers in the separate offices faced with decision making under constraints of time and conflicting interests.

Taking the above factors into account in the schedule production process, while trying to increase the operational time for each platform, was a difficult and tedious. In the past, the schedule was produced manually (with pencil and paper). Adjusting the basic input repeatedly, to produce a more accurate schedule, required many professional staff hours. So, the quality of the schedule, measured as a function of cost and platform utilization, was degraded by this inability to produce new schedules, as the old ones became infeasible, in a timely manner. In addition, examining proposed changes to policy such as increasing operational day limits within a particular patrol location, required generating new schedules to test out the impact of these changing policies.

The scheduling authority's stated goals are to write a "maximum utilization" schedule, provide statistics such as the number of operational days per quarter for a particular platform and offer alternative solutions based upon changes in scheduling doctrine. Historically, the Coast Guard relied on the skill and experience of

human schedulers alone to "load" the platforms with assignments. However, as the number of platforms grew, the scheduling process became so complex that human schedulers could not handle the process without being aided by the latest computer hardware and software. So, the scheduling process required automation.

This paper will discuss a joint effort by the U. S. Coast Guard Research & Development Center, Idaho National Engineering Laboratory and Brunel University to provide the necessary tools to increase the human scheduler's capability to handle the scheduling process more efficiently and effectively. Automating the scheduling process required a system that could "think" independently of the scheduler, that is, the system needed its own control mechanism and knowledge base. Further, automated schedule generation became a design requirement and sophisticated algorithms were formulated to solve a complex combinatorial problem. In short, the resulting design can be viewed as a hybrid knowledge-based/ mathematical programming application system.

Section 2. below examines the design for an intelligent platform scheduling support system which is expected to be operational early next year. Next, section 3 describes a generalized set partitioning model which is being used to automatically generate "good" schedules for USCG Area cutters. This model has been fully tested and "fine-tuned" to produce timely schedules for implementation in the scheduling support system within the next few months. Although a mathematical program lies at the heart of the system, the model and the surrounding user interface have been designed in such a way that the user needs no optimization expertise. For instance, a screen-based user modeling interface allows the scheduler to specify new scheduling requests by either amending an existing scheduling request stored in the database or by creating a new schedule. Then, section 4 describes the use of expert system technology to analyze generated schedules since the mathematical formulation cannot capture all the rules that satisfy all the scheduling requirements. Finally, section 5 describes a particular application, the Cutter Scheduling Assistant Program (CSAP), which is being used in both Area offices to schedule large cutters. In particular, unique features such as the graphics-oriented direct manipulation interface and expert system are shown to be invaluable components for both usability and accuracy of schedule outputs.

2. AN OVERVIEW OF THE INTEGRATED SYSTEM

Figure 1. below is an overview of the integrated system. The Relational Data Base System (RDBS) contains platform information, port information, historical assignments, planned schedules and task requirements. To protect the integrity of schedule data all transactions and queries must pass through the 'Data Conversion and Validation' buss. This will prevent existing schedules from becoming corrupted during analysis studies. The buss also controls access to all records of schedule changes to be used for planning purposes.

Figure 1. Overview of Integrated System

The interface buss provides a mechanism for transfer of schedule data between (bi-directional) scheduling support modules such as the RDBS and the 'Diagnosis & Analysis' module. The most typical usage pattern is; furnish state description information to the 'Diagnosis & Analysis' module, create and change assignments in a schedule, provide description of the modified schedule to the 'User Interface' module, load, display and (optionally) store the changed schedule in RDBS.

The 'User Interface' module is designed to be mouse-controlled and menu driven, with elements of graphics-oriented direct manipulation. The menu system allows a novice user to start using the system quickly, eliminating syntax errors or the need to memorize the commands. An example of direct manipulation is choosing a patrol location to assign a given platform. The user positions the cursor on the desired platform's timeline and drags the mouse for the desired patrol duration.

