Manuscript Number:

Title: University Space Planning and Space-type Profiles

Article Type: Application and Methodologies for P&S

Keywords: timetabling; space planning

Corresponding Author: Dr Andrew J. Parkes, Ph.D.

Corresponding Author's Institution:

First Author: Camille Beyrouthy

Order of Authors: Camille Beyrouthy; Edmund K Burke; Barry McCollum; Paul McMullan; Andrew J Parkes

University Space Planning and Space-type Profiles

Camille Beyrouthy, Edmund K. Burke,
School of Computer Science, University of Nottingham
{ cbb , ekb }@cs.nott.ac.uk

Barry McCollum, Paul McMullan
Queen's University, Belfast
{ b.b.mccollum , p.p.mcmullan }@qub.ac.uk

Andrew J. Parkes
School of Computer Science, University of Nottingham
{ ajp }@cs.nott.ac.uk

June 20, 2008

Abstract

Universities planning the provision of space for their teaching requirements, need to do so in a fashion that reduces capital and maintenance costs whilst still providing a high quality level of service. Space plans should aim to provide sufficient capacity without incurring excessive costs due to over-capacity. A simple measure used to estimate over-provision is *utilisation*. Essentially, the utilisation is the fraction of seats that are used in practice, or the ratio of demand to supply. However, studies usually find that utilisation is low, often only 20-40%, and this is suggestive of significant over-capacity.

Our previous work has given methods to improve such space planning. They identify a critical level of utilisation as the highest level that can be achieved whilst still reliably satisfying the demand for places to allocate teaching events. In this paper, we extend such work to incorporate the notions of event-types and space-types. Teaching events have multiple 'event-types', such as lecture, tutorial or workshops, and there are generally corresponding space types. Matching the type of an event to a room of a corresponding space type is generally a soft constraint. However, realistically, allocation happens in a mixed spacetype environment where, teaching events of a given type

1

are allocated to rooms of another spacetype e.g. tutorials will borrow lecture theatres or workshop rooms.

We propose a preliminary model and methodology to quantify the effects of spacetype mixing and establish methods to search for better space-type profiles; where the term "space-type profile" refers to the relative numbers of each type of space. We give evidence that these methods have the potential to improve utilisation levels. Hence the contribution of this paper is twofold. Firstly, we present insightful studies of the effects of space-type mixing on utilisation, and critical utilisations. Secondly, we present novel methods to determine better space-type profiles and we present empirical evidence to justify our claims.

1   Introduction

University managers need to provide space for a wide variety of teaching activities, such as lectures, tutorials, seminars and workshops. They need to do this in a fashion that satisfies proper academic and pedagogical requirements. However, few universities have the luxury to do this without also carefully controlling their costs.

Furthermore, the natural timescales and lifetimes associated with the provision of space range from years for remodelling, to decades for new buildings. In contrast, timetables might easily vary from term (or semester) to term. Accordingly, space provision is rarely possible in a reactive 'just-in-time' fashion, but instead it must be planned significantly in advance, and in the presence of uncertainty as to the exact sets of activities that will need to be accommodated.

The success of such space planning might be measured in various fashions. However, a standard and simple goal would be to estimate the extent to which there was a surplus of space. This can be done by measuring the *'utilisation'*, U. An exact definition of utilisation is given in [Beyrouthy et al., 2007d], but it is essentially the fraction of time that available seats are actually used. Surveys in English Universities [HEFCE, 1999] have found that the utilisation rates are often alarmingly low, such as $20-40\%$. Loosely speaking, rooms are used only half the time and even when in use they are often only half full.

Even more alarmingly, there is little consensus as to whether such low utilisation rates are an inevitable result of the constraints within space planning process, or (more likely) that they can be improved by better (long-term) space planning and better (short-term) space management.

In a series of papers [Beyrouthy et al., 2006, 2007a,d, 2008a, Beyrouthy, 2008], we have been setting the scientific foundations for better space planning and for determining which levels of utilisation ought to be achievable within a given institutional context. That is, we introduced new models, and solution methods, aimed at understanding this low value and at improving the utilisation of teaching space within a university setting.

