



Towards improving the utilization of university teaching space

C Beyrouthy¹, EK Burke¹, D Landa-Silva¹, B McCollum^{2,3}, P McMullan^{2,3} and AJ Parkes^{1*}

¹University of Nottingham, Nottingham, UK; ²Queen's University of Belfast, Belfast, UK; and

³Realtime Solutions Ltd, Belfast, UK

There is a perception that teaching space in universities is a rather scarce resource. However, some studies have revealed that in many institutions it is actually chronically under-used. Often, rooms are occupied only half the time, and even when in use they are often only half full. This is usually measured by the 'utilization' which is defined as the percentage of available 'seat-hours' that are employed. Within real institutions, studies have shown that this utilization can often take values as low as 20–40%. One consequence of such a low level of utilization is that space managers are under pressure to make more efficient use of the available teaching space. However, better management is hampered because there does not appear to be a good understanding within space management (near-term planning) of why this happens. This is accompanied, within space planning (long-term planning) by a lack of expertise on how best to accommodate the expected low utilizations. This motivates our two main goals: (i) To understand the factors that drive down utilizations, (ii) To set up methods to provide better space planning. Here, we provide quantitative evidence that constraints arising from timetabling and location requirements easily have the potential to explain the low utilizations seen in reality. Furthermore, on considering the decision question 'Can this given set of courses *all* be allocated in the available teaching space?' we find that the answer depends on the associated utilization in a way that exhibits threshold behaviour: There is a sharp division between regions in which the answer is 'almost always yes' and those of 'almost always no'. Through analysis and understanding of the space of potential solutions, our work suggests that better use of space within universities will come about through an understanding of the effects of timetabling constraints and when it is statistically likely that it will be possible for a set of courses to be allocated to a particular space. The results presented here provide a firm foundation for university managers to take decisions on how space should be managed and planned for more effectively. Our multi-criteria approach and new methodology together provide new insight into the interaction between the course timetabling problem and the crucial issue of space planning.

Journal of the Operational Research Society advance online publication, 28 November 2007

doi:10.1057/palgrave.jors.2602523

Keywords: course timetabling; multi-objective

1. Introduction

In this paper, we are concerned with understanding the efficient planning and management of teaching space allocation within academic (or similar) institutions. Teaching space not only includes the usual lecture rooms, but also includes rooms for tutorials, seminars, workshops, etc. Generally, the efficiency of teaching space management is measured by the 'Utilization' U . Exact definitions will be given later but, basically, U is a simple measure of the fraction of the available space that is actually used. A utilization of 100% corresponds to every seat being occupied at all available times. Unfortunately, and perhaps surprisingly, utilization figures are often very low; often around 20–30% in practice. The

'Higher Education Funding Council for England' (HEFCE) has reported low utilizations (HEFCE, 1999), and two of the authors have commercial experience of such low utilizations from their work with Realtime Solutions Ltd (McCollum and McMullan, 2004; McCollum and Roche, 2004). As another example, in work at the University of Puget Sound in the USA, Fizzano and Swanson (2000) report that the registrar asked them space-related questions such as 'How many classrooms does the University need to hold the classes it currently offers?' They include, as one of their conclusions, that 'the university is not using all of its classroom space as efficiently as it might'. Naturally, many institutions would like to improve this situation in order to reduce costs, improve services, or to identify teaching space that might be converted to other uses (eg office space might be in higher demand).

The overall area of usage of space can be divided into two broad areas: 'Space Management' for the near-term

*Correspondence: AJ Parkes, Computer Science Department, School of Computer Science & IT, University of Nottingham, Jubilee Campus, Nottingham NG8 1BB, UK.

E-mail: ajp@cs.nott.ac.uk

usage of existing resources, and ‘Space Planning’ for long-term decisions relating to the provision of space resources. For example, space management handles the assignment of people and activities to existing rooms, whereas space planning is concerned with decisions as to which rooms ought to be built or re-allocated to different tasks. In particular, the long-range nature of space planning implies that decisions need to be made before the exact details of current timetables, student numbers, etc, become available. One approach to cope with this incomplete information is to rely upon some ‘tried-and-tested’ standard practice. This corresponds to relying upon what are called ‘space norms’. An example of a norm might be a physical objective such as ‘5 m² per PhD student’, in which case it can form the basis for space management. Norms of this form can provide the basis for space management in Office Space Allocation (Landa-Silva, 2003). Such norms provide a basis for space planning: use the norms to calculate the overall demand for office space, and then design the supply of rooms to closely match the demand. However, this works well for Office Space Allocation only because, generally, most offices are used by single occupants without the transient use associated with teaching space. Attempting to use similar norms for Teaching Space Allocation is more difficult because the expected low utilization implies that planners must build in a corresponding excess capacity. Furthermore, expected utilizations are such that this in-built ‘safety margin’ has to be as much as a factor of two, or more. This has an obvious and large impact on costs.

Attempts to remedy this situation, and so to carry out better space planning are hampered because there is not an agreed or qualitative understanding of why utilization is so low in the first place. Furthermore, the safety margins incorporated into space norms are obtained from standard sources, the origin of which is generally unclear, and might well be inappropriate for modern module systems, as the sizes of classes will usually have changed significantly. Hence, we have two primary goals:

1. To develop an understanding of the factors that lead to low utilizations.
2. To develop methods to choose safety-margins that are more cost-efficient: aiming to reduce the teaching space that needs to be provided, while not increasing the risk of it turning out to be inadequate.

To these ends, we first consider a simple ‘pure’ event allocation problem in which we optimize utilization by taking events from a pool of courses and assigning them to the available timeslots. On the data sets we have available, this immediately gave utilizations of 85–90%. This is far too high to match reality, and so indicates that a model based purely on space issues, and given free choice of courses, is inadequate to reflect the problem of managing teaching space allocation in real-world universities.

To extend the model we moved in two independent, and complementary, directions:

Extra constraints: Event-allocation usually takes place within the context of many constraints on locations and timings of events. Accordingly, we include within our model objectives that are intended to provide a simplified approximation or abstraction of real timetabling issue.

Threshold phenomena: We study notions arising from the threshold phenomena (also called phase transitions) common in many large systems. Such phenomena arise when typical properties of a system tend to be reliably predictable, based merely on overall properties of an instance; see for example Bollobas (1985). (Phase transitions in course timetabling have been studied from a different perspective in Ross and Corne (1996); see Section 9.)

We find that the location and timetabling-based objectives do indeed have the potential to drive down utilizations, when performing trade-offs in the multi-objective sense. Also, the achievable utilization measures are statistically predictable, and this supports the case for reliable space planning.

The problem classes we use are not new within the general area of course timetabling (for general surveys of the area see de Werra, 1985; Bardadym, 1996; Schaerf, 1999; Carter and Laporte, 1998; Burke and Petrovic, 2002; Petrovic and Burke, 2004). The underlying problem that we will consider is the event allocation problem. This is a well-known existing problem and indeed often occurs as the classroom assignment problem (Carter and Tovey, 1992) and within timetabling problems in order to select feasible room assignments for events.

In standard instances of the assignment problem the sets of events and rooms are fixed within the statement of the problem. This implies that the utilization is fixed from the outset (see Section 2.2), the goal is to improve other objectives such as avoidance of unpopular times, or avoiding unpopular sequences of events. However, our intent is to study the factors that impact on utilization and so we must in some way allow the utilization to vary. This means that our methodology must be different than for standard course timetabling. In this paper, we will allow the set of allocated events to vary. Hence, a novel feature of our approach is that it is not always a hard constraint that all the events must be allocated. Instead, we will use two modes:

Fixed-choice mode: Given a fixed set of courses, the solver is not given freedom to select those that should be allocated or not, but instead we want to know if we can allocate them *all*. This is the mode that standard course timetabling systems use.

Free-choice mode: The solver is allowed a free choice of which courses to allocate when it is optimizing utilization, U . In this mode the selection of events becomes part of the solution.

We work with both the free and fixed choice modes to model the range of management possibilities. At one end of the range, corresponding to fixed choice, the set of courses to be offered is determined without respect to the resulting efficiency of space usage. However, it is also possible that the set of courses themselves are, to some extent, selected within the university so as to improve space usage, and this corresponds to our free choice mode. This is discussed further in Section 4.