The 'Schedule Generation' module constructs the scheduling model and performs analyzes of the solution output produced by the optimization module. The module's routines check the solution for feasibility and send updates to the 'Data Conversion & Validation'

buss for possible inclusion as a planned schedule in the Data Base. The scheduling model is constructed by generating, for each platform, a set of possible schedules. The operational rules that govern the tasks and the platform duties are obtained from the Data Base through the 'Data Conversion & Validation' buss. Finally, the 'Schedule Generation' module uses an optimization package to determine the "best" fleet schedule by selecting one of the possible schedules. The optimizer is specially customized to solve a generalized partitioning integer program efficiently. The next section fully describes this approach for a specific platform application.

A flat ASCII file format is used for the data interface. Although a commercial database product could be used, this has the disadvantages of additional cost, and would not be available for all computer systems that might be used in exploring methods for platform schedule construction, modification and optimization. The flat file format, while inelegant, also simplifies transfer of data from one physical location to another using electronic mail.

The content of the interface file fully describes the state of the schedule. This interface will not include all of the information known to the system, but only that which is considered necessary for the automated schedule construction program. The following general types of information is represented: starting date for known schedule, description of each platform involved in the schedule, description of each activity (type of assignment) involved in the schedule, description of each assignment in the schedule, description of historical assignments occurring before the schedule start date, and description of each physical location relevant to the schedule.

A platform is described by the following information; name, class (type) of the platform, current location, future location, commissioning date and decommissioning date. An activity is described by the following attributes; name and a location (one or more occurrences) where this activity can take place. An assignment can be thought of as a relation between a platform and an activity for a specified period. It is described by platform name, activity name, start date, end date, location name (the location where this assignment will take place), statistical information such as the number of days underway, and the start date of the last assignment of a platform to a particular activity prior to the start date of the current context.

Finally, the 'User Interface' module graphically displays the "planned schedules" on a large color monitor in Gantt chart form. Using rules obtained from experienced schedulers, the module performs various diagnostics to critique generated schedules. In addition, the module contains software to edit a "planned Schedule." The software can be used to enter the entire plan manually or to update an existing or computer generated schedule.

3. DISCRETE OPTIMIZATION MODEL FOR SCHEDULING

In the following, we refer to a particular application discussed in section 5. In this application, the difficulty of creating and maintaining cutter schedules was substantially reduced by the 'User Interface' module and the 'Diagnosis & Analysis' module. However, to adjust the basic input repeatedly in the reactive mode or to examine proposed changes to policy requires generating new schedules to test out the concepts. Manual generation is time consuming and would demand an unacceptable amount of professional staff hours. Therefore, a model was needed to generate realistic 'good' schedules for the 'Schedule Generation' module.

As discussed in section 1, the tasks assigned to the cutters are varied and include patrolling in specified localities of the area, training exercises and maintenance. The feasibility of a particular cutter schedule is governed by a set of operational rules that depend, in part, on the timing and nature of the tasks already assigned to the cutter before the start of the scheduling year in question. Other factors include transit allowances before and after a task, the duration of in-port time after the completion of a task and cutter capabilities. The requirements placed on the fleet fall into two principal categories. The first relates to minimum levels of coverage in terms of the number of cutters of given classes on patrol in each category at any given time. The second category of requirement concerns training and cutter maintenance. In these cases upper limits are placed on the number of cutters performing these tasks at any given time.

One approach to modeling scheduling problems of this type is to generate, for each cutter, a set of possible schedules, and to determine the "best" fleet schedule by selecting one of the possible schedules for each cutter. This formulation leads to an integer programming model which has been widely advocated (e.g. [3], [5]) in various guises.

However, in many scheduling applications, especially in the area of vehicle scheduling, the list of "requirements" often results in the non-existence of a feasible solution. In such cases, a relaxation of the requirements is necessary to obtain a schedule. On further examination of the practical issues it is frequently the case that some of the "requirements" merely reflect desirable characteristics rather than strict requirements. In developing a scheduling model that yields useful solutions, Darby-Dowman and Mitra[2] proposed the extended set partitioning model which was essentially an integer goal programming formulation of a set partitioning model and admitted set covering, set packing and set partitioning as special cases.

In that model, the requirements were treated as targets and undercover (below target) and overcover (above target) were allowed but penalized. Our approach to the cutter scheduling problem follows a similar vein.

