Interfaces

Publication details, including instructions for authors and subscription information:
http://pubsonline.informs.org

A General Examination Scheduling System

Michael W. Carter, Gilbert Laporte, John W. Chinneck,

To cite this article:
Michael W. Carter, Gilbert Laporte, John W. Chinneck, (1994) A General Examination Scheduling System. Interfaces 24(3):109-120. http://dx.doi.org/10.1287/inte.24.3.109

A General Examination Scheduling System

MICHAEL W. CARTER
Department of Industrial Engineering
University of Toronto
Toronto, Ontario M5S 1A4, Canada

GILBERT LAPORTE
École des Hautes Études Commerciales de Montréal
5255 avenue Decelles
Montréal, Quebec H3T 1V6, Canada

JOHN W. CHINNECK
Department of Systems and Computer Engineering
Carleton University
Ottawa, Ontario K1S 5B6, Canada

Scheduling examinations is a problem in virtually every high school, college, and university. The basic challenge is to schedule examinations over a limited time period so as to avoid conflicts and to satisfy a number of side constraints. Over 10 years, we have developed EXAMINE, a robust and flexible examination scheduling system suited to the needs of most educational institutions. EXAMINE runs on a PC and is fast and user-friendly. It provides a choice of feasible schedules and solutions in which examinations are well spread out for most students. It handles requirements regarding the proximity of a student's exams, room or time availability, and other factors. Over the past two years, the University of Toronto's engineering faculty and Carleton University have collaborated with us to develop a software package, which has been implemented at both institutions.

Scheduling examinations is a problem in virtually every high school, college, and university. The basic challenge is to schedule examinations over a limited time period so as to avoid conflicts and to satisfy a number of side constraints. Papers on this subject have appeared periodically over the past 30 years. Many have concentrated on graph theoretical and integer linear programming models and algorithms [Mehta 1981; Peck and Williams 1966; Tripathy 1984; Welsh and Powell 1967]. In

0091-2102/94/2403/0109$01.25
This paper was refereed

EDUCATION SYSTEMS—OPERATIONS
COMPUTERS—SYSTEM DESIGN/OPERATION

INTERFACES **24:** 3 May–June 1994 (pp. 109–120)

addition, several authors have developed heuristic algorithms capable of handling a number of the secondary features of the problem [Broder 1964; Cole 1964; Desroches, Laporte, and Rousseau 1978; Hertz 1991; Johnson 1990; Kiaer and Yellen 1992; Laporte and Desroches 1984; White and Chan 1979; Wood 1968]. The survey by Carter [1986] covers the most significant work carried out before 1986. Carter observed that the field is disorganized: several authors have developed algorithms to meet the requirements of one particular school, and the majority seemed unaware of the existence of most of the published literature.

We developed EXAMINE, a robust and flexible examination scheduling system suited to the needs of most educational institutions. The algorithm has evolved from the early work of Laporte and Desroches [1984]. EXAMINE runs on a PC, following a trend away from mainframe programs to more user-friendly micro-computer systems [Romero 1982].

Constraints and Objectives

The main difficulty in examination timetabling is to obtain a conflict-free schedule within a limited number of time periods and under room availability constraints. It is well known that producing a solution without conflicts, ignoring side constraints, is a graph coloring problem, and it is therefore not surprising that considerable effort has been devoted to the study of that particular graph theory problem. Most algorithms seek to avoid first order conflicts, that is, students having more than one examination scheduled at the same time. Laporte and Desroches considered the more general problem of avoiding higher order conflicts, that is, students having two examinations in two consecutive periods, or two examinations separated by one, two, three, or four free periods. In attempting to minimize such conflicts, the algorithm tends to spread out each student's examinations more evenly. In EXAMINE, we have generalized this notion: Students should not write x or more examinations within any y consecutive time periods.

Undesirable patterns can either be prohibited, that is, treated as hard constraints, or penalized through the use of weights in the objective function. The philosophy behind EXAMINE is to let the user decide how these should be handled and which weights are appropriate. Another requirement encountered in practice is that certain periods may be prohibited or undesirable for some examinations. Again, this is handled in EXAMINE as a constraint or by using penalties, at the user's discretion. Finally, EXAMINE can handle two additional types of constraints:

(1) For every period, the maximum number of students and examinations may be specified.