To the best of our knowledge, the free-choice mode differs from all of the course timetabling literature in which it is a hard constraint that every event must be allocated a place. This motivated our implementation of a new solver rather than attempting to use an existing one.

We strongly emphasize that this paper focuses on the nature of the space of solutions, and the impact on utilization, rather than on the algorithms used to discover this nature. Although exploration of the solution space of course requires solution algorithms, they are entirely secondary. Hence, descriptions of algorithms are deferred until Section 3.

Another difference of our work from existing course timetabling research lies in the focus and methodology for using the problem instances and optimization algorithms. Typically, timetabling research focuses on a small number of instances, and attempts to obtain excellent solutions with the intention of using the entire solution. In contrast, we take a large number of problem instances, derive reasonable solutions, and then take only ‘aggregate’ properties such as utilization and frequency and discard the rest. We then look at how these aggregate properties change as we manipulate the overall resources, sizes, and other aggregate properties of the problem instances. Generally, we only need a reasonable solver as improving the solver will have minimal difference on the patterns we are studying. We expect it is more important for our solver to be robust, in the sense of consistency between instances, than for it to be particularly well-performing on a restricted set.

We use problem instances based on real data for courses and rooms obtained from a University in Sydney Australia, and so we expect our methods, and the broad picture of our results, are likely to be applicable to other institutions.

In this paper, Section 2 covers the basics of the problem: the terminology and algorithms used. For example, we will see that maximization of utilization alone is a straightforward optimization problem often reducing to maximum weight bipartite matching. Section 3 covers the algorithms, and the data instances. Section 4 displays the threshold phenomena, and introduces the question of when a request for a specific amount of utilization or frequency is likely to be satisfied; which we will call safe *versus* unsafe requests. Section 5 introduces specific location and timetable penalties. Section 6 presents the Pareto fronts, or multi-objective trade-off surfaces, for utilization, location and timetabling objectives. Section 7 returns to the issue of safe or unsafe requests, but this time in the presence of timetabling

constraints. Section 8 covers safety in the presence of location constraints.

2. Background and basics

In this section, we cover the basic background needed for the main results. We describe the terminology of the domain, the constraints and the objective functions that measure the space usage.

2.1. Basic terminology and the hard constraints

For each teaching room, assume that we are given:

1. *Capacity*: the maximum number of students that the room can accommodate.
2. *Timeslots*: the number of timeslots for which the room is available during the week (or other relevant scheduling time period).
3. *Department*: the department that ‘owns,’ or is most closely associated with, the room.

An ‘event’ requires the following information:

1. *Students*: the number of students that must be accommodated.
2. *Department*: the owning/associated department.

The primary task is to assign events to rooms so as to satisfy the following hard constraints:

1. *Room capacity*: the size of an event (students) must not exceed the room capacity.
2. The number of events allocated to a room must not exceed the number of timeslots, as events cannot share room timeslots.

In fixed-choice mode, there will also be a constraint saying that every event/course must be allocated to some room. However, in free-choice mode it is part of the problem to find a set of events that are to be allocated so as to maximize objectives.

2.2. Quantifying space usage

The simplest and most direct measure of the space usage is to take the sum over all timeslots and rooms of the number of students allocated to that room-slot, which we will refer to as ‘seat-hours’ (though of course there is no implication that the timeslots really need to be an hour long). The intent is that ‘Utilization’ measures the fraction (typically expressed as percentages) of the total available (or maximum) seat-hours that are actually used:

$$\text{Utilization} = \frac{\text{achieved seat-hours}}{\text{available seat-hours}} \quad (1)$$

Let C_i be the capacity of room i , and $S_{i,t}$ the number of students allocated to room i at timeslot t . Then the total

number of seat-hours (denoted by B) is

$$B = \sum_{i,t} S_{i,t} \quad (2)$$

Since we enforce

$$S_{i,t} \leq C_i \quad \text{for all } i, t \quad (3)$$

we then have $B \leq B_M$, where B_M is the maximum number of seat-hours and is simply defined as follows

$$B_M = \sum_{i,t} C_i \quad (4)$$

Generally, the utilization is defined by means of ‘occupancies’ and ‘frequencies’ (McCullum and McMullan, 2004; McCullum and Roche, 2004). The occupancy $O_{i,t}$ of room i at time t is the fractional usage at that time

$$O_{i,t} = \frac{S_{i,t}}{C_i} \quad (5)$$

The occupancy, O_i , of a room, i , is defined as the mean of its occupancies over all *occupied* timeslots. Suppose that for room i the number of timeslots is t_i and the number of occupied timeslots is t_i^{occ} . Then the occupancy for room i is defined as

$$O_i = \frac{1}{t_i^{\text{occ}}} \sum_t O_{i,t} \quad (6)$$

The frequency usage, F_i , for a given room, i , is defined as the fraction of its timeslots to which some event is assigned:

$$F_i = \frac{t_i^{\text{occ}}}{t_i} \quad (7)$$

The utilization, U_i , of room i , is the product of its occupancy and frequency:

$$U_i = O_i F_i \quad (8)$$

and so

$$U_i = \frac{\sum_t S_{i,t}}{\sum_t C_i} \quad (9)$$

that is, U_i is simply the fraction of the room’s seat-hours potential that is actually used. However, to obtain an overall utilization we will need to combine the utilizations from different rooms. We will take a weighted mean over the rooms

$$U^w = \frac{\sum_i W_i U_i}{\sum_i W_i} \quad (10)$$

where W_i is the weight assigned to room i . Usually, one just finds an unweighted mean U^{UW} corresponding to the special case that $W_i = 1$. However, a natural and simple choice is that larger rooms have a larger weight; and so we take the weight

to be the room capacity, $W_i = C_i$. In this case, straightforward manipulation yields

$$U_i = \frac{\sum_{i,t} S_{i,t}}{\sum_{i,t} C_i} = \frac{B}{B_M} \quad (11)$$

which is just the promised overall fractional usage of the seat-hours. In our view, U as defined in 11 is preferable, as it is conceptually simpler than U^{UW} , at least as good a measure, and that the practical differences will generally only be a secondary effect. (In some experiments not presented here, we looked at both U and U^{UW} and found them to be tightly correlated anyway.)

It is important to observe that the utilization is a function only of the rooms and the events allocated. It does not depend on the details of the allocation itself. As discussed in the introduction, optimizing the allocation with respect to objectives usually used within course timetabling has no effect on the utilization. To study utilization, it is essential to allow the events or the rooms to vary, and this motivated the free and fixed choice modes.

We will also measure the overall frequency F of a solution,

$$F = \frac{\text{timeslots used}}{\text{timeslots available}} \quad (12)$$

We do not weight frequency by the size of rooms, because we want a measure that is direct and simple to understand, and also because F is a ‘counting measure’ that ought not in itself take account of room sizes. Again, in any case, we would expect other frequency measures to give similar results.

3. Optimization algorithms and data

As discussed in the introduction, this paper focuses on the nature of the space of solutions rather than on the algorithms that we employ. However, for completeness, we briefly describe them: firstly, we use mathematical programming to exploit cases that reduce to a max-weight matching problem, and secondly we employ a local search algorithm. Firstly, however, we briefly discuss some existing Timetabling algorithms. We remark that recent work (Avella and Vasil’ev, 2005) on exact (probably optimal) solutions was limited to relatively small instances of course timetabling; up to 69 courses and 15 rooms. For larger instances it is necessary to use (meta)-heuristics. In this case, a general pattern of the most successful studies is that firstly a constructive algorithm is used in order to produce initial feasible solutions, followed by improvement of the feasible solution using some form of heuristic local search. For example, in the ‘International Timetabling Competition 2002’ (Metaheuristics Network, 2003) the top four solvers used were: (1) simulated annealing (SA) (Kostuch, 2004), (2) tabu search (Cordeau *et al*, 2003), (3) Dueck’s Great Deluge (Dueck, 1993) in Burke *et al* (2003), and (4) Tabu search with shakes (Di Gaspero and Schaerf, 2004). Also, in a comparison of performances of different meta-heuristics (Rossi-Doria *et al*, 2003, it is

noteworthy that the authors restricted themselves to local search ‘in an evolutionary algorithm, and ant colony optimization algorithm, and an iterated local search. An SA, and a tabu search were restricted to the same neighbourhood structure’. Hence, local search is the favoured method for improving feasible solutions. Hybrid algorithms have also been used successfully (Chiarandini *et al.*, 2006). In our case, because we do not have the hard constraint that all events must be allocated a time and place, constructing an initial feasible solution is trivial. Hence, our algorithm ‘only’ needs to do improvement, and hence will do so using local search (with SA).