Within the operational rules that govern the tasks and the cutter duties, a set of possible schedules is generated for each cutter. An optimal schedule is one which is as close to meeting requirements as is possible. The generic model is formulated as minimize

$$\sum_{i=1}^{ng} \sum_{j=1}^{nt} (w_{ij}^- u_{ij} + w_{ij}^+ o_{ij})$$

Subject to

$$\sum_{k=1}^{nc} \sum_{s=1}^{n_k} a_{ijk s} x_{ks} + u_{ij} - o_{ij} = r_{ij}$$

$$\sum_{s=1}^{n_k} x_{ks} = 1$$

where

Parameters

- nc = number of cutters to be scheduled
- n_k = number of columns (possible schedules) for cutter k ,
 $k = 1, 2, \dots, nc$
- nt = number of time periods in the scheduling year
- ng = number of constraint groups (schedule requirements)

Index sets

- i = schedule requirements, $i = 1, 2, \dots, ng$
- j = time period, $j = 1, 2, \dots, nt$
- k = cutter identifier, $k = 1, \dots, nc$
- s = identifier of possible schedule for cutter k ,
 $s = 1, \dots, n_k$

Model Variables

- x_{ks} = 1 If possible schedule s for cutter k is selected
0 otherwise
- u_{ij} = Extent of the under-achievement in respect of schedule requirement i in time period j
- o_{ij} = Extent of the over-achievement in respect of schedule requirement i in time period j

Model Coefficients

$a_{ijks} = 1$ If possible cutter schedule s for cutter k
contributes to schedule requirement i in time
period j
 0 otherwise

r_{ij} = target/limit/threshold for schedule requirement i in
time period j in terms of number of cutters
contributing to the schedule requirement

w_{ij}^- = Penalty for each unit of under-achievement in respect
of r_{ij}

w_{ij}^+ = Penalty for each unit of over-achievement in respect
of r_{ij}

The model is stated in a generic form. In any given application instance, simplifications may be possible. For example, if schedule requirement i is such that r_{ij} represents a desired lower limit than the penalty for over-achievement, w_{ij}^+ , can be set at zero and o_{ij} considered as a logical variable. Similarly, if schedule requirement i is such that r_{ij} represents a desired upper limit then the penalty for under-achievement, w_{ij}^- can be set at zero and u_{ij} considered as a logical variable.

The number of time periods in the scheduling year is a matter for judgement in relation to the individual application. A day-to-day schedule covering for one year is required. Thus in its simplest form, n_t equals 365 or 366. However, model size and hence solution time can be reduced by considering a larger time unit. In certain cases this can be achieved without loss of model validity. For example, if the duration of tasks and activities is always an integral number of weeks, the problem can be modeled on a week-to-week basis with n_t reduced to 53. Even if the duration of tasks and activities is not always an integral number of weeks it may still be worthwhile to adopt the time unit change to obtain solutions more quickly with a possible sacrifice on solution quality.

In this application, the approach taken is to generate a daily model and convert it to a weekly model which is then solved. The optimal weekly schedule is then matched with the columns of the daily model to obtain the daily schedule. A post processor is then applied which determines whether beneficial time shifts within weeks can be made to tasks. The daily to weekly conversion adopts a "conservative" approach such that the weekly solution results in

an upper bound on the penalty associated with the schedule. Thus, if a constraint group i is a "minimum cover" group,

$$a_{ijks} = \begin{cases} 1 & \text{if the possible (daily) schedule } s \text{ for cutter } k \\ & \text{covers all 7 days of week } j. \\ 0 & \text{Otherwise} \end{cases}$$

However, if constraint group i is an upper limit group,

$$a_{ijks} = \begin{cases} 1 & \text{if the possible (daily) schedule } s \text{ for cutter } k, \\ & \text{covers any day of week } j. \\ = 0 & \text{Otherwise} \end{cases}$$

Although schedules are obtained using an integer programming model, the user runs the system routinely without any specialist mathematical programming knowledge. To achieve this, a robust model that can be reliably and predictably solved is needed. The structure of the model is such that an integer feasible solution is guaranteed since undercover and overcover are allowed. Solving the model in a predictable time has implications on model size and solution strategy. The number of rows in the model is constant in the medium term and will only change if, for example additional cutters are allocated to the area or a major redefinition of localities within the area takes place. The number of columns is governed by the number of potential cutter schedules generated. Experience has shown that, although this number could be extremely large, a stratified random selection of a fixed number of potential schedules for each cutter results in a high quality schedule produced by the model. Regarding the solution strategy, a full branch and bound search is impractical to implement on models of this size. The following simple strategy is highly effective.