(2) Certain examinations require special facilities, such as laboratories or equipment. The user can specify limits on the availability of such facilities for each period.

Algorithm

Formally, the examination scheduling problem can be modeled as an integer program. Such formulations are generally of large scale and frequently exhibit nonlinearities, so that they are usually intractable by means of exact optimization techniques. We therefore propose a heuristic algo-

rithm, having four properties:
(1) The algorithm is robust: it produces a feasible solution (provided one exists) on a wide range of problems.
(2) It is flexible enough to handle the variety of constraints encountered in real-life problems.
(3) It is quick: it runs on a personal computer, and users are able to execute several runs within reasonable time.
(4) It is user friendly: easy to learn, easy to use, and helpful.

Basically, the algorithm gradually assigns examinations to periods in order to optimize an objective function that reflects the quality of the schedule. Strict constraints are taken into account by prohibiting illegal assignments, and softer requirements are handled through the objective function. Specifically, a cost is associated with each type of undesirable condition. This cost is multiplied by the number of students who have the condition, and all such costs are accumulated to give the total cost of the schedule. The user controls the costs completely to reflect the particular institution's assessment of the importance of each violation. When no legal period is available for a particular examination, backtracking is necessary: the algorithm must remove examinations that have already been assigned to make room for this exam and then try to reschedule the displaced examinations into new periods. To ensure that the algorithm converges, we limited backtracking using an approach similar to that used in tabu search methods [Glover 1989, 1990]. Once it has scheduled all examinations, the algorithm executes a postoptimization phase. Laporte and Desroches [1984] give the details of the original underlying principles. EXAMINE contains two important features added since then:
(1) Initial assignment: To ease the assignment process, the program handles the examinations that are deemed the most difficult to assign first. It constructs an incompatibility graph first and determines a maximum *clique* (the largest set of mutually conflicting examinations) by means of the heuristic described by Carter and Gendreau [1992]. These examinations are then assigned to different periods. For the remaining assignments, it considers examinations in nondecreasing order of the number of available periods remaining [Brelaz 1979].
(2) Maximum number of examinations per student within a given time frame: Many schools have instituted rules of the form, "No student shall write x or more examinations in any y consecutive periods," or the relaxed version "Minimize the number of occurrences of a student writing x or more examinations in y consecutive periods." Like most timetabling algorithms, EXAMINE will repeatedly evaluate the feasibility and the cost of assigning a particular exam i to a specific period p. In order to implement rules of the form "No x in y," it is necessary to know exactly which students are writing each exam. The original Laporte and Desroches paper considered only the number of students in common between each pair of exams. In EXAMINE, we have developed efficient methods for

A cost is associated with each type of undesirable condition.

computing the set of students in common among the exams in any group. When we try to assign an exam to some period p, we can identify the precise students that will be affected and the associated cost.

Computer Implementation

EXAMINE is coded in C and operates on an IBM Personal Computer. It is interactive, uses simple commands and makes use of help screens. One important feature is that EXAMINE gives the user the ability to produce several alternative schedules. Indeed, although EXAMINE always generates a locally optimal timetable with respect to given penalty values, users generally prefer to be given a number of schedules to choose from. The user can also fix a subset of examinations and let the program schedule the remainder.

The program generates several reports to assist the registrar's operations:

(1) A full schedule, listing all examinations, number of students, and room numbers for each period;

(2) Partial schedules, for example, for core courses, for particular programs, or for groups of students;

(3) An alphabetical or numerical list of courses with enrollments, examination times, and room numbers; and

(4) A room utilization report.

In addition, the system includes an interactive feature that allows the user to modify the completed schedule, in response to last minute unforeseen circumstances. For any specified examination, the system will produce a list of all periods to which the exam could be moved without creating conflicts or violating room capacities. The user may then move the exam into any chosen period.