3.1. Mathematical programming methods

Suppose we call each (room, timeslot) pair a ‘room-slot’, then the event allocation problem is to assign events to room-slots, and to maximize the allocated seat-hours. In the absence of other constraints or objective functions, it is well known that this is just a standard assignment problem, and reduces to a maximum weight matching problem in a bipartite graph (see, eg, Cormen *et al.*, 2001). The events are taken to correspond to one set of nodes in the graph, and room-slots to the other set. The edges are the set of possible assignments of events to room-slots for which the capacity is sufficient. The weight, or value, of an edge is the contribution of the assignment to the total seat-hours, hence, simply, the number of students in the event. We are forced to have a bipartite matching because events can be assigned to at most one room-slot, and each room-slot can have at most one event allocated to it. Max weight matching has polynomial time complexity using the standard network flow algorithms. The simple optimization of utilization for event-based assignment is not a hard problem. For simplicity, we instead exploit this by converting the assignment problem to a mathematical programming formulation. We encode it as an (binary)integer programming problem (see Nemhauser and Wolsey (1988), or see Bosch and Trick (2005) for a recent brief introduction to mathematical programming), but then relax to a linear programming (LP) problem, and will still expect to obtain integer solutions, which are hence optimal for the integer program as well. We use this to derive optimal solutions when appropriate. This is used for checking that the local search is working well.

In some cases, the problem reduces to the assignment problem but with just an extra constraint which means that the solutions from the LP are not necessarily integer. We exploit a ‘rounding’ method as follows. We solve the problem as an LP and extract the integer parts of the solution. The integer parts are then added as constraints to the original problem, which is then generally small enough to be solved as an IP. That is, we take the variables that are set to 0/1 in the LP, but leave the fractionally valued ones to be determined by the integer program. In the very simplest cases, when we only want to maximize utilization with no other constraints, it is

sufficient to use a simple greedy method of ‘scanning events largest first and allocating whenever possible’. However, our main aim is to solve instances with other constraints, and for these the greedy method will fail to be optimal, and so we use mathematical programming and local search.

3.2. The local search algorithm

Local search is performed on solutions in which some events are allocated to room-slots and others are unallocated. Operators are used that maintain feasibility (do not break the hard constraints such as capacity), and are as follows

1-swap-rand: Select two different rooms at random, and from each room randomly select an allocated event. If it maintains feasibility, then swap the two events between the room-slots.

2-swap-rand: Similar to *1-swap-rand* except select four rather than two events and swap them while maintaining feasibility of the given solution.

Move-exterior-rand: Randomly selects an allocated and an unallocated event. If it maintains feasibility, then the allocated event is deallocated, and the previously unallocated one given its room-slot.

Push-rand: Randomly select one unallocated event and one room. Try to allocate the event to the room; selecting the timeslot at random from those (if any) that would maintain feasibility.

Pop-rand: Randomly select one event from a randomly selected room and deallocate it.

Move-inner: Swap the timeslots of two randomly selected events in a single randomly selected room.

The operators use random sampling because the underlying neighbourhoods tend to be quadratic in the number of events and too large to be searched completely.

The search itself is performed with either standard Hill-Climbing (HC) or SA (Kirkpatrick *et al.*, 1983). Each move operator is assigned a static probability for selection. On each iteration, we first select an operator according to their probabilities. Multiple, but limited (we use 10), attempts are then made to apply the operator in order to generate an improving candidate move. An iteration ends when a move is accepted or a pre-defined number of failures is reached. In SA, worsening moves can be accepted depending in standard fashion on the temperature. In our experiments we use standard geometric cooling with reheats.

When possible, the local search was compared against optimal solutions derived from the LP solver. When this was not possible, we compare standard runs of the SA against runs using much slower cooling, and many more reheats. Our standard run using SA is four coolings, 4 m iterations each, cooling by a factor of 0.998 each 650 iterations, taking 20–60 min, and is chosen so that the search seems to become static at

the end of each cooling. We take the best result obtained at the end of each cooling, and found that the multiple reheats do help. We have checked that even with much longer runs (10 coolings, 15 m iterations each, cooling of 0.9995 each 650 iterations, taking up to 3 h) the graphs presented do not change significantly. This gives us confidence that the results presented here are a good reflection of the underlying properties of the solution spaces, and have not been biased by the search methods.

3.3. Problem instances

The real data set that we use arises from the ‘Appleby’ building of a university in Sydney, Australia. The data contains many different space types; lectures, workshops, seminars, etc. However, for the purposes of this paper we are not covering the issues of splitting. We seek clarity, and so we select only the lectures (and also eliminate one lecture that is so large that it would need splitting). We have 20 rooms, and each has 50 timeslots. This gives a total of 1000 timeslots, whereas the lecture courses only have 608 events. Also, the total seat-hours demand from the lecture courses is 69 983 whereas the total supply from the rooms is 202 650. Hence, in the initial data set, the lectures are substantially under-subscribed, in the sense that the total demand for seat-hours and timeslots from the courses is much smaller than the supply of seat-hours and timeslots from the rooms.

In order to explore a wider range of these supply-to-demand ratios we need to do one or more of (i) add more courses, (ii) reduce the number of rooms, or (iii) have fewer timeslots per room. We opt against creating more courses, as it would make the problems unnecessarily large. The options of reducing rooms or reducing timeslots are similar in that they reduce the available seat-hours. Eliminating rooms requires a decision of which ones to remove, and it is hard to know what counts as a fair reduction, especially as we suspect that it is the distribution of room and course sizes that is the most important, and so do not want to change it accidentally (and this is also why we do not attempt to use a random generator for instances). So, instead, we uniformly reduce timeslots for all rooms. Hence, we create ‘Lecture Room’ problem instances, LR(T), with the timeslots per room reduced to T . In the original data $T = 50$, but we also studied $T = 10, 18$, and 30. The case $T = 18$ is the smallest T in which the seat-hours demand could potentially still be met by the rooms. The results were similar to those we present here. We have now covered the basic terminology, algorithms and data sets used, and so can move to the main methodology and results.

4. The safety of utilization and frequency requests

Suppose that we are carrying out space planning, and have a proposal for a set of rooms and a reliable forecast for the expected demand for total seat-hours from courses. We would like to know whether we can be confident that it will be possible to satisfy the demand, but we do not yet know exactly

how the demand for seat hours will break down into actual courses. Instead we just expect that the demand will arise from a subset of some much larger set of potential courses. Given the set of rooms and so the supply of seat-hours, then the expected seat-hours demand can be converted to a ‘requested’ utilization, U_R . In the absence of low utilizations, we could be confident that as long as $U_R \leq 100\%$ then we would be able to satisfy all the demand; that is, the ‘achieved’ utilization, U_A would equal U_R ; but maybe this is no longer true when U is expected to be low? Hence, in this section we build towards answering the question

‘Under what conditions is a request for utilization fully satisfiable?’

To answer this question, and as a general tool for analysis, we introduce the idea of ‘Achievement Curves’. These will represent the quality (either U or F) of valid solutions in terms of the quality requested, U_R or F_R , that is the quality that would have been achieved if the entire request could have been satisfied, and compare to the quality, U_A or F_A , of the Achievable solutions.

We find achievement curves using the following procedure: for each value of probability $p \in [0.01, 0.02, \dots, 1.0]$.

$S :=$ a random subset of the courses; taking each course independently with probability p .

1. Sum the sizes of events in S to find the total requested seat-hours, B_R .
2. Optimize the utilization for S to determine the achieved seat hours, B_A .

