Journal of the Operational Research Society (2006) 57, 950–956

www.palgrave-journals.com/jors

A methodology to optimize foundation seminar assignments

KA Willoughby[1]* and CJ Zappe[2]

[1]*Health Quality Council, Saskatoon, SK, Canada; and* [2]*College of Arts and Sciences, Bucknell University, Lewisburg, PA, USA*

First-year students entering the College of Arts & Sciences at Bucknell University (USA) are required to enroll in a first-year experience course called a foundation seminar during their first semester. A few months before arriving at Bucknell, students submit a prioritized list of foundation seminars of interest to them, given course descriptions of all available foundation seminar sections. Then, based on capacity and scheduling constraints, each student is assigned to a particular seminar. Currently, this assignment of students to specific seminars is carried out using both manual and heuristic methods. We propose to apply an optimization methodology to this interesting real-world problem in an attempt to determine assignments that better satisfy the highest preferences of entering first-year students.

Journal of the Operational Research Society (2006) **57,** 950–956. doi:10.1057/palgrave.jors.2602070
Published online 21 September 2005

Keywords: linear programming; methodology; optimization; practice of OR

Introduction

Decision-makers use assignment models to optimize resource allocation. A variety of successful applications have been reported in the literature, including medical residents to hospitals (Roth, 1986), loaner equipment to customers experiencing downtime (Karmarkar and Kubat, 1983), Navy personnel to jobs (Liang and Thompson, 1987; Blanco and Hillery, 1994), law students to employment fair interview slots (Bartholdi III and McCroan, 1990), buyer–seller meetings to trade show appointment periods (Ernst *et al*, 2003), female collegiate students to sorority organizations (Mongell and Roth, 1991) and flight training resources to aeronautical students in various time blocks (Bazargan-Lari, 2004).

In this paper, we develop a linear programming model to optimize a series of assignment decisions encountered annually at Bucknell University. Founded in 1846, this private, residential institution educates about 3500 (nearly all undergraduate) students. One of its important curricular components is a first-year experience course called a foundation seminar. Over 700 students are admitted annually into the College of Arts & Sciences, and each of these students is slotted into one of roughly 45 available seminar sections. Foundation seminars are offered on a number of different interdisciplinary themes and are capped at 16 students.

A few months before arriving at Bucknell, students are asked to submit a prioritized list of their top 10 seminar choices. Up until 2003, the Registrar's Office manually assigned students to seminar sections, attempting to ensure that students received their highest preferences while maintaining capacity restrictions on section size. One staff member in the Registrar's Office was charged with the responsibility of making the section assignments. She had about 5 years experience in overseeing this particular process. To develop the assignments, she would begin by sorting the seminars into those that were highly popular (the 'oversubscribed' sections that received very high preferences from many entering first-year students) and those that were less so (the 'undersubscribed' seminars). She then examined each undersubscribed foundation seminar. Proceeding in alphabetical fashion, she would assign students to this class, beginning with all those that had given the particular section a #1 ranking, then moving on to those students that provided a #2 ranking, and so on. (Owing to the fact that these seminars were undersubscribed, it was highly doubtful that the class would be totally comprised of students who had each indicated it was their #1 choice.) The assignment of students to a section would halt when either capacity restrictions were reached or the staff member had scanned the entire student list (and room was still available in the class).

She then turned her attention to the oversubscribed sections, repeating the procedure as outlined above. The foundation seminar manual assignment process took about a week's worth of time for this single staff member.

Beginning in 2004 (for those first-year collegiate students who comprised Bucknell's class of 2008), the Registrar's Office undertook a collaborative effort with the Information Services & Resources Department to automate this assign-

*Correspondence: KA Willoughby, Health Quality Council, 241-111 Research Drive, Saskatoon, SK, Canada S7N 3R2.
E-mail: kwilloughby@hqc.sk.ca

ment process. A systems integrator, herself a Mathematics graduate with over 30 years of programming experience, created a heuristic procedure to assign students to sections as efficiently as possible. In her approach, students were processed in random order. They would be slotted into the first of their selections provided there was room in the course. If the course was full, then the procedure examined students already assigned to this course, searching for the student with the lowest preference ranking for an unfilled course ('re-assignment candidate'). This student would then be re-assigned to the unfilled course. If there were no students who could be slotted into available courses, then the heuristic would explore the original student's second choice, and so on. Discussions with the systems integrator revealed that such an approach mimicked (in automated fashion) the manual method as used by the Registrar's Office for several years. Moreover, this approach took about 30 min to complete the entire assignment procedure, a significant time reduction from that encountered earlier with the manual method.