The University of Toronto Engineering Faculty

The University of Toronto is the largest university in Canada with over 50,000 students. Unlike most smaller institutions, Toronto has a very decentralized administrative structure; each faculty has its own registrar and schedules its own courses and examinations. The faculty of arts and science has over 20,000 students. It administers about 200 examinations in the fall and summer terms and over 800 in the spring. (Many of its courses are eight months long and have no fall term examination.) The faculty of applied science and engineering has about 2,400 undergraduate students taking nearly 200 engineering courses each term. Engineers also take a large number of arts and science courses as electives. In the spring term of 1991, for example, they took 128 courses from A and S.

Because of its size, the faculty of arts and science always produces its examination timetable first. The engineering faculty then attempts to schedule its examinations around arts and science course conflicts. (To our knowledge, no one has ever attempted to centralize the timetabling function.) The timetable for engineering covers 10 working days (Monday to Friday over two weeks), two periods per day (morning and afternoon). Each examination is two and one half hours long with a two and one half hour break between morning and afternoon exams.

Prior to the fall of 1989, the engineering faculty scheduled examinations using a computer-assisted algorithm that was developed in-house. Briefly, the method started by using the previous year's timetable groupings. In particular, each period

from the previous schedule consisted of a group of exams that were conflict free at that time. Several events made the last year's groups unacceptable to the subsequent year: students took different combinations of courses creating conflicts within groups, new courses were added, and old courses were dropped. The program notified the user of conflicts. He or she would then try to move courses around into different groups to eliminate or reduce the conflicts. Once the new groups were established, the program tried to assign them to periods, taking into account arts and science conflicts, and tried to rotate the groups so that the same exams were not always late in the schedule.

The method was never totally successful, and there were always a dozen or so students who had direct conflicts (two exams in the same period). These students were assigned to a special clash room where they would begin their first exam before their classmates and then start the second one after a short break. This arrangement was not very popular with students, but it appeared unavoidable. When the EXAMINE system was implemented in 1989, all direct conflicts were eliminated.

Table 1 summarizes the examination timetabling rules for engineering. To implement Rule 2, we used the original objective function suggested by Laporte and Desroches [1984] with a few variations. Laporte and Desroches recommended the following: two exams in two consecutive periods cost 16, two exams in three periods cost 8, two exams in four periods cost 4, two exams in five periods cost 2, two exams in six periods cost 1, no direct conflicts are allowed. The effect of this structure is that student exams tend to be evenly distributed over the whole examination schedule. In engineering, there are only two exams per day. However, it is clearly worse for a student to have two exams on the same day than it is to have an exam on Monday afternoon followed by another on Tuesday morning. To capture this preference, we inserted a dummy evening time period solely for the purpose of calculating costs. Therefore, in a perfect timetable, every student's exams would be two days apart. It is also clear that an exam on Friday afternoon should have no effect on the Monday morning assignments in the following week. Therefore, we introduced more dummy periods on Saturday and Sunday to reflect this notion. This is now the default period system used in

Rule 1:	No two exams at once (outright conflicts).
Rule 2:	Spread each student's exams as evenly as possible.
Rule 3a:	All first- and third-year exams in morning periods.
Rule 3b:	All second- and fourth-year exams in afternoon periods.
Rule 4:	In the fall term, the six first-year exams (common first year) must be preassigned and perfectly spaced, one exam every two days, to give these students a break.
Rule 5:	In the fall term, the six first-year exams (with 700 students in each class), must be rotated each year so that the same exam is not always last, to give the professors a break!

Table 1: A summary of the Toronto University scheduling guidelines: These rules exhibit some unusual properties along with the normal preferences.

EXAMINE.

Notice that the rules simplify the problem by dividing the students into two groups: those with exams in the morning (10 periods) and those with exams in the afternoon (10 periods). In particular, since students have only one examination per day, there is a virtual guarantee of good spacing for everyone. Of course, there are many third-year students taking fourth-year courses, and so forth, but the normal student should have a good schedule. Furthermore, the faculty decided to give an extra break to first-year students in the fall term. Since they all take the same six courses, they can be preassigned to every second day to give them a perfect spread: one examination every two days.

Toronto has a very decentralized administrative structure.