We repeat this many times to generate more points.

The requested and achieved values for seat-hours are converted into achieved and requested Utilizations U_R and U_A . We also measure the total number of requested and achieved events to give the requested frequency (F_R) and achieved frequency (F_A). Thus, each fixed, but randomly generated, subset S , generates data points (U_A, U_R) and (F_A, F_R) . Note that we can request a U (as total seat-hours), or an F (as total timeslots), but it does not seem to be useful to talk about a ‘Requested occupancy’. Although we have three measures, U , F and O , only two of them are independent, as they are related by ‘ $U = FO$ ’. It seems simplest and clearest to select the two independent measures to be U and F .

Figure 1(a) presents the results of following the above procedure for the room data LR(10). We find that as well as plotting U_A versus U_R it is also helpful to plot the ‘Fractional Achievement Ratio’; that is the fractions of the requested U or F that turn out to be achievable. The results in terms of fractional achievement are given in Figure 1(b).

The first, and crucial result, is that the values of achieved U and F , for given corresponding requests, tend to be ‘grouped around the mean’. That is, the variation of U_A between points near to some value of U_R is small compared to the value of U_A itself, and similarly for F_A . This implies that properties of the system are statistically predictable. In our case, take

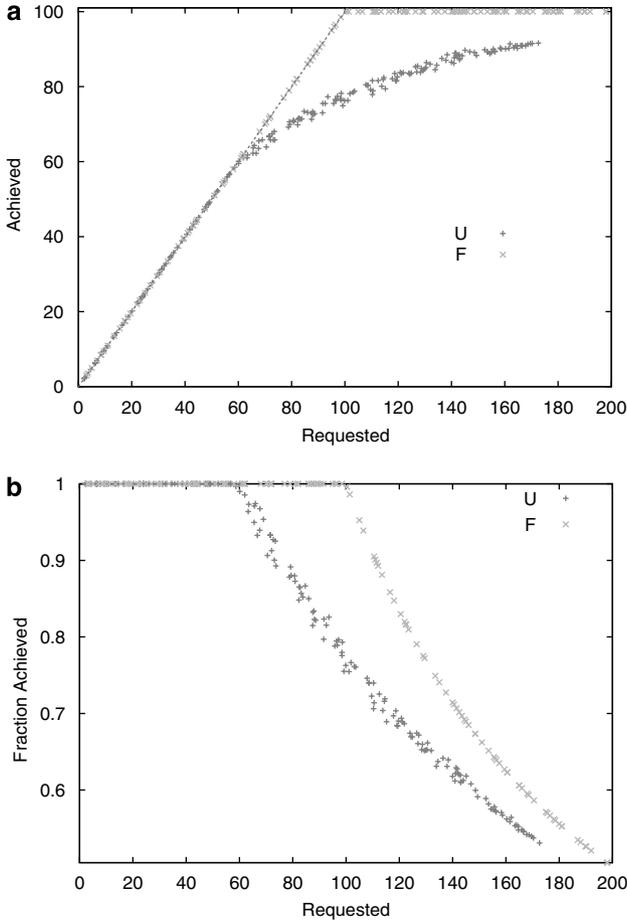


Figure 1 (a) Requested *versus* Achieved percentages for U and F , for random subsets of the courses, and with the rooms LR(10). The diagonal line, ‘Achieved = Requested’, is given for reference purposes. (b) Same data but for the ‘fractional achievement’, that is, the y-axis is U_A/U_R or F_A/F_R .

for example, $U_R = 80\%$ then the mean value for the achieved utilization is $U_A \approx 70\%$ but, crucially, the variation between instances in that region of the curve is relatively small. This is important, because if the variation were very large we would not be able to make reliable and fairly tight predictions of the achievable utilization or frequency.

The second, and also crucial result from Figure 1, is that we see a threshold phenomenon on U (see also Beyrouthy *et al.*, 2006b). There is a ‘critical value’, U_C , for the requested utilization, U_R . In this case, $U_C \approx 60\%$, and this value divides the results into two distinct regions:

SAFE: $U_R < U_C$. Requests for seat-hours are almost always totally satisfied.

UNSAFE: $U_R > U_C$. Requests for seat-hours are almost never totally satisfied. Even in the cases when there are enough seat-hours available, it turns out that the oversupply is actually unusable.

The region around U_C between these, the ‘critical region’, is relatively narrow and within it the satisfiability of requests is less predictable.

This has important implications. When planning course offerings, we cannot assume that we can simply count seat-hours, but must account that we are unlikely to be able to rely upon using more than 55% of the seats available (in this case). But for $U_R > U_C$, we will (almost) inevitably find that some of the events will need to be dropped.

We refer to a request for U as ‘safe’ when statistically there is high probability (eg, with better than 95% chance) that it will be possible to satisfy all of the request. We use the term ‘safe’ to imply ‘low-risk but not an absolute guarantee’.

Note that, in this case, the frequency is always maximally satisfiable. Obviously requests $F_R > 100\%$ are unachievable, but here all requests $F_R \leq 100\%$ are safe. In this case, it turns out that the frequency is the limiting factor. Even if $F_R = 100\%$ and all of the events are allocated, $F_A = 100\%$, and all the timeslots are occupied then the overall utilization is only around 50–60% because most of the rooms are not fully filled. However, in general, maybe there are different regimes according to the most important limiting resource, whether it be seats, or timeslots, or something else.

The other interesting points on Figure 1(a) are the endpoints at the largest values of requested U and F . These correspond to taking all the courses, but allowing the solver to make a free choice of which events are going to be allocated a room-slot. Recall, the free-choice mode allows the solver to pick events that are better suited to the room sizes, and so to increase U . From Figure 1(a), for this data set and no extra constraints, with free choice we can reach $U \approx 92\%$ but with the ‘fixed choice’ we are only safe up to $U \approx 57\%$. The real situation would probably be somewhere in between these two extremes.

That is, using both the fixed- and free-choice modes gives information about the range of utilizations possible. Roughly, the fixed choice corresponding to the case that utilization is not considered when deciding which courses (or modules) are offered. Conversely, the free-choice mode can be taken to model the case that utilization is the primary objective when deciding on offerings.

If courses were selected from the pool with no regard to overall utilization, then the safe choice would be limited to $U < 57\%$. However, in practice, there probably is an effect (that accrues from term-to-term) that the sizes of the courses will evolve towards being a better fit to the rooms. So, arguably, a natural evolution might push us a little above the safe point. However, it seems unlikely that such natural evolution would be so strong as to achieve the highest ends of the utilization values.

The ‘grouping about the mean’ and thresholds observed here are fairly common properties of problem classes in which instances are selected from a large set of possibilities. (It is important to remember that the number of subsets is exponentially large: with n courses there are 2^n possible subsets.) The phenomenon is analogous to that of phase transitions in

physical systems (such as water into ice), and in computer science is best known in the context of random graphs (Bollobas, 1985). For example, a standard distribution for random graphs is to take n nodes and add every potential edge independently with probability p . In this case, many properties of the graphs become statistically predictable from the values of n and p , and boolean properties, such as ‘the graph is connected’, will exhibit a threshold at some critical values of p .

5. Introducing location and timetable objectives

In the example of the previous section, we found that the safety requirements were $F \leq 100\%$ and $U < 57\%$. However, this corresponds to using all timeslots, and the rooms being more than half full, and is still unrealistically optimistic. We suspect that, in practice, the need to take account of other objectives and constraints will drive down the achievable U and F . Real problems have many different constraints, and we cannot consider all possibilities for constraints. Instead, we focus on two standard ones. In this section, we introduce specific ‘location’ and ‘timetabling’ objectives.

Location penalty (L): Even if the allocation decisions are made by a central administration: lecturers and students will generally prefer that the events they attend should be close to their ‘home’ department. We decided to model this using a simple penalty determined by the department ‘owning’ the event, and the department ‘owning’ the room. Events allocated to rooms from the same department receive zero penalty. If the event department and room department are different then the allocation is penalised. (Owing to the absence of specific distance information, we arbitrarily selected values from range and to mimic the likely range of distances involved.)