Our optimization model was independently constructed in a parallel development to the collaborative efforts between the Information Services & Resources Department and the Registrar's Office. We were pleased to learn that the Registrar's Office was amenable to the use of quantitative models in helping to analyse this important set of decisions. In a way, this reinforced the need for an optimization focus on the current problem. We were eager to learn if we could develop such a model and, if so, how the results of such a modelling effort would compare to the assignment decisions made either manually or with the heuristic procedure.

The remainder of our paper proceeds as follows. The next section provides a review of pertinent literature, followed by the development of our optimization model. We then provide model results, after which we conclude the paper with some summary comments and directions for further study.

Literature review

Other researchers have explored the assignment of college students to courses. Early contributors to this problem were Macon and Walker (1966). Examining students in random order, their algorithm tallied the number of requests for a particular course. It then compared the amount of requests to the number of seats remaining in making its assignments.

Sabin and Winter (1986) developed a scheduling programme to assign 2600 first-year students to courses in the School of General Studies at Memorial University (Newfoundland, Canada). Their objectives were to provide better control of section sizes (courses had capacities of 45 students each) and offer a more equitable distribution of students into time slots. They assigned weights to each course to assess the degree of complexity associated with scheduling them. Higher-weighted courses were scheduled first and consisted of those with fewer sections or time slots available. The scheduling algorithm assigned students to classes by determining the sum of the individual course weights in a particular student's choices. More awkward schedules (those students with higher sums) were slotted first.

At the Ecole Polytechnique de Montreal, one of Canada's leading Engineering schools, Laporte and Desroches (1986) created an algorithm to allocate 1800 students to course sections. Their methods sought to respect student course selections, observe room capacities, and minimize the number of associated student movements between the two campuses, located about two miles apart.

Graves *et al* (1993) implemented an innovative auction method for course registration at the University of Chicago's Graduate School of Business. Their process permitted changes in registration by developing a market model (complete with clearing prices) that allowed students to bid on particular course schedules. This process now annually assigns 18 000 seats in about 475 course sections in four respective degree programmes.

Other researchers have addressed the coupling of student course assignment with the more complete problem of course scheduling. Using a local-search heuristic procedure, Sampson *et al* (1995) wanted to maximize the enrollment of students in their most preferred classes. Students initially provided a rank-ordering of their course selections. Using this information, the heuristic procedure then determined a complete schedule for each student. This was successfully implemented at the University of Virginia's Darden Graduate School of Business Administration.

Hertz and Robert (1998) decomposed the entire course scheduling problem into a series of subproblems, each consisting of a separate resource allocation. They assigned starting periods to lectures (timetabling), students to sections (grouping) and classrooms to sections (classroom assignment). Ferland and Fleurent (1994) developed a decision support system to handle the timetabling and grouping issues. In particular, their approach proved quite useful for courses involving large groups of students. Implemented at two Canadian universities (in Montreal and Sherbrooke), users reported that the final solutions provided by the system were better than those produced previously, plus generating the complete scheduling solution now saved plenty of time for university staff. Mirrazavi *et al* (2003) formulated a large-scale integer goal programming model to investigate a timetabling problem. In their specific approach, they analysed both the allocation of lectures to rooms and, secondly, start-times to lectures. The second assignment was optimally solved with a genetic algorithm. Solving an important timetabling problem for Greek high schools, Papoutsis *et al* (2003) used a column generation approach. They successfully accomplished a number of goals with their method, including the elimination of idle hours for the teachers.