Unfortunately, there is also a chronic shortage of suitable examination rooms. Students need a decent workspace where they can spread out their papers. Lecture theatres with small arm rests are not acceptable exam rooms. There are over 1,000 seats in suitable classrooms, but the faculty also insists that two different examinations cannot be scheduled simultaneously in the same examination room. Therefore, an exam for five students will occupy a room that holds 30, and the extra 25 seats will be lost. For this reason, the scheduler estimates that we can normally schedule only about 850 seats; there must be a 15 percent unused capacity.

The lack of space creates two very serious problems. Consider the schedule for the 10 morning periods with six periods preassigned to first-year students (700 of them) on Monday, Wednesday, and Friday. This leaves very little space on those days for the third-year students, and only four days really open to them, on Tuesdays and Thursdays. Moreover, third-year students tend to have a large number of elective courses, so they need far more examination days than the first-year students. To obtain a conflict-free examination schedule, the third-year students need the full set of 20 available periods instead of four.

Using the EXAMINE system, we were able to convince the associate dean that the morning and afternoon division of exams was simply a bad idea. We ran schedules with the forced separation and new schedules in which the second-, third- and fourth-year exams could move freely. (First-year exams were still fixed.) The new results were so much better that we were able to sell the idea to the faculty and change a long-standing practice. In 1989, this had one tragicomic effect. Many students and a few faculty members were so used to the old morning/afternoon split that they never bothered to read the time on the examination schedule that was posted; they just read the day and assumed that second- and fourth-year exams were always in the afternoon. More than a few students missed their morning exams. Fortunately, the reverse error just meant that people showed up too early.

We were also able to convince the faculty to let the first-year exams float a little. Instead of preassigning the first exam to Monday morning, we preassigned it to one

of the four periods on the first Monday or Tuesday. It is much better to let the system determine the best possible place for each examination. The large size of these courses and the underlying cost structure of EXAMINE automatically gives a high priority to evenly spacing the first-year examinations. The faculty still insisted on forcing a specific sequence of the first-year exams, so that each exam is scheduled last only once every five years. (The one humanities elective course is always scheduled third.)

The shortage of acceptable examination space created a second major problem. Under the old system, the scheduler would post the preliminary timetable in the various departments and then try to accommodate requests for changes. There were always several requests, since the timetable contained conflicts. After the deadline to make changes, the timetable was considered fixed. At this point, the scheduling officer would attempt to manually assign the examinations on each day to acceptable rooms. This frequently resulted in an eleventh hour panic with no space for examinations that were required to occur on the scheduled day and time. In response to this, we built a room assignment module as part of the EXAMINE system. During the examination timetabling process, a limit is placed on the number of examinations and on the number of students that can be assigned to any given period. We then try to automatically assign examinations to rooms. The algorithm is fairly simple in that it basically attempts to assign the largest class to the largest available room. However, there are several side constraints. In particular, when an examination is too large to fit in any room, it must be assigned to two or more adjacent rooms, so that the professor can easily move around between the various locations to answer questions and ensure that everyone is treated fairly. Our algorithm asks the scheduler to define valid room combinations and then uses only these groupings and, of course, never uses the same room twice. The most important contribution of the room assignment module is that the scheduler now knows if there is a feasible room assignment solution before the preliminary timetable is posted. If the program cannot find an acceptable solution, the scheduler can move some of the examinations manually, or change the limits on the number of students per period, and run a completely new timetable. (EXAMINE also includes a random number feature so that the user can construct many different timetables with the same parameters and select the best one.)

Students with direct conflicts were assigned to a clash room.

Beginning in 1989, the scheduling officer was able to post the preliminary timetable and tell people: "We have a conflict-free schedule. Therefore, virtually no changes will be allowed." Indeed, no changes are necessary. If professors have special requests, they are required to report them before the schedule is run. EXAMINE is flexible enough to handle almost any special situation easily. The new system has changed the administrative aspects of examination timetabling from a dreadful chore to a fairly simple, automated pro-

cess. The scheduler still has to deal with professors' special requests by telephone, but the post-schedule volume is down to a trickle.

The original version of EXAMINE implemented in the fall of 1989 was unfortunately not very user-friendly. It had been developed by several students and programmers, and the current student expert had to be physically present to run the timetable. We could not just give it to the scheduling officer. Since then, the system has become user-friendly, easy to run, and helpful. We have now developed a version that is basically load and go. The system uses a full display to guide even a novice user through each stage. There is a built-in help facility to provide explanations. The 1992 version of EXAMINE is truly simple to use.