Note that such a location matching is a common desire within course timetabling. For example, after their conclusion that ‘the university is not using all of its classroom space as efficiently as it might’, Fizzano and Swanson (2000) continue:

‘Our results do not guarantee that there are practical schedules that use the number of classrooms we determined because our process does not consider things like teachers’ room preferences or class location requirements (English classes might not end up near the English department).’

It is important to note that the penalty depends only on the room and event. This has an implication on the algorithms that should be used. Any weighted linear combination of location and utilization still corresponds to a maximum weight bipartite matching. The combination of location penalty and seat-hours together generate a new set of edge weights and network flow methods give efficient solutions. However, if L and U are treated as independent objectives we end up with a bi-criteria matching problem which is harder due to the presence of unsupported solutions (Steuer, 1986; Tuytens *et al.*, 2000).

Timetable penalty (TT): In order to take some account of the effects of timetabling, we introduce a conflict graph between events. Enrollment or conflict data was not available and so we used simple randomised generators for the conflict matrix. Also, the conflicts are again based on the owning department for each event. Specifically, we generate conflict matrices using procedures denoted by ‘TT(p, q)’, and according to the following:

1. conflicts between events from the *same* department are generated (independently) and randomly with probability p ,
2. conflicts between events from *different* departments are generated independently and randomly with probability q .

This corresponds to expecting that events from the same department are more likely to have students in common, or simply that departments strongly prefer that their own events do not clash. The penalty is simply taken to be the number of edges in the conflict graph that are violated because the events are allocated to the same time. We expect that the timetabling constraints of this form will capture some of the broad effects of real problems; though an important aspect of future work will be to devise more realistic methods to produce conflict matrices. A similar structure was used by Ross and Corne (1996) where they had a ‘probability p_w , for within-clump constraints, or p_b for between-clump constraints’. Their motivation was that ‘Real timetabling problems are typically rather more clumped than homogeneous. For example, exams within an arts faculty may typically form a distinct clump, largely separate from those within a science faculty.’

The case $p=q$ corresponds to ignoring the department, that is, a standard random graph (Bollobas, 1985), and will refer to this as simply TT(p). Another simple case is TT(100,0): the conflict graph has edges between any two events in the same department but none otherwise.

6. Multi-objective optimization in the free-choice mode

Obviously, adding location or timetabling constraints and penalties to a problem cannot increase utilization. Instead, it is a natural expectation that they will drive it down. However, the issue is the magnitude of such an effect. In particular, it is important to know whether such effects are sufficiently large to have the potential to cause the low values of utilization seen in practice. To study this, we are first interested in reducing the upper estimates on utilization. Recalling the discussion in Section 4, this suggests that we first consider free-choice mode. We take all the events, and allow the solver to select those that will be allocated. We will treat the system as a multi-objective problem using the utilizations, and (the negatives of) the location penalty L , and timetable penalty, TT, and determine the appropriate (approximate) Pareto fronts (see Steuer (1986), Deb (2005) for descriptions of the concepts of Pareto optimization).

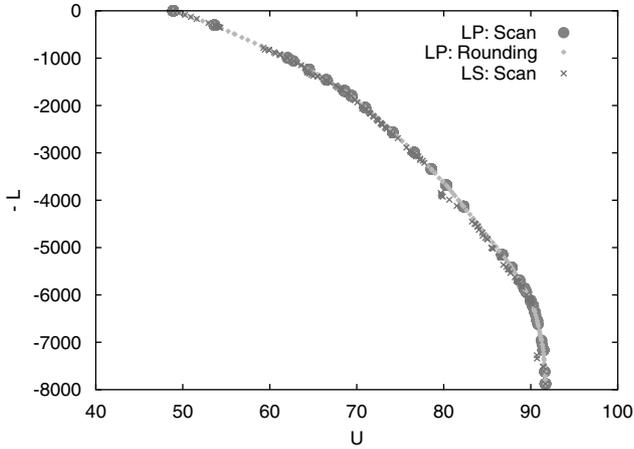


Figure 2 Pareto front between location L and Utilization, U , for the data set LR(10). ‘LP: Scan’ and ‘LP: Rounding’ are obtained from the mathematical programming methods. ‘LS: Scan’ uses the local search method.

Figure 2 presents the results for the simple two-objective case of the tradeoff surface, Pareto front, between Utilization and Location objectives. For clarity, we plot $-L$ rather than the location penalty L itself, so that all axes correspond to maximization problems. Also, we remove all Pareto-dominated points.

Each of the first set of points, ‘LP: Scan’, are obtained by a standard procedure of solving many different possible linearizations of the problem. That is, each point is obtained by giving each objective a weight and then solved using linear programming (LP), as discussed in Section 3.1. Each point in ‘LP: Scan’ is hence Pareto Optimal, but some Pareto optimal solutions can be ‘unsupported’ meaning that are not reachable with any set of weights (Steuer, 1986; Tuytens *et al*, 2000), and this leads to gaps in the front. To generate unsupported solutions and to fill in such gaps, we used the IP formulation together with an upper bound L_{\max} on the location penalty. The resulting problem is now *constrained* maximal matching and the LP relaxation no longer automatically produces integer optimal solutions. However, in practice, the LP optimal solution contained very few non-integer values (just 5–10 variables out of thousands). Hence, we used the rounding method of Section 3.1 to convert these to give the integer ‘LP: Rounding’ solutions in Figure 2; each point being obtained from different choices for L_{\max} . There was little difference in objective values between the LP relaxation and the solutions from rounding. This indicates that the underlying problem is rather easy in this case (It was also observed in the early days of the office space allocation problem (Ritzman *et al*, 1979) that the LP relaxations of IP formulations can result in very few non-integer variables, and so the problems are relatively tractable for their size.).

The final set of points ‘LS: Scan’ in Figure 2 are obtained using our local search method (SA). It gives points that are

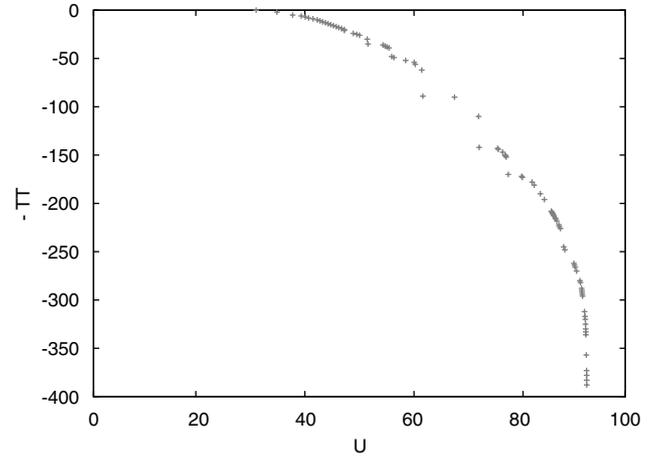


Figure 3 The trade off between Utilization, U , and Timetable Objectives, $-TT$, for the rooms LR(10).

also very close to the optimal ones from the LP methods—the difference being small enough so as to not significantly change the shape of the curve. Altogether, these results give us confidence that (i) the Pareto Fronts are non-trivial and (ii) using local search instead of an exact method is sufficient to obtain a reasonable approximation to the front.

The primary result from Figure 2 is that incorporating the location objective can indeed significantly reduce the utilization: In this example, driving it down to about 50 from 93%. Figure 3 gives the results of a similar experiment for the trade-off curve between utilization and timetabling conflicts. Conflicts are introduced using TT(100,0), and broken conflicts are penalized. In this case, we do use an exact method, but only the local search. We see that the reduction of the timetabling penalty in this free-choice mode forces some events to be left unallocated and so the utilization is significantly reduced. In the case of hard conflict constraints, or zero timetabling penalty, the most that can be achieved is a utilization of 32%. We have also studied the Pareto fronts obtained by using all three utilization, location, and timetabling objectives. Lack of space means they are not presented here. However, they have a similar standard structure. For example, for the LR(10) data set, forcing the penalties for both location and timetabling to zero drives the utilization down even further, to $U \approx 29\%$.