Besides modelling the allocation of students to courses (and, in some cases, subsequently generating student schedules), a variety of other 'college-centred' assignment problems have been examined in the literature. Bloomfield and McSharry (1979) developed a heuristic approach to assign classes and course sections to particular days of the week, times of the day and specific rooms, paying special attention to the needs of faculty. For example, faculty stipulated that they be permitted certain non-teaching times or days during the week, as well as back-to-back scheduling if they taught multiple sections of the same course. This approach was successfully employed at Oregon State University's School of Business.

Mulvey (1982) created a trans-shipment network optimization model to assign classes to various time periods (slots) with the goal of maximizing the number of occupied seats. Constraints encountered in his model included classroom seating capacities as well as particular logical restrictions (eg a class could be assigned to exactly one slot, and faculty were could only be assigned to at most one class during a particular slot).

Using a priority system, Hojati and Hoang (1997) assigned classes to time slots during the week, and then subsequently assigned these classes to available classrooms to maximize instructor preferences and minimize student conflicts. Classes with the highest priority were scheduled first. These high-priority classes typically involved those for which an instructor specified a particular schedule pattern (days of the week to meet, starting time, specific room, etc).

Hinkin and Thompson (2002) developed a computer program called SchedulExpert for courses at Cornell University's School of Hotel Administration. Their approach automated the entire scheduling process with the objective of eliminating conflicts among core courses as well as electives within specific functional areas. College staff can now achieve results in a few hours, instead of a couple of weeks as was encountered under the previous manual system.

Other assignment models developed for collegiate operations include the assignment of proctors to final examinations at Carleton University in Ottawa, Canada (Awad and Chinneck, 1998). At this university, over 100 proctors are required to oversee exams during the hectic 15-day examination periods in the fall and winter semesters. These authors developed their assignment system using problem-specific heuristics and a genetic-algorithm framework. Čangalović *et al* (1998) used special-purpose heuristics and Tabu search to assign students to specific exam periods. Their model was then applied for 5000 students enrolled in over 60 subjects in the Law School of Belgrade University during their 30-day examination phases. Gosselin and Truchon (1986) considered the assignment of specific university rooms to particular courses. They determined the number of faculty requests of each type that ought to be met with rooms of certain categories.

Optimization model

We shall now describe the development of our model to optimize the assignment of incoming first-year students to various foundation seminar sections. For notational purposes, we will use i to represent students and j to denote seminars. Moreover, we note that most seminars offer a standard capacity of 16 students. As described earlier in this paper, such a class size permits greater interaction between instructor and student and helps to facilitate the student's transition to college-level learning. However, a few seminars each year feature enrollments of more than 16 students. Overriding the standard capacity may result due to a faculty member's willingness to accept a larger class size (this may be the case for especially popular sections), or when two or three instructors agree to teach concurrent sections of the same seminar. In this latter case, the Registrar's Office will permit relatively large enrollments into the class and then randomly apportion the students into the respective individual sections. In any event, it is rather uncommon for course enrollments to be more than 16 students. We examined two recent years of seminar assignments and found that around 10% of the classes featured enrollments in excess of standard capacity levels. We shall use the notation S_j to refer to the subset of courses offering standard capacities, while A_j shall denote those seminars that permit amalgamated enrollments greater than standard levels.

Our linear programming model has the following form:

Minimize

$$\frac{1}{m} \times \left[\sum_{i=1}^{m} \sum_{j=1}^{n} R_{ij} X_{ij} \right]$$

Subject to:

$$\sum_{j=1}^{n} X_{ij} = 1 \quad \forall i = 1, 2, \ldots, m \tag{1}$$

$$\sum_{i=1}^{m} X_{ij} \leqslant 16 \quad \forall j \in S_j \tag{2}$$

$$\sum_{i=1}^{m} X_{ij} = C_j \quad \forall j \in A_j \tag{3}$$

The decision variable, X_{ij}, is defined as follows:

$$X_{ij} = \begin{cases} 1 & \text{if student } i \text{ is assigned to seminar } j \\ 0 & \text{otherwise} \end{cases}$$

Although the decision variables have an integer formulation, we do not need to restrict them to 0–1 values when actually solving the model. They will naturally take on values of 0 or 1 due to the unimodularity present in our constraint matrix.