Carleton University

Carleton University, located in Ottawa, has about 21,000 students in some 50 disciplines grouped into four faculties The student population has grown rapidly over the past decade, causing acute problems in scheduling the academic year. The fall term is the most difficult: between Labor Day and the beginning of the Christmas holidays, the university must find time for orientation and other first-week activities (registering and paying fees), 13 weeks of lectures, a grace period before exams, and the examinations themselves. In particular, the number of days the university needs to schedule the examinations has increased as the student body has grown. In the 1990–91 academic year, the university was forced to adopt a 12-week lecture period to allow enough time for the examinations to take place. However, in 1991–92, the 13-week lecture period was reinstated, beginning directly after Labor day in the fall term.

The havoc wreaked by these sudden changes in the teaching schedule led the vice-president (academic) to form a task force on academic year scheduling. One of the duties of the task force was to "consider, evaluate, and recommend on, appropriate computer software that might provide assistance to the university in the area of scheduling." One of us (Chinneck) served on the task force. After a survey of other universities in Ontario, the task force concluded that 12 days of examinations should be sufficient. Unfortunately, the current exam-scheduling software was generating schedules of 15 days in the fall, and 17 days in the winter (1990–91).

The software currently in use at Carleton was developed in-house many years ago and uses an unsophisticated one-pass approach. It schedules exams one by one, from largest to smallest enrollment. It finds a time for the exam currently under consideration by trying it in all possible exam periods and calculating "penalty points" for each possible exam period. The exam is scheduled in the period giving the lowest penalty points and as early as possible. Penalty points are assigned for violation of the scheduling rules. Exams that cannot be scheduled without breaking some of the rules are reported. The user must then adjust the penalty-point parameters that guide the scheduler and begin again. In practice, the user runs the software repeatedly, adjusting the penalty points each time, until an acceptable schedule is achieved. This is a tedious process.

In addition to using scheduling software

Rule 1:	No two exams at once (outright conflict).
Rule 2:	No x exams in x periods, where $x \geq 3$.
Rule 3a:	No four or more exams in five periods.
Rule 3b:	No five of more exams in seven periods.
Rule 4a:	Single section evening courses (all faculties) must be scheduled in an evening exam slot.
Rule 4b:	Multi-section course (arts and social sciences faculties): evening sections must be scheduled in evening exam slots, day sections can be scheduled in day or evening exam slots.
Rule 4c:	Multi-section courses (all other faculties): all sections must be scheduled together, either day or evening.
Rule 5a:	Penalize (but do not forbid) "two in two."
Rule 5b:	Penalize (but do not forbid) "two in three."
Rule 6:	Preschedule courses taught on instructional television (ITV). These are placed later in the examination schedule to allow the maximum amount of time for tapes of the final classes to reach ITV students.
Rule 7:	A maximum of 1,550 suitable seats are available simultaneously.

Table 2: A summary of the Carleton University scheduling guidelines: Evening classes and instructional television courses generate many difficult constraints.

that had become inadequate for the larger number of exams, Carleton used examination scheduling rules (Table 2) that were rather restrictive, making the exams hard to schedule. There are three exam periods per weekday, and two exam periods on Saturdays (three if necessary). No exams are scheduled on Sundays.

The task force asked one of us, Michael Carter, to run some experiments with EXAMINE using actual examination data for the 1991 examinations. For comparison purposes, these exams had already been scheduled using the current software (requiring 15 exam days in the fall and 17 in the winter). EXAMINE allows easy testing of scenarios: we examined various scenarios of numbers of seats available and inclusion or exclusion of some of the guidelines in Table 2. A test result that piqued the interest of the task force was the observation that it was the instructional television restrictions in particular that were forcing the exam schedule over 12 days. This insight alone was a valuable contribution.