Step 1: Solve the LP relaxation. If integer feasible go to END.

Step 2: Fix the cutter schedule variable with the largest value to 1. (Fix the other schedule variables for that cutter to 0).

Go to step 1.

END: Report solution

Experimentation with more sophisticated strategies has not resulted in a significant improvement in solution quality although requiring substantial increases in solution time.

The size of the models when the number of possible cutter schedules is restricted to a maximum of 150 per cutter is shown below.

	Model	
	Daily	Weekly
Number of constraints	3324	507
Number of variables x_{ks}	4156	4156
Number of variables u_{ij}, o_{ij}	3294	477
Average number of non-zeros per x_{ks} column	240	30

Weekly models of this size can be routinely solved in 20-30 minutes on a 486 (33MHz) machine using the optimizer FortLP developed at Brunel University.

To summarize, the overall system can be viewed as a hybrid knowledge-based and mathematical programming application system. The model generator incorporates not only the strict rules governing schedule acceptability but also other knowledge that is used to restrict the number of possible cutter schedules included in the model. The result is a model containing columns of good potential. The optimizer then acts as an efficient search procedure for schedule selection. Finally, the approach has the important virtue of being robust and is widely applicable. Extensions to other scheduling problems are now under investigation and development.

4. SCHEDULE DIAGNOSIS AND ANALYSIS

The use of heuristic methods for schedule construction, that is, in the form of expert system rules, was initially considered as a parallel effort to the development of a mathematical programming formulation. It became rapidly apparent that, in the cutter application, the mathematical programming approach would yield optimal (or near-optimal) schedules within the limits of available computing resources. Since heuristic methods were unable to provide guaranteed optimal solutions (or measures of suboptimality) and appeared almost as computationally expensive, we thus abandoned our investigation of heuristic methods for schedule construction.

The mathematical formulation cannot capture all the rules that satisfy all the scheduling requirements. Hence, the system has been given the capability to diagnose the generated schedules. Using rules which have been provided by experienced schedulers, the system will inform the user of potential problems within a particular cutter schedule. In this way, the system can produce coherent, complete and timely schedules for the major cutters.

As is typically the case in complex resource-constrained environments, it is virtually impossible to meet all constraints and goals when developing a cutter schedule either manually or automatically using a mathematical program. The critiquing capability helps to construct an initial schedule by identifying outstanding commitments, and provides a critical evaluation of an existing schedule. The difficulty of creating and maintaining the schedule is largely reduced, particularly for officers rotating into new scheduling officer billets. Examples of problems detected by the schedule critic include:

- cutters overdue for maintenance or crew training
- insufficient number of cutters assigned to critical patrol localities
- inappropriate assignment sequences, e.g., a training assignment followed by a maintenance period
- cutters spending two consecutive Christmases on patrol
- incorrect assignment durations

The diagnostic program checks for activities that must occur on a periodic basis, e.g., training or maintenance, past due assignments, and constraints such as minimum or maximum durations for required assignments. In addition, the program must identify periods in which shortages occur in the targeted number of cutters assigned to a particular activity. Conversely, too many cutters may be assigned to the same activity at one time. In this case,

the amount of excess cutters and the durations involved must be reported.

Situations arise which require human intervention after schedule construction. For example, it is possible that the generated schedule could contain two contiguous, identical assignments. Human judgement would be required to decide whether or not to combine these assignments into a single assignment. Therefore, the diagnostic program must find all instances of this nature and allow the scheduler to make the appropriate modifications, if desired.