A key component of these methods are studies of "Achievement Curves" [Beyrouthy et al., 2007d, 2006]. Given a fixed set of room resources, the set of teaching activities are varied and this corresponds to varying the "Requested Utilisation" U_R. For each set of potential activities, a solver is then used in order to select activities to be allocated that maximise the "Achieved Utilisation" U_A. This maximisation of U_A is performed in the presence of appropriate sets of constraints, typically conflict constraints between events, but also, for example, 'location penalties' for placing events in 'incorrect' departments. In addition, the work has also studied the case in which some teaching activities, such as tutorials, require splitting into small groups [Beyrouthy et al., 2007a, 2008a]. Note, it is not our intention to discuss details of course timetabling and associated algorithms here. Instead, we refer the reader to [Burke and Petrovic, 2002, Carter and Laporte, 1998, Rossi-Doria et al., 2003, 2002, Schaerf, 1999], and also to work on course splitting in [Boronico, 2000, Mirrazavi et al., 2003, Selim, 1988]. One should observe that space planning is very different from standard timetabling because it involves multiple time-scales and also because of the need to plan resource capacities rather than taking them as fixed.

We found that the achievement curves can give rise to a 'critical utilisation' U_C that marks the transition or threshold between under- and over-constrained regions. When the requested utilisation is less than critical, $U_R < U_C$ then 'almost always' all the events can be allocated, that is, $U_A = U_R$. Above the critical value, $U_R > U_C$, then it is 'almost never' possible to allocate all events, that is, $U_A < U_R$. Such a threshold between 'almost always' and 'almost never' behaviour is well-known in many systems. See, for example, the case of random graphs [Bollobas, 1985].

The achievement curves encapsulate the potential for space allocation. In particular, within a given institutional context and set of resources and constraints, the critical utilisation gives a measure of the largest utilisation that can be safely achieved. Achieving such critical utilisation can also be computationally expensive as the threshold is also associated with a large peak in the computational cost of solving the allocation problem [Beyrouthy et al., 2008a].

Initial work on space planning focussed on determining the critical util-

3

isation for a given fixed set of (teaching space) resources. However, the natural goal in space planning is then to develop methods to adjust the sets of resources so as to increase the critical utilisation. We will, therefore, be concerned with altering the structure and size of rooms, which we refer to collectively as the *space-profile*. Space planning in academic institutions [Bleau, 1981, Bullock, 1974, Fizzano and Swanson, 2000] is closely related to maintaining and developing a given space profile so that future demand for space will be satisfied.

One particular reason for the need to consider changes to the *space profile* is that it is often the case that it was established, when the estate was first built and made available for teaching. It was hence tailored to the institution needs at the time. Often there was no consideration of future growth and the changes in pedagogical practices; for example, a general move from a few large events to many smaller events. Changes to the nature of events so that they no longer match the space profile will be likely to lead to a loss of utilisation. A new space-profile is therefore required to match the new teaching activity requirements. However, remodelling or rebuilding so as to change the space profile is expensive, and so it is essential to have computer modelling methods to evaluate space profiles, and to create good ones. One might also think of space-type as encoding the room features and so implicitly these studies also relate to the decision as to which teaching facilities a room ought to be given.

Naturally, room sizes can be expected to have a significant effect on utilisation. In a previous paper, we looked at the effects of changing the sets of room sizes [Beyrouthy et al., 2007b]: giving a method to pick a better set of room sizes with a target total capacity, with the result of significantly increasing the critical utilisation. We refer to this as adjusting the "room-size profile".

Besides room sizes, another natural candidate for improvement of utilisation arises from the issue that both events and rooms generally have associated event- and space-types and that it is generally considered desirable to match these up appropriately. Teaching space is constituted of (not restricted to) rooms of different sizes and types, e.g. large tiered rooms (also known as lecture theaters), small tutorial rooms, laboratory rooms, seminar rooms, etc. Rooms, therefore, have a given spacetype associated to them (e.g. lecture, tutorial, workshop, seminar etc.) and generally host academic activities of a similar type. For example, lectures are offered in lecture theaters, tutorials in tutorial rooms, etc. In practice this might well be because the features needed for the rooms differ depending on the teaching activity.

In previous work, we considered each space-type independently and sep-

4

arately. The work in this paper is novel in that it simultaneously considers multiple space-types, and explores their interactions with each other. This allows consideration of the following two important issues.

- **Spacetype mixing.** Although it is desirable to match event-types with space-types this is not necessarily a hard constraint. For example, in practice, and when needed, a small lecture might be placed in a tutorial room, or vice-versa. That is, the space-type matching will usually be a soft constraint, and some limited mixing is allowed. From informal discussions with school managers [McCollum and McMullan, 2004, McCollum and Roche, 2004, Space Management Group (SMG), 2006], events are often assigned rooms of different spacetypes, e.g. tutorials can be offered in lecture theaters if they are available, or seminar and workshop rooms etc.

- **Spacetype Profile.** By this, we loosely mean the resources of each spacetype that are available. For example, the numbers of lecture rooms in comparison to the tutorial rooms.