For an incoming first-year class size of, say, 740 students with 45 available seminars, our model formulation has in the neighbourhood of 33 300 decision variables with nearly 790 constraints. There is a single constraint for each student (assignment restriction) and seminar (capacity level).

Other terms in this model are:

R_{ij} = ranking that student i provided for seminar j
C_j = amalgamated section enrollment provided by either manual or heuristic methods.

For any student who failed to include a particular seminar in his/her prioritized top 10 list, we use an R_{ij} value of 99 999 (a very large positive number). Since the model seeks to minimize the objective function (by allocating students into their higher-preferred sections), this would it make very unlikely (if not impossible) for a student to be assigned to a class not indicated on their prioritized lists. This preserves the reality of the current procedures conducted within the Registrar's Office since, although they cannot guarantee that every student will get their top choice, they assure that each student will be assigned a seminar included in his/her submitted list.

Constraint (1) ensures that each student is assigned to exactly one seminar, while the second set of constraints provides the capacity restrictions for standard courses. For the amalgamated courses in constraint (3), our model formulation maintains the enrollment specified by either of the previous approaches (manual assignment or heuristic methods). We felt that this was the preferred plan for these courses since instructors may have specified a particular enrollment (in excess of the 16-student level). In these cases, we wanted to preserve some degree of comparison between the assignments generated previously and those provided with our optimization model. In all likelihood, the fact that relatively few sections exceed the 16-student limit suggests that following this convention ought not to significantly affect our overall results.

Model results

We formulated and solved our linear programming assignment model using a software package known as *What's Best* (Extended 7.0 version from Lindo Systems, Inc.) on a Dell Latitude D800 laptop computer with an Intel Pentium M processor. This particular package works within a spreadsheet environment. Our success with such software further demonstrates the applicability of spreadsheet models in exploring real-world analytical problems (Albright *et al* (2003) and Ragsdale (2004) promote spreadsheets in model-building activities). Moreover, the Registrar's Office already captured much of our data requirements (particularly the objective function coefficient matrix) in spreadsheet format, thus making this tool even more amenable to the specific assignment problem.

We applied our model to student preferences and available seminars in two prior years of course assignments. These previous years included prioritized choices, seminar capacities and assignment results obtained in 2003 (for students due to graduate in 2007) and 2004 (for the class of 2008). As described earlier in this paper, the Registrar's Office used manual methods to generate seminar assignments for the 2003 data, while a computerized, heuristic procedure was adopted in 2004.

Our 2003 data set included 721 first-year students in the College of Arts & Sciences selected from among 46 seminars, with six of these courses being amalgamated sections. For the 2004 data, the Registrar's Office needed to slot 759 students among 45 available seminars (where five courses had allocations above standard levels).

The objective function attempted to minimize the mean ranking of preferences of students slotted into various seminars by trying to satisfy the highest preferences of these entering first-year undergraduates. Obviously, the lowest possible optimal value would be 1.000, resulting if the optimizer allocated each and every student into their highest-preferred seminar while obeying course capacity restrictions.

In each case we explored throughout this study, the What's Best software package determined optimal assignments in a matter of seconds, leading to a rather substantial time reduction as compared to the allocations generated with manual or heuristic approaches. Table 1 compares the solutions obtained with our optimization model to those achieved with earlier methods. While the previous methods worked rather well (a testament to the strong skills of the Registrar's Office in accurately allocating students via manual approaches), our linear programming model provides better assignments for each of the two previous years. The percentage decrease in average rankings in the 2003 data approaches 10%, while it exceeds 13% for the 2004 data. In each year, the number of students receiving their top choice increases (32 extra students in 2003 and 80 in 2004). Meanwhile, this augmentation of highly preferred allocations comes at the expense of fewer students receiving less-preferred seminars. Our optimization model assigns students to a seminar no worse than fourth on their prioritized list. By comparison, the manual assignments for 2003 featured eight students who received a fifth or worse-ranked class (including four rather unfortunate undergraduates who were given their seventh-choice selections).