Altogether, our results support our hypothesis that the location and timetable penalties have the potential to adversely affect utilizations. These penalties provide some explanation for low utilizations in the real world.

7. Fixed choice with various timetabling models

In the previous section, we found that the location and timetable requirements can significantly reduce utilizations within the free-choice mode. In this section, we investigate

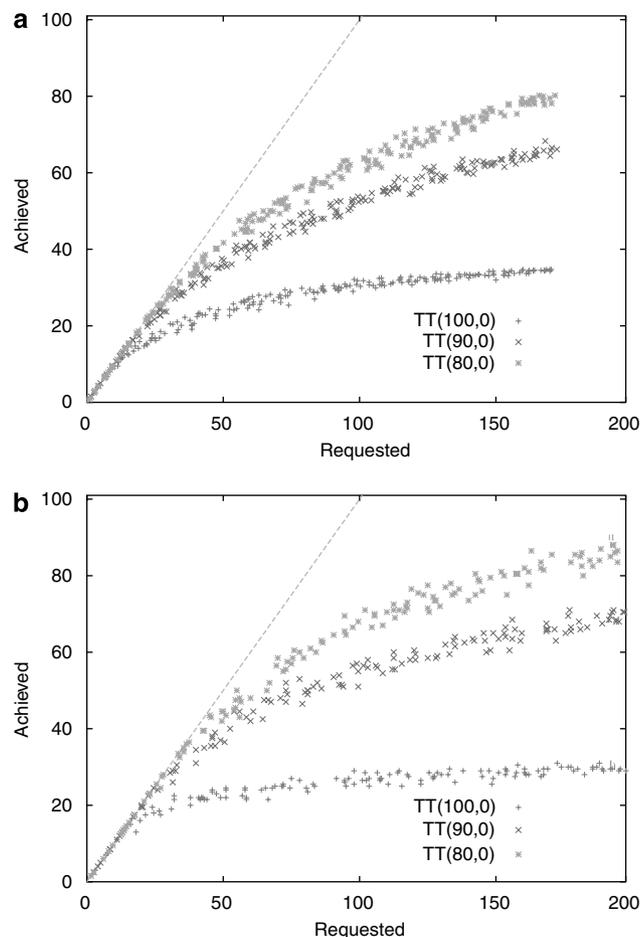


Figure 4 'Requested versus Achieved' Curves for (a) Utilization, U and (b) Frequency, F . In the presence of hard timetabling constraints produced by $TT(p, 0)$ with $p = 100, 90,$ and 80% (but with no restriction on location penalty).

the effect of timetabling within the fixed-choice mode. In particular, we look at the effects of timetabling on the safety of requests for utilization and frequency. The basic procedure is the same as in Section 4, except that for each selected subset of the courses we also generate a corresponding conflict graph, and then the allocation problem is solved treating the conflicts as hard constraints.

Figure 4 shows the effect on the achievement curves of enforcing hard $TT(p, 0)$ constraints. For comparison, Figure 1 corresponds to the case that $p=0$. We see that the introduction of the conflict graphs has significantly reduced the utilizations and, in particular, the critical utilizations. Furthermore, in the absence of timetabling, Figure 1, it was the case that all legal frequencies, $F \leq 100\%$, were safe, but this is no longer true. The hard conflict constraints force some timeslots to remain unfilled and the safely achievable frequency drops. Also, note that the difference between $TT(100,0)$ and $TT(90,0)$ is quite large, indicating that safety regions can be fairly sensitive to the parameters of the model.

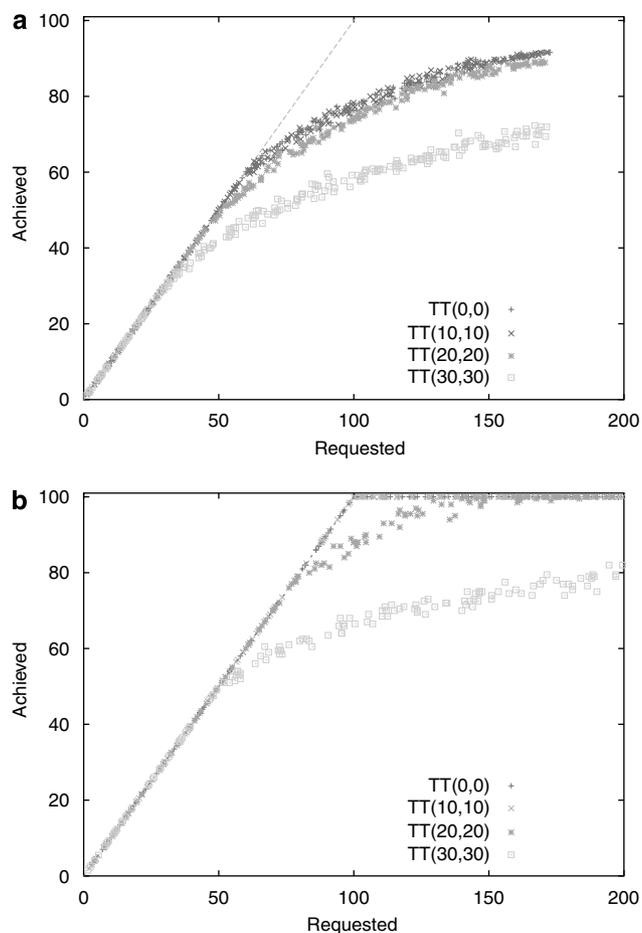


Figure 5 The same as Figure 4 except that the hard timetabling constraints are produced using $TT(p, p)$ with densities $p = 0, 10, 20,$ and 30% .

Figure 5 shows the effects on the achievement curves of enforcing hard $TT(p) = TT(p, p)$ constraints; the conflicts are independent of the owning departments of the events. Most notable from Figure 5(b) is that conflict densities of 10% do not lower the safe region. In this instance, the safe region for F only starts to become reduced when the conflict density reaches about 15% . In this case, the timetable conflicts have no effect on the safe regions until the conflict density exceeds some value. Of course, the details will depend on the particular problem instance, and so the exact numbers obtained on this instance are not important. Instead, the message is that safe regions may well be insensitive to the imposition of 'small' amounts of other objectives.

Given the $TT(p, q)$ model, then the parameters p and q correspond to a parameter space controlling the conflict matrices. We have just seen that the critical values of U and F can change rapidly in some regions of the parameter space, but not at all in other regions. We believe that understanding

such sensitivity effects will be important in safely using critical values within space planning.

8. Fixed-choice together with both hard L and TT

In this final section of experimental results, we briefly study the effects of demanding that the assignments totally match the location; that is, we treat location as a hard constraint ($L = 0$). This can be taken to correspond to an institution with very localized control of rooms or no sharing between departments.

Figure 6 shows the achievement curves with both a hard location constraint, and a hard timetabling, TT(90,0), constraint. Again the achieved U and F are statistically predictable in that they are ‘clustered about the mean’. At first sight, it seems that the constraints have merely reduced the safe regions as in previous cases. However, closer inspection of the achievement ratio, Figure 6(b), reveals that the effect is more extensive. There is no safe region in which we are almost always sure of satisfying all of the request. Instead we get a ‘weakly-safe’ region, a region in which we

are ‘only’ almost always sure of satisfying a large fraction of the request. In particular, if we request any F less than about 40% then we are very likely to be able to satisfy about 95–98% of the request, but not the 100% we would expect in a safe region. We suspect that this indicates there is a mismatch between rooms and events, and will explore such cases in future work. Again, this serves as an example of what can happen; but the exact details will of course depend on the instance under consideration.

9. Related work

Room minimization: As mentioned in the introduction, Fizzano and Swanson (2000) also studied space usage within a university. They carry out room minimization by the simple procedure of removing rooms whenever doing so would not result in the problem no longer being solved by their algorithm. After removing unneeded rooms, the overall frequency increases correspondingly. Their constraints used within are substantially different from the majority of the course timetabling literature. Their time restrictions arise because courses consisting of multiple events must take place on some specified subset of the days of the week and with the restriction that the course events must take place at the same time of day, but they do not have the usual ‘conflict matrix’. Also, they do not include the effects of location objectives. However, the most important difference is that they do not perform our multi-objective or phase transition studies.