When space-type mixing is disallowed or tightly restricted, then it can be expected that the space-type profile will have a significant effect on the utilisation. It is the purpose of this paper to extend and enhance previous methods so as to quantify the interaction between space-type mixing and profiles and the achievable utilisations. The work is done in the context of the 'Teaching Space Allocation problem with splitting' [Beyrouthy et al., 2007a, 2008a, Beyrouthy, 2008]. We give evidence that tightly matching spacetypes when performing activities allocation, using the current space-profile, has a detrimental effect on the *utilisation*. This supports the need for the space-type mixing which occurs in many universities.

Hence, this paper has two primary goals:

1. Measuring the effects of permitting spacetype mixing.

2. Providing methods to improve the spacetype profile.

As part of the work we present an integer programming model that combines together

- Space planning variables
- Timetabling variables

To the best of our knowledge, this is the first time that an integer programming model has directly incorporated both these space planning and timetabling aspects.

Outline of the paper: In section 2, we describe the basic allocation problem that we use. Section 3 gives an integer programming formulation, "SPM" of the allocation problem for the case with multiple fixed space-types. Section 4 uses this SPM formulation to illustrate the effects on utilisation of varying the amount of space-type mixing permitted. In section 5, we present two methods to generate new space-type profiles: One method based on an extended integer programming model with dynamic space-types, and the second based on local search. Section 6 gives results showing that altering the space-type profile improves the expected utilisation. Conclusions and a discussion of future work are given in section 7.

2   Problem Description

Space planning has goals that are different than course timetabling, however, they are tightly inter-related. In particular, planning relies on an underlying model of timetabling. Unfortunately, the terminology of course timetabling is varied, and depends on the countries and the academic institutions. (For a detailed description see [Beyrouthy, 2008].) Hence, we now briefly summarise the problem description to fix the terminology we use.

A 'course', in our terminology, generally lasts multiple years and represents the type of study that students embark upon within an institution. Course requirements dictate that students should enroll on modules, which are usually taught once a term (or semester), but several times in a given week: e.g. the module "Programming for CS" is taught on Monday 10:00 am, Wednesday 1:00am, Friday 10:00 pm. Students enrolled in that module should attend *all* of those taught meetings, and such meetings are called classes. In [Beyrouthy et al., 2007a, 2008a], we have allocated classes of a module to roomslots, but when the number of students of a class is larger then the room size, we are required to split it so that it fits. Also, there are often pedagogical reasons in order to split classes, for example, tutorials should 'by definition' be small groups. The resulting "splitting" will ultimately generate new groups of this class. Therefore a student would attend *one of* those groups [Beyrouthy et al., 2007a, 2008a].
Therefore:

For every module, $k \in \{1, \ldots, q\}$ we associate the following:

1. Size S_k : Number of students in module k.

2. Timeslots T_k : Number of timeslots required by the module in a weekly schedule.

3. Department d_k: Department administering module k.

Other aspects, belonging to modules, such as special module features, or module preferences, could also be used. However, we are not considering them in this study.

Classes will carry the same information as their respective modules except for the timeslots and spacetype. For a class $i \in \{1, \ldots, n\}$, we associate the following:

1. Size S_i: number of students of the class (equal to the number of students of the respective module).

2. Type EP_i: lecture, workshop or tutorial, etc.

3. Department d_i: department offering/managing the class.

For every room $j \in \{1, \ldots, r\}$ we have:

1. Capacity c_j: maximum number of students in the room

2. Timeslots T_j: the number of timeslots per week

3. Spacetype SP_j: space for lectures, workshops, tutorials, etc.

4. Department d_j: the one that owns/administers the room.

The hard constraints that are always enforced are:

1. **Capacity constraint**: class/group size cannot exceed room capacity.

2. **No-sharing constraint**: at most one class/group is allowed per "roomslot", where by roomslot we refer to a (room, timeslot) pair.

For the definition of the utilisation and the soft constraints that are used in this model, we refer the reader to [Beyrouthy et al., 2007d,a]. In this paper, however, we also introduce an additional constraint of spacetype matching as follows.

Spacetype Penalty: We use the notion of spacetype penalty, to account for the desire of matching teaching events of a certain type to rooms with the same given spacetype. Fro example, tutorials would need to be assigned to tutorial rooms. We simply set a penalty if the type of the class does not match that of the given room. Specifically, if we assume that a given class i with type EP_i, is assigned to room r with Spacetype SP_j , then there is a penalty matrix represented by $\Gamma(EP_i, SP_j)$ where all entries are non-negative.