To substantiate the model's applicability in helping to analyse this important, real-world problem, we explored some modelling extensions. First, we modified our optimization model so that it would maximize the number of students who received their top foundation seminar. We accomplished this by multiplying the R_{ij} values in our objective function by 1000, with the exception of the student's first choice. Consequently, the rankings associated with the second through 10th choices became 2000, 3000, 4000, etc. This served to ensure that the optimizer would allocate as

Table 1 Optimization results

	2003		*2004*	
	Current solution (manual approach)	*Optimal solution*	*Current solution (heuristic approach)*	*Optimal solution*
Weighted average ranking (indicate % improvement)	*1.517*	*1.373 (9.49%)*	*1.568*	*1.357 (13.46%)*
Rankings				
#1	477	509	492	562
#2	156	157	142	135
#3	64	53	88	50
#4	16	2	36	12
#5	3	0	0	0
#6	1	0	1	0
#7	4	0	0	0
#8	0	0	0	0
#9	0	0	0	0
#10	0	0	0	0

Table 2 Analyzing other modelling scenarios

	Maximize number of students receiving first choice				*Fixed class sizes*			
	2003		*2004*		*2003*		*2004*	
	Original optimal solution	*Revised optimal solution*	*Original optimal solution*	*Revised optimal solution*	*Current solution*	*Optimal solution*	*Current solution*	*Optimal solution*
Weighted average ranking	*1.373*	*1.376*	*1.357*	*1.357*	*1.517*	*1.437*	*1.568*	*1.368*
Rankings								
#1	509	526	562	575	477	489	492	559
#2	157	128	135	110	156	162	142	132
#3	53	58	50	61	64	58	88	57
#4	2	9	12	13	16	11	36	11
#5	0	0	0	0	3	1	0	0
#6	0	0	0	0	1	0	1	0
#7	0	0	0	0	4	0	0	0
#8	0	0	0	0	0	0	0	0
#9	0	0	0	0	0	0	0	0
#10	0	0	0	0	0	0	0	0

many students as possible into their top course since significant penalties existed for failing to do so. These results are captured in Table 2. For each class year, we report the original optimal solution (the same as the results provided in Table 1) and our revised optimal solution. For the revised cases, we re-scaled our priority rankings back to the original values so we could compute a comparable objective function value. On the whole, little difference exists between the set of solutions for either year in our study (in fact, the optimal solution values are identical in the 2004 data). Admittedly, when we attempt to maximize first-choice assignments, we provide a few more students with their top choice (an increase of 17 in 2003 and 13 in 2004), but this comes at the price of some students now receiving their third and fourth choices. As a result, even though a few more students obtain that which they highly desire, we would have roughly equivalent allocations on average.

To enhance the comparability of our previous approaches with our optimization model, we explored the issue of fixed class sizes (reported in the right-most portion of Table 2). We took the actual class sizes generated in the previous years and used these as right-hand side values in constraint (2). By adopting such a plan, we guaranteed that the class sizes would be equivalent; the only difference would involve the composition of students in the respective seminars. Improvement in overall average ranking would suggest that we 'shuffled' students between seminars by replacing less-preferred selections with more highly-preferred choices. For each class year, we indicate the actual solution adopted by the Registrar's Office and the optimal solution generated with our optimization model. Even when we restrict class sizes to be exactly equivalent between the different approaches, our approach still obtains a reduction in average ranking. As we initially observed in Table 1, this reduction occurs due to additional students receiving their top choices, while fewer obtain less-preferred selections. Further, we observe a very small difference between our original optimal solution for 2004 (average ranking of 1.357) with the optimal results for fixed class sizes (average ranking of 1.368).

Finally, we endeavoured to explore the relationship between our optimal weighted average rankings and the right-hand side values adopted in constraint (2). By convention, the standard capacity levels were set at 16 students. We examined a range of different capacities and re-solved our model for each case, as reported in Table 3. We shaded the 16-student capacity column for illustrative purposes since it represents the original optimal solution. As we relax the constraint, we obtain better solutions. For each year, the average rank drops and more students get their first selections as right-hand side values climb towards 18. Although we are not aware of any outright inclination to change the standard capacities of 16 students, we felt these results were nonetheless important for Registrar's Office personnel as a basis for 'what-if' analysis. One can quickly observe the amelioration in rankings should these seminars be permitted to carry two additional students. As this constraint tightens, we eventually produce infeasible solutions. If the standard class level were set at 14 students, then there would simply be insufficient seminars (and too many students) to handle such arrangements.