The mathematical program can handle dates or acceptable time frames for yards, docksides, training cycles, last years days away from home port, target days away from homeport, etc. However, certain special operations such as military exercises, cadet cruises and some types of cooperative exercises cannot be accounted for during schedule construction. Common sense reasoning must be applied to many situations. For instance, trading in lesser capable cutters because of higher priority jobs for the more capable cutter at some location and time.

The needs of the crews are reflected in workload balancing, maximum cruise lengths, minimum times to be spent in homeport, and other items that affect the crew's "morale." These morale-related limitations result in goals that the scheduler tries to satisfy. The evaluation and comparison of two different schedules that satisfy the mission requirements and morale-related goals in different balances can not be performed using strictly objective criteria. Judgement and individual preferences are involved.

In the early stages of this project a rule-based expert system shell was used to perform the diagnosis of generated schedules. However, to improve performance of the critic, the rule-based evaluation was converted to a procedural form in the LISP code. The resulting loss of flexibility (for adding or deleting evaluation rules) was offset by a more than 90% reduction in the time required to complete a critical evaluation of the schedule. This experience is typical in a rapid-prototyping project: as system requirements become well-defined after user experience through several prototype evolutions, implementors can sacrifice design flexibility for more carefully tuned designs yielding higher performance.

6. DISCUSSIONS

Considering the large potential for increasing operational productivity and the cost reduction involved in improving the quality of the scheduling process, the U.S. Coast Guard Research and Development Center embarked on a research project two years ago to automate various scheduling functions. The general goal was to build prototypes for each scheduling authority in which the computer system worked independently of explicit direction to help the user in scheduling a variety of platforms. Two prototypes of this nature have been installed in both Area offices which are the scheduling authorities for the high and medium endurance cutters and all the aircraft assets. The prototypes were designed in an evolutionary manner according to user requirements.

The Cutter Scheduling Assistant Program (CSAP) is providing Coast Guard schedulers with a powerful software tool to expedite the cutter scheduling process. In particular, cutters are assigned to various localities within an area to perform a variety of missions. The cutters are subject to availability of yards, docksides, and other maintenance facilities or requests such as "show and tell" for special occasions, e.g., holiday festivities and patriotic meetings. Many constraints are imposed on the schedule, such as military exercises, refresher training, long cruises, etc.

Each cutter's capabilities can be quite different and there are certain dependencies that exist among cutters that prevent treating this problem as a simple scheduling problem assigning independent, identical cutters to independent, identical tasks. For instance, one cutter may be preferred over another due to its special capabilities. Another complication arises because certain cutters are to be allocated together in a given place at a given time based upon their ability to cooperate in the performance of some mission task(s). Moreover, the scheduler must try to balance the workload democratically, e.g., all winter patrols for one crew and holiday weather for others would be considered unfair to all but the most hearty sailor.

In a reactive scheduling mode, CSAP helps in the design of a robust (reliable) cutter schedule, as opposed to one that could be considered optimal according to naive criteria. CSAP starts with a given cutter schedule and tests its feasibility; if a given set of schedules is found to be infeasible, the system offers interactive or automatic procedures to change the given schedules until they are feasible. The detailed problems may have occurred for several reasons such as incompatible requirements specification, manual amendments, etc. If an individual problem is considered minor then the schedule may well be acceptable operationally since it is frequently impossible to satisfy each requirement, request, and desire, simultaneously.

CSAP automatically collects statistical data for the number of days away from home port and underway days for each cutter. In addition, the total number of cutters assigned to a particular district each day is automatically updated and displayed to the user as assignments are added or deleted. Soon, an ad-hoc query capability will allow the user to gather statistical data and other information that is not accounted for in the hard-coded CSAP graphical interface, but is contained within the data stored as a result of building cutter schedules with CSAP.

For the presentation of output, we continued the Coast Guard tradition of using the well-known Gantt chart to represent cutter assignments. The current CSAP prototype runs on an IBM compatible personal computer requiring at least 12 MB extended memory and 1 MB of free space on the hard disk. Additional capabilities include a variety of schedule hardcopy options, statistical reports and displays, daily accounting of cutters assigned to critical patrol areas, and alternate views of the schedule. Coding of the interface was substantially accelerated using an expert system graphics interface. The expert system shell provided a library of menu building and control routines that can be called from LISP.