An early version of EXAMINE (that was not able to handle "No x in y" rules) successfully generated 12-day exam schedules for both the fall and winter terms, with only small numbers of students having illegal exam patterns. For the fall 1990 data set with 50,892 student exams, only 42 students had serious rule violations (rules 2 and 3). For the winter 1991 data set with 57,371 student exams, only 93 students had serious rule violations, but 1,650 seats were required. The examinations officer indicated that the extra seating would not be difficult to find. The task force thought these to be excellent results and began to consider whether an institutional way could be found to deal with the students who had illegal patterns, such as setting second exams or cloistering the students involved. The task force also noted that

more than 12 days are actually available for examinations in the winter term, which would reduce the number of illegal patterns substantially.

At the same time, in response to the Carleton results, Mike Carter began a research program to incorporate "No x in y" rules of the Carleton type directly into EXAMINE. Over the next couple of months, he generated improved results for the two Carleton data sets. By March 1992, he had constructed a fall 1990 schedule that satisfied all constraints using 12 periods with a maximum of 1,667 seats required. (Of the 36 exam periods, 14 required more than 1,635 seats.) In particular, one exam was to be given to 1,667 students enrolled in a course! Again, it was possible to come up with the additional required seats. The scheduler was also able to construct a solution for the winter 1991 data satisfying all constraints in 13 days without ever exceeding the 1,550 seat limit. The 12-day limitation on the length of the exam period is more critical in the fall than in the winter, and the task force thought that 13 days would after all be quite acceptable.

Carleton now uses it as the main exam scheduler.

Meanwhile, the task force submitted its report in October 1991, recommending that Carleton acquire the EXAMINE program. An important recommendation in the task force report concerning ITV courses is directly due to the EXAMINE experiments. The report was adopted unanimously by the senate of the university in November 1991. Carleton has used EXAMINE in parallel with the current software to generate the exam schedule for the winter term 1992 and now uses it as the main exam scheduler.

Conclusion

Between 1980 and 1992, we developed a general examination scheduling system that is of wide applicability. The special characteristics of EXAMINE contributed directly to its adoption by a number of institutions. Foremost among these is its flexibility in allowing various rules and weights to be expressed and modified to create different scenarios.

We can also draw lessons about selling OR tools from our experiences with EXAMINE at Toronto and Carleton. First, nothing convinces a client organization better than experience with its own exact problem. Second, a new tool will be adopted if it either produces a better solution than the existing tool or provides insight into the problem that is not provided by the existing tool. EXAMINE did both. It helps to close the sale if a tool can be used in three modes, as EXAMINE was: as an analysis tool to give insight, in a dry-run mode to convince clients, and finally as a full-scale production tool.

A number of universities have already adopted EXAMINE, and the system is being implemented in various institutions worldwide, including the London School of Economics and Otago University in New Zealand.

Acknowledgments

This work was partly supported by the École des Hautes Études Commerciales de Montréal, by the Canadian Natural Sciences and Engineering Research Council (grants OGP0001359 and OGP0039682),

and by a Quebec-Ontario cooperative research grant. We express our gratitude to the following students and programmers who have worked on EXAMINE over the years: Guy Brousseau, Sylvain Desroches, Khaled El Sahli, Sau Yan Lee, Jennifer Lobo, Hayley Ohlig, and Victor Sitsch.

This work has been awarded the 1991–92 Practice Prize by the Canadian Operational Research Society.

References

Brelaz, D. 1979, "New methods to color the vertices of a graph," *Communications of the A.C.M.*, Vol. 22, No. 4, pp. 251–256.

Broder, S. 1964, "Final examination scheduling," *Communications of the A.C.M.*, Vol. 7, No. 6, pp. 494–498.

Carter, M. W. 1986, "A survey of practical applications of examination timetabling problems," *Operations Research*, Vol. 34, No. 2, pp. 193–202.

Carter, M. W. and Gendreau, M. 1992, "A practical algorithm for finding the largest clique in a graph," *Publication CRT-820*, Centre de recherche sur les transports, Montréal.

Cole, A. J. 1964, "The preparation of examination timetables using a small-store computer," *The Computer Journal*, Vol. 7, No. 2, pp. 117–121.

Desroches, S.; Laporte, G.; and Rousseau, J.-M. 1978, "HOREX: A computer program for the construction of examination schedules," *INFOR*, Vol. 16, No. 3, pp. 294–298.