Conclusions

This paper has described the development of a mathematical programming model to explore an important set of decisions made annually by Bucknell University's Registrar's Office. These decisions involved the assignment of incoming first-year Arts & Sciences students to Foundation Seminars, an important component within the University's general education programme. Clearly, our intent was not to replace the accumulated years of expertise already contained within Registrar's Office personnel; as a matter of fact, the manual assignments for the 2003 data, as well as those obtained with automated means in 2004, worked reasonably well. Rather, we wanted to extend their decision-making capability by developing a model that could generate optimal assignments. We note that the assignments produced by our LP model could be used in the capacity of initial, starting assignments of students to seminars. Presumably, one could tweak these allocations in a minor fashion to permit agreement with other scheduling and university concerns, issues beyond the consideration of our mathematical model.

We feel that our optimization approach is beneficial for a few reasons. First, it provides better solutions than those obtained through other approaches for either year in our data set. It reduces the overall average priority ranking of students in their assigned classes, and provides a greater number of undergraduates with their most-preferred choices.

Secondly, it generates optimal solutions very quickly. The *What's Best* optimizer took about 20 s to produce the optimal results. This compares quite favourably with the single week's worth of effort required to generate the manual assignments, or even the 30 min needed for the heuristic approach.

Finally, this optimization model permits a wide range of sensitivity analyses. We illustrated a few such possibilities in our attempt to maximize the number of students receiving

Table 3 Changing standard capacity levels

	2003					*2004*				
Weighted average ranking	*14*	*15*	*16*	*17*	*18*	*14*	*15*	*16*	*17*	*18*
		1.449	*1.373*	*1.325*	*1.289*		*1.440*	*1.357*	*1.300*	*1.269*
Rankings										
#1	No	489	509	533	545	No	546	562	575	593
#2	feasible	157	157	142	144	feasible	126	135	140	128
#3	solution	58	53	46	32	solution	57	50	44	38
#4	obtained	17	2	0	0	obtained	26	12	0	0
#5		0	0	0	0		4	0	0	0
#6		0	0	0	0		0	0	0	0
#7		0	0	0	0		0	0	0	0
#8		0	0	0	0		0	0	0	0
#9		0	0	0	0		0	0	0	0
#10		0	0	0	0		0	0	0	0

their most-preferred selections or to alter the sizes of standard capacity courses. The fact that these solutions were obtained so quickly speaks to the ability of this modelling effort to assist in real-world decision-making.

As an aside, we note that our findings may provide pedagogical benefits for instructors engaged in the teaching of operational research (OR). Regrettably, students may not be inherently motivated in the details of our pedagogy, nor do they find our particular methodologies especially captivating! However, all students understand issues involved in course selection and eventual assignment; by investigating a particular example that is both realistic and highly relevant, students may be more prone to appreciate the material.

A possible direction for further study would involve a reformulation of our model. We would like to permit students to 'weight' their seminar choices when submitting their original prioritized lists. That is, students could divide, say, 100 points between their 10 choices so as to provide a more definitive indication of their desire for particular foundation seminars. This adjustment in the prioritization process would permit the university to obtain a clearer understanding of student choice; consequently, one could determine allocations that more effectively fulfilled student preferences.

This new optimization model would possess the same structural form as the one developed in this paper. The only changes would involve the objective function. It would become a maximization problem (since we would want to place students into sections for which they have indicated a relatively high degree of desire). Moreover, those seminar choices not indicated in a specific student's list would now carry a weight of −99 999, a very large negative number.

We are now working with Registrar's Office staff to implement our model in their annual foundation seminar assignment process. We plan to showcase the improved results obtained with our linear programming model, as well as the efficiency with which additional scenarios can be examined. Optimization models can truly enhance real-world decision-making.