Computational hardness and phase transitions: Phase transitions in the area of course timetabling were studied by Ross and Corne (1996). However, this was from the point of view of the computational hardness rather than the positions of the phase transition (as a function of the controlling parameters such as conflict density). Many systems have a threshold, and it is well known that the computationally difficult decision problems, ‘hard problems’, typically occur at the threshold; that is, at the phase transition between ‘almost always yes’ and ‘almost always no’ regions (Cheeseman *et al*, 1991). (Possibly one of the most well-studied thresholds, and associated hard problems, is the satisfiability transition in the ‘Random 3SAT’ domain (Mitchell *et al*, 1992; Parkes, 2002, and many others).) However, in Ross and Corne (1996) the instances themselves are generated in such a fashion as to guarantee that they are solvable. Their focus is instead on the hardness of the instances for a solver, and they do indeed observe transitions in hardness. We differ in that we are not investigating the hardness (in this paper) but rather the satisfiability of a request for a particular overall utilization. In our studies, it is reasonable to expect that there will be a peak in hardness for the satisfiability decision problem near the transition from safe to unsafe utilization requests, and this is under investigation. However, conversely, we expect the hardness to drop rapidly as we move away from the threshold. Somewhat paradoxically, this means that it is generally not very hard

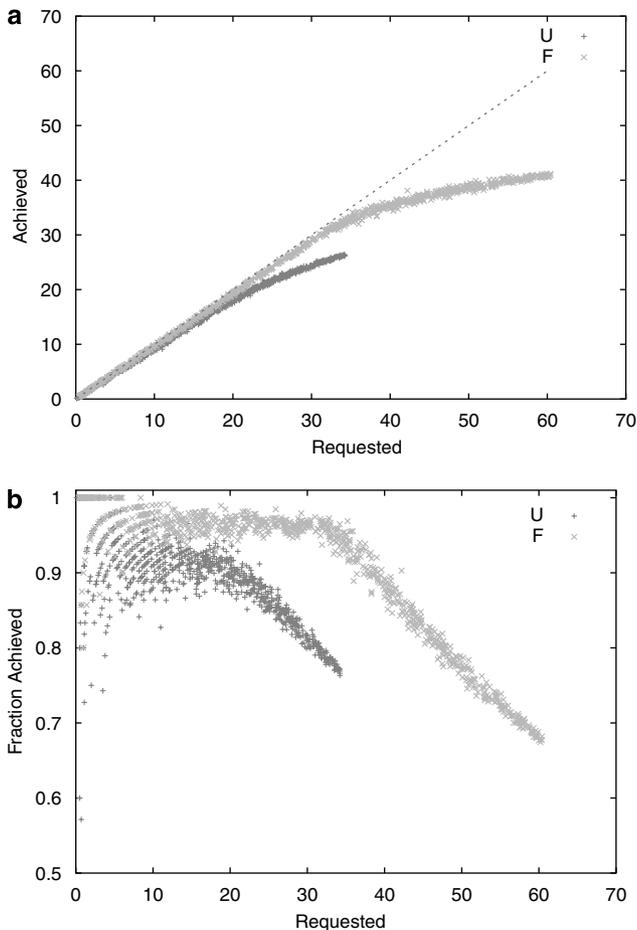


Figure 6 ‘Requested versus Achieved’ Curves for F and U . With Hard L constraint and TT(90,0) for the data set LR(50). (a) Achieved (b) Fraction achieved

to obtain a good approximation to the location of the critical point.

10. Conclusions and future work

The issues of space allocation, space management and space planning, are crucially important to universities in general, and to companies acting as consultants (in particular, to Realtime Solutions Ltd, to which two of the authors are affiliated). Accordingly, we have studied teaching space allocation with two goals in mind. Firstly, we aimed to understand the factors that have the potential to explain the low utilizations and frequencies observed in real-world institutions. Secondly, a requirement was to start to devise methods to determine the safety margins that must be included within space planning methods in order to compensate for the expected low utilization figures when space is actually put into use.

In preliminary studies, it was clear that considering utilization alone gives unrealistic results, in the sense that the realized utilization was too close to the maximum possible utilization. However, if we also include objectives to mimic the effects of timetabling and physical location, and plot the resulting multi-objective trade-off surfaces, then in some regions the utilization falls to much more realistic and observed levels, in the range of 20–40%.

We also introduced a new tool, ‘Achievement *versus* Request curves’, for the analysis of potential room utilizations. The associated methodology detailed in Section 4, is exemplified by the results on Figure 1. We found that when selecting courses at random from a pool then whether or not the selections are fully achievable (‘safe’) becomes statistically predictable. This means that the typical behaviour of different instances can be predicted. Also, the behaviour displays threshold phenomena: There is a critical value of requested utilization below which there is a high probability of satisfying it all, but above which the probability drops sharply.

The intended usage of the results are (i) to build a better understanding of the factors that affect utilization and frequency, a necessary first step to being able to improve them in practice, (ii) to use the statistical predictions of safe regions of U and F in order to give better, more cost-effective, safety margins to be used in space planning.

Together the results give two effects that can lower utilization:

Threshold effects: The finding of a critical utilization placing an upper bound on safe utilization. Moving from free choice to the more realistic fixed choice mode substantially lowered utilization.

Constraints: Location or timetable conflicts further drive down the utilization.

The observation that constraints lower utilization was expected. However, the threshold effects are more interesting.

A positive effect of the thresholds is that the safely achievable utilization might well be predictable—though in a statistical sense rather than absolute sense.

The practical implication of the threshold results is that it gives a methodology to generate a set of rooms with the appropriate safety margins for space planning. Given a set of potential events, the critical utilization will depend upon the set of rooms. Hence, one should pick the best set of rooms for which the critical utilization is at least the requested utilization.

This work has provided an important foundation for a range of research issues that need to be explored. We emphasize that we are developing a methodology, and it is not the details of the algorithm or the exact detailed numbers in the results that are important. For example, we believe that the universality of threshold phenomena in large systems (Bollobas, 1985; Huberman and Hogg, 1987, and others) will lead to wide applicability of the methods.

This work is foundational and so opens many avenues for future work, in particular in extending the model and making the algorithms more efficient. We believe that our model of events and rooms in Section 2 captures enough of the real world for the purposes of this research study (our belief is based on our dealing with universities through Realtime Solutions Ltd). A more complete model would include other effects such as spacetypes and splitting. Rooms often have a ‘spacetype’ that gives their intended usage: lecture, seminar, workshop, etc, and a fuller model would allow the mixing of spacetypes. Courses are typically not single ‘atomic’ events, but instead might need multiple timeslots. Also, courses can need splitting into smaller events, called sections, because they are too large for the rooms or there is a recommended section size. We study this ‘splitting problem’ in Beyrouthy *et al*, 2006a.

A particularly important future research direction is to try and improve methods to generate timetabling constraints. This will rely upon developing statistical characterization of timetabling constraints in an institution, and to do this without relying on details that will not be available at the space planning stage. The main challenge is to see whether useful statistical information can be produced without having to resort to full simulations. Simulations are conceptually straightforward, but often difficult in practice because of the lack of relevant data.

Other important aspects of future work will be to carry out a comprehensive series of comparisons against other real problems. Also, an important aspect of space planning is to determine how ‘room size profiles’—the distribution of room sizes—affect these results. Finally, we note that our current implementations are rather inefficient—many of the graphs here needed many hours of CPU time—and so we will be implementing more efficient methods to produce the trade-off surfaces and achievement *versus* request curves, for example, to use the methods of Gandibleux *et al* (2001) for finding Pareto fronts.