Glover, F. 1989, "Tabu search, Part I," *ORSA Journal on Computing*, Vol. 1, No. 3, pp. 190–206.

Glover, F. 1990, "Tabu search, Part II," *ORSA Journal on Computing*, Vol. 2, No. 1, pp. 4–32.

Hertz, A. 1991, "Tabu search for large scale timetabling problems," *European Journal of Operational Research*, Vol. 54, No. 1, pp. 39–47.

Johnson, D. 1990, "Timetabling university examinations," *Journal of the Operational Research Society*, Vol. 41, No. 1, pp. 39–47.

Kıaer, L. and Yellen, J. 1992, "Weighted graphs and university course timetabling," *Computers and Operations Research*, Vol. 19, No. 1, pp. 59–67.

Laporte, G. and Desroches, S. 1984, "Examination timetabling by computer," *Computers and Operations Research*, Vol. 11, No. 4, pp. 351–360.

Mehta, N. K. 1981, "The application of a graph coloring method to an examination scheduling problem," *Interfaces*, Vol. 11, No. 5, pp. 57–64.

Peck, J. E. L. and Williams, R. L 1966, "Algorithm 286—Examination scheduling," *Communications of the A.C M.*, Vol. 9, No. 6, pp. 433–434.

Romero, B P. 1982, "Examination scheduling in a large engineering school: A computer assisted participative approach," *Interfaces*, Vol. 12, No 2, pp. 17–24.

Tripathy, A. 1984, "School timetabling—A case in large binary linear ınteger programming," *Management Science*, Vol. 30, No. 12, pp. 1473–1489.

Welsh, D. J. A. and Powell, M. B. 1967, "An upper bound for the chromatic number of a graph and its application to timetabling problems," *The Computer Journal*, Vol. 10, No. 1, pp. 85–86.

White, G. M. and Chan, P. W. 1979, "Towards the construction of optimal examination schedules," *INFOR*, Vol. 17, No. 3, pp. 219–229.

Wood, D. C. 1968, "A system for computing university examination schedules," *Computing Journal*, Vol 11, No. 1, pp. 41–47.

Note

This paper was awarded the first prize of $1,000 in the 1992 Practice Prize Competition of the Canadian Operational Research Society (CORS) held in Toronto on June 2, 1992. There are two basic purposes behind this competition, the first being that of recognizing outstanding applied OR work per se, and the other that of focusing attention on OR and its applications in Canada by attracting quality papers to the annual CORS national conference where the papers are presented before a jury. Contestants must be residents of Canada and include in their submission a letter by an ex-

ecutive of the client organization that sponsored the application attesting that this application truly had an impact on its operations.

Barbara McCann, Faculty Registrar, University of Toronto Faculty of Applied Science and Engineering, 35 George Street, Toronto, Ontario, Canada M5S 1A4, writes, "The Faculty of Engineering has been using the EXAMINE system since the fall of 1989. Examination timetabling in our faculty is a major problem every term. Nearly 200 exams have to be scheduled and this task had become quite unmanageable using our previous computer system. In particular, conflicts were unavoidable, and this created a considerable amount of dissatisfaction among students. As a result, once the preliminary timetable had been posted, we were asked to consider a large number of changes, and the situation had become untenable. With EXAMINE, all conflicts have been eliminated, requests for changes have virtually disappeared, and the entire process of building exam schedules is now almost painless. The EXAMINE system has had a great impact on the registrar's operations, and on students' satisfaction."

D. R. Gardner, Professor and Chairman, Department of Biology, Chairman, Vice-Presidential (Academic) Task Force on Academic-Year Scheduling, Carleton University, Ottawa, Ontario, Canada K1S 5B6, writes, "use of the EXAMINE computer program to analyze our examination scheduling problems had a major impact on the recommendations included in the final report submitted by the Task Force, especially on our conclusions about the length of the examination period. The task force report was accepted unanimously by the senate of the university, so the recommended changes to our scheduling procedures arising out of the EXAMINE experiments are now permanent. I note also that EXAMINE has now been transferred to Carleton, and it is expected that, subject to further tests during the current examination period, it will become the main examination scheduler for the university."