CSAP is being used routinely by the U.S. Coast Guard to schedule a several large cutters in both the east and west coasts. Significant reductions in professional staff hours and the higher quality of the schedules produced using the system have provided ample evidence that CSAP will soon pay for its development costs. In addition, CSAP is a general-purpose system that can be used along with analytical models to help with the detailed investigation of alternative operating policies thereby providing a valuable planning and research tool for strategic analysis. By limiting a great deal of the 'experimentation' to the computer, it has the potential to reduce the cost of implementing new strategies, both direct dollar cost and indirect cost, involving Coast Guard personnel and the public.

Although the current CSAP prototype is designed specifically for cutter scheduling at the fleet level, much of it is directly applicable to similar resource scheduling applications, such as district level scheduling of cutters and patrol boats, aircraft, and crews. This flexibility has been achieved through an object-oriented design/implementation that decouples the display manipulation of objects (cutters, assignments, etc.) from their internal representation for procedural manipulation.

As the CSAP system moved from the laboratory to become a fielded prototype, performance (response time) has become an issue of increasing importance. This "need for speed" was most readily apparent in the earlier versions of the rule-based schedule critic. For example, the earliest versions required over an hour to critique a one-year schedule. Although this was not regarded as a critical deficiency (pun intended), the slow turnaround adversely

affected the prototype's usefulness as a tool for the "what-if" tasks essential in a decision support role.

The CSAP prototype has been fielded since late 1990 as a standalone monolithic system (in several versions). Current (1992) prototyping efforts are focused on moving the schedule data out of CSAP and into a relational database system. This will provide more flexible access to the data, such as for ad hoc queries, customized text reports, an interface to the schedule generation and optimization modules, and potential future extensions.

Even with the improved execution speed gained over several prototype evolutions, the current (mid-1992) CSAP prototype provides only marginally acceptable response times. The CSAP hardware platform (80386-based personal computer) and software environment (MS-DOS, Microsoft Windows, and expert system shell) all present various constraints on improving this performance. We expect these constraints to become increasingly binding as we incorporate database links and the aforementioned schedule generation modules into CSAP. Accordingly, the Coast Guard is investigating more suitable environments (e.g., RISC-based UNIX workstations, X11 or Motif windowing systems, Common LISP) for a next-generation version of the CSAP prototype.

In summary, CSAP, incorporating the most important of the mission requirements and morale-related goals, is generating potential working schedules using a reasonable amount of computer resources. These potential schedules present the decision makers and schedulers with more choices for objective and subjective evaluation than the old manual method. Soon, this managerial tool will be applied to strategic and tactical decision processes, and is already improving these processes by relieving a heavy clerical workload.

6. REFERENCES

1. Darby-Dowman, K., Hajian, M., Lucas, C., Mitra, G., and Smith, J., Solution Strategies for Zero-One Programming with a Class of Vehicle Scheduling Problem, Presented at EURO XII/TIMS XXXI, Helsinki, Finland, 29 June-1 July 1992.
2. Darby-Dowman, K., and Mitra, G., An Extension of Set Partitioning with Application to Scheduling Problems, European Journal of Operational Research, Vol 21, pp 200-205, 1985.
3. Hoffman, K.L. and Padberg, M., Solving Airline Crew Scheduling Problems by Branch and Cut, Working Paper, George Mason University, April 1992.
4. L. Kingsley and J. Smith, A Scheduling Support System for the U.S. Coast Guard, In proceedings of conference on Artificial Intelligence Applications for Military Logistics, American Defense Preparedness Association, Fort Magruder Inn, Williamsburg, Virginia, 27-30 March 1990
5. Marsten, R.E. and Shepardson, F., Exact Solution of Crew Scheduling Problems using the Set Partitioning Model: Recent Successful Applications, Networks Vol II, pp 165-177, 1981.
6. Mitra, G., Darby-Dowman, K., Lucas, C., Moody, S. and Smith, J., Extended Covering and Partitioning Models for Scheduling in Industry, presented at the 1992 IBM Europe Institute, Oberlech, Austria, 3-7 August 1992.

END

DATE
FILMED

6 / 29 / 93