References

- Avella P and Vasil'ev I (2005). A computational study of a cutting plane algorithm for university course timetabling. *J Schedul* **8**: 497–514.
- Bardadym VA (1996). Computer-aided school and university timetabling: The new wave. In: Burke EK and Trick MA (eds). *Selected Papers from the First International Conference on the Theory and Practice of Automated Timetabling (PATAT 95)*. Lecture Notes in Computer Science, Vol. 1153. Springer-Verlag: New York, pp 22–45.
- Beyrouthy C, Burke EK, Landa-Silva D, McCollum B, McMullan P and Parkes AJ (2006a). The teaching space allocation problem with splitting. In: Burke EK and Rudova H (eds). *Proceedings of the Sixth International Conference on the Practice and Theory of Automated Timetabling (PATAT 2006)*. To appear in selected proceedings in Lecture Notes in Computer Science. Springer-Verlag: New York.
- Beyrouthy C, Burke EK, Landa-Silva D, McCollum B, McMullan P and Parkes AJ. (2006b). Understanding the role of UFOs within space exploitation. In: Burke EK and Rudova H (eds). *Proceedings of the Sixth International Conference on the Practice and Theory of Automated Timetabling (PATAT 2006)*. Masaryk University: Czech Republic, pp 359–362.
- Bollobas B (1985). *Random Graphs*. Academic Press: London, England.
- Bosch R and Trick M (2005). Integer programming. In: Burke EK and Kendall G (eds). *Search Methodologies: Introductory Tutorial in Optimization and Decision Support Techniques*. Springer-Verlag: New York, pp 69–95, Chapter 3.
- Burke EK and Petrovic S (2002). Recent research directions in automated timetabling. *Eur J Opl Res* **140**: 180–266.
- Burke EK, Bykov Y, Newall J and Petrovic S (2003). A time-predefined approach to course timetabling. *Yugoslav J Opns Res* **13**: 139–151.
- Carter MW and Laporte G (1998). Recent developments in practical course timetabling. In: Burke EK and Ross P (eds). *Selected Papers from the Second International Conference on Practice and Theory of Automated Timetabling (PATAT 97)*. Lecture Notes in Computer Science, Vol. 1408. Springer-Verlag: New York, pp 3–19.
- Carter MW and Tovey CA (1992). When is the classroom assignment problem hard? *Opns Res* **40**: 28–39.
- Cheeseman P, Kanefsky B and Taylor WM (1991). Where the really hard problems are. *Proceedings of the Twelfth International Joint Conference on Artificial Intelligence, IJCAI-91*. San Mateo, Kaufmann, CA, Morgan: Los Altos, CA, pp 331–337.
- Chiarandini M, Birattari M, Socha K and Rossi-Doria O (2006). An effective hybrid algorithm for university course timetabling. *J Scheduling* **9**: 403–432.
- Cordeau J-F, Jaumard B and Morales R (March 2003) Efficient timetabling solution with tabu search. <http://www.idsia.ch/Files/ttcomp2002/jaumard.pdf>, accessed July 2007.
- Cormen TH, Leiserson CE, Rivest RL and Stein C (2001). *Introduction to Algorithms*, 2nd edn MIT Press, Cambridge, MA and McGraw-Hill: New York.
- de Werra D (1985). An introduction to timetabling. *Eur J Opl Res* **19**: 151–162.
- Deb K (2005). Multi-objective optimization. In: Burke EK and Kendall G (eds). *Search Methodologies Introductory: Tutorial in Optimization and Decision Support Techniques*. Springer: New York, pp 273–316, Chapter 10.
- Di Gaspero L and Schaerf A (2004). A multineighbourhood local search solver for the timetabling competition TTComp-2002. In: *Selected papers from the Fifth International Conference for the Practice and Theory of Automated Timetabling (PATAT 2004)*. Lecture Notes in Computer Science, Springer-Verlag: New York, pp 262–275.
- Dueck G (1993). New optimization heuristics: The great deluge algorithm and the record-to-record travel. *J Comput Phys* **104**: 89–92.
- Fizzano P and Swanson S (2000). Scheduling classes on a college campus. *Comput Optim Appl* **16**: 279–294.
- Gandibleux X, Morita H and Katoh N (2001). The supported solutions used as a genetic information in a population heuristics. In: Zitzler E, Deb K, Thiele L, Coello CAC and Corne D (eds). *Proceedings of the 1st International Conference on Evolutionary Multi-criterion Optimization (EMO 2001)*. Lecture Notes in Computer Science, Vol. 1993. Springer: London, pp 429–442.
- HEFCE (1999). *Estates management statistics project*. Technical Report 99/18, Higher Education Funding Council for England, http://www.hefce.ac.uk/pubs/hefce/1999/99_18.htm accessed July 2007.
- Huberman BA and Hogg T (1987). Phase transitions in artificial intelligence systems. *Artif Intell* **33**: 155–171.
- Kirkpatrick S, Gelatt CD and Vecchi MP (1983). Optimization by simulated annealing. *Science* **220**: 671–680.
- Kostuch P (2004). The university course timetabling problem with a three-phase approach. In: Burke EK and Trick MA (eds). *Selected Papers from the Fifth International Conference on Practice and Theory of Automated Timetabling (PATAT 2004)*. Lecture Notes in Computer Science. Springer-Verlag: New York, pp 109–125.
- Landa-Silva JD (November 2003) *Metaheuristics and multiobjective approaches for space allocation*. PhD thesis, School of Computer Science and Information technology, University of Nottingham.
- McCollum B and McMullan P (2004). *The cornerstone of effective management and planning of space*. Technical report, Realtime Solutions Ltd, Belfast. <http://www.realtimesolutions-uk.com/SpaceManagement.doc>, accessed July 2007.
- McCollum B and Roche T (2004). *Scenarios for allocation of space*. Technical report, Realtime Solutions Ltd, Belfast.
- Metaheuristics Network (2003). International timetable competition 2002. <http://www.idsia.ch/Files/ttcomp2002/> Organised by the Meta-heuristics network, <http://www.metaheuristics.net/> and PATAT 2002 <http://www.asap.cs.nott.ac.uk/patat/patat02/patat02.shtml>, accessed 2 July 2007.
- Mitchell DG, Selman B and Levesque HJ (1992). Hard and easy distributions for SAT problems. In: Rosenbloom P and Szolovits P (eds). *Proceedings of the Tenth National Conference on Artificial Intelligence*. AAAI Press: MenloPark, California, pp 459–465.
- Nemhauser GL and Wolsey LA (1988). *Integer Programming and Combinatorial Optimization*. John Wiley & Sons: New York, NY.
- Parkes AJ (2002). Scaling properties of pure random walk on random 3-sat. *CP '02: Proceedings of the 8th International Conference on Principles and Practice of Constraint Programming*. Lecture Notes In Computer Science, Vol. 2470. Springer-Verlag: New York, pp 708–713.
- Petrovic S and Burke E (2004). University timetabling. In: Leung JY-T (ed). *Handbook of Scheduling: Algorithms, Models, and Performance Analysis*. Chapman Hall/CRC Press: Boca Raton, FL, Chapter 45.
- Ritzman L, Bradford J and Jacobs R (1979). A multiple objective approach to space planning for academic facilities. *Mngt Sci* **25**: 895–906.
- Ross P and Corne D (1996). The phase transition niche for evolutionary algorithms in timetabling. In: Burke EK and Trick MA (eds). *Selected Papers from the First International Conference on the Theory and Practice of Automated Timetabling (PATAT 95)*. Lecture Notes in Computer Science, Vol. 1153. Springer-Verlag: New York, pp 309–324.
- Rossi-Doria O, Samples M, Birattari M, Chiarandini M, Dorigo M, Gambardella L, Knowles J, Manfrin M, Mastrolilli M, Paechter B,

- Paquete L and Stützle T (2003). A comparison of the performance of different metaheuristics on the timetabling problem. In: Burke EK and Causmaecker P (eds). *Selected Papers from the Fourth International Conference on Practice and Theory of Automated Timetabling (PATAT 2002)*. Lecture Notes in Computer Science, Vol. 2740. Springer-Verlag: New York, pp 329–351.
- Schaerf A (1999). A Survey of automated timetabling. *Artif Intell Rev* **13**: 87–127.
- Steuer RE (1986). *Multiple Criteria Optimization: Theory, Computation and Application*. John Wiley: New York.
- Tuytens D, Teghem J, Fortemps P and Nieuwenhuyze KV (2000). Performance of the MOSA method for the bicriteria assignment problem. *J Heuristics* **6**: 295–310.

*Received March 2006;
accepted September 2007 after two revisions*