Acknowledgements—Without implicating them, we thank the following employees of Bucknell University for their valuable assistance: Melissa Weber, Associate Registrar and Rita Malick, Systems Integrator.

References

Albright SC, Winston WL and Zappe CJ (2003). *Data Analysis & Decision Making with Microsoft Excel*. Thomson Brooks/Cole: Pacific Grove, CA.

Awad RM and Chinneck JW (1998). Proctor assignment at Carleton University. *Interfaces* **28**(2): 58–71.

Bartholdi III JJ and McCroan KL (1990). Scheduling interviews for a job fair. *Opns Res* **38**: 951–960.

Bazargan-Lari M (2004). Flexible versus fixed timetabling: a case study. *J Opl Res Soc* **55**: 123–131.

Blanco TA and Hillery RC (1994). A sea story: implementing the Navy's personnel assignment system. *Opns Res* **42**: 814–822.

Bloomfield SD and McSharry MM (1979). Preferential course scheduling. *Interfaces* **9**(4): 24–31.

Čangalović M, Kovačevic-Vujčić V, Ivanović L and Dražić M (1998). Modeling and solving a real-life assignment problem at universities. *Eur J Opnl Res* **110**: 223–233.

Ernst AT, Mills RGJ and Welgama P (2003). Scheduling appointments at trade events for the Australian Tourist Commission. *Interfaces* **33**(3): 12–23.

Ferland JA and Fleurent C (1994). SAPHIR: a decision support system for course scheduling. *Interfaces* **24**(2): 105–115.

Gosselin K and Truchon M (1986). Allocation of classrooms by linear programming. *J Opl Res Soc* **37**: 561–569.

Graves RL, Schrage L and Sankaran JK (1993). An auction method for course registration. *Interfaces* **23**(5): 81–92.

Hertz A and Robert V (1998). Constructing a course schedule by solving a series of assignment type problems. *Eur J Opnl Res* **108**: 585–603.

Hinkin TR and Thompson GM (2002). SchedulExpert: scheduling courses in the Cornell University School of Hotel Administration. *Interfaces* **32**(6): 45–57.

Hojati M and Hoang V (1997). *College class timetabling. Presentation at the Administrative Sciences Association of Canada (ASAC) Conference*. St John's, Newfoundland.

Karmarkar US and Kubat P (1983). The value of loaners in product support. *IIE Trans* **15**: 5–11.

Laporte G and Desroches S (1986). The problem of assigning students to course sections in a large engineering school. *Comput Opns Res* **13**: 387–394.

Liang TT and Thompson TJ (1987). A large-scale personnel assignment model for the Navy. *Dec Sci* **18**: 234–249.

Macon N and Walker EE (1966). A monte carlo algorithm for assigning students to classes. *Communications of the ACM* **9**: 339–340.

Mirrazavi SK, Mardle SJ and Tamiz M (2003). A two-phase multiple objective approach to university timetabling utilising optimisation and evolutionary solution methodologies. *J Opl Res Soc* **54**: 1155–1166.

Mongell S and Roth AE (1991). Sorority rush as a two-sided matching mechanism. *Am Econ Rev* **81**: 441–464.

Mulvey JM (1982). A classroom/time assignment model. *Eur J Opnl Res* **9**: 64–70.

Papoutsis K, Valouxis C and Housos E (2003). A column generation approach for the timetabling problem of Greek high schools. *J Opl Res Soc* **54**: 230–238.

Ragsdale C (2004). *Spreadsheet Modeling and Decision Analysis*. Thomson South-Western: Cincinnati.

Roth AE (1986). On the allocation of residents to rural hospitals: a general property of two-sided matching markets. *Econometrica* **54**: 425–427.

Sabin CGW and Winter GK (1986). The impact of automated timetabling on universities—A case study. *J Opl Res Soc* **37**: 689–693.

Sampson SE, Freeland JR and Weiss EN (1995). Class scheduling to maximize participant satisfaction. *Interfaces* **25**(3): 30–41.

Received October 2004;
accepted June 2005 after two revisions