In the case we study here, we use lecture (`Lec`), tutorial (`Tut`), and workshop (`Wksp`) for both space and event-types. For the sake of concreteness, we use the following (ad-hoc) matrix for Γ:

7

Event-type \ Space-type	Lec	Wksp	Tut
Lec	0	7	6
Wksp	7	0	5
Tut	6	5	0

Of course any other appropriate weight matrix can also be used. Note that the matrix need not be symmetric, but we do expect that if the types match, $EP_i=SP_j$, then the penalty is zero.

The total spacetype penalty of a given assignment is the sum of this penalty over all rooms and all classes. In summary, this can be considered as a count of space-type mixing violations as we allocate classes and groups to available rooms.

3 The Initial Space Planning Model (SPM)

This section gives an integer programming formulation of the basic problem. It is closely based on the model given in [Beyrouthy et al., 2008a], but extended so as to include multiple space- and event-types.

3.1 Variables and Parameters

The following sets, parameters and variables are used for modeling the problem as an Integer Program (IP). Note that we will call a '(room, timeslot)' pair a roomslot. Roomslots denote the available space to which classes or groups are allocated. So having r rooms and p timeslots per room, the number of roomslots would be rp.

Given sets:

Q : set of all modules, with total number of modules, $q = |Q|$

N : set of all classes, with total number of classes, $n = |N|$

P : set of all timeslots, with total number of timeslots, $p = |P|$

R : Set of all rooms, with total number of rooms, $r = |R|$

M : set of all roomslots, with total number of roomslots, $m = |M| = rp$

D : set of types for both event and space-types, $D = \{\texttt{Lec}, \texttt{Wksp}, \texttt{Tut}\}$

Event and space-types are given by:

$EP_i \in D$: Event-type of class i.

$SP_j \in D$: Space-type of roomslot j.

We also use the following derived and associated sets:

$T^z \subseteq M$ is the set of roomslots corresponding to a given timeslot $z \in P$. The sets T^z are mutually disjoint and give a partition of the roomslots; $M = \bigcup_{z \in P} T^z$.

8

$A^b \subseteq M$ is the set of roomslots corresponding to a given room $b \in M$. The sets A^b are mutually disjoint and give a partition of the rooms slots; $M = \bigcup_{b \in P} A^b$.

$F^k \subseteq N : k \in Q$ is the set of classes belonging to a given module $k \in Q$.

$E^d = \{ i \mid i \in N : EP_i = d\}$ is the set of classes belonging to a given type $d \in D$.

$H^d = \{ j \mid \in R : SP_j = d, d \in D\}$ is the set of rooms belonging to a given spacetype $d \in D$.

Other needed parameters are:

S_i : number of students enrolled in class i

c_j : capacity of roomslot j

$C_{i_1 i_2}$: conflict matrix between classes i_1 and i_2.

L_{ij}: location penalty matrix between classes i and room j

Γ_{ij}: spacetype penalty matrix between classes i and room j

G_i^t: target group size for class i

G_i^{low}: lower limit on group size for class i

G_i^{up}: upper limit on group size for class i

G_i^{nb}: upper limit on number of groups of class i

O^l : minimum occupancy allowed, i.e. minimum fraction
      of room seats to be filled

The location, group-size, and space-type penalties are not treated as objectives, but are thought of, instead, as constraints on solutions. We impose upper bounds on their values

B_L^{up} : upper limits on the location (L)

B_{GZ}^{up} : upper limit on group size (GZ) penalty

B_{SP}^{up} : upper limits on the spacetype penalty (Γ)

    For example, $B_L^{up} = \infty$ will correspond to no limit on locations, whereas $B_L^{up} = 0$ will force no location penalty, i.e. that all locations are perfect matches.

3.2 Decision variables

The primary decision variables are the non-negative integers:

$v_{ij} =$ the number of students of class $i \in N$ allocated to roomslot $j \in M$

Derived variables are

$$y_{ij} = \begin{cases} 1 & \text{if one group of class } i \text{ is allocated to roomslot } j. \\ 0 & \text{Otherwise} \end{cases}$$

9

$$x_i = \begin{cases} 1 & \text{if class } i \text{ is allocated.} \\ 0 & \text{Otherwise} \end{cases}$$

$$z_k = \begin{cases} 1 & \text{if module } k \text{ is allocated.} \\ 0 & \text{Otherwise} \end{cases}$$

3.3 Objective function

The objective to be maximised is the overall *seat-hours*:

$$Obj = \left(\sum_{i=1}^{n} \sum_{j=1}^{m} v_{ij} \right) \tag{1}$$

The utilisation U is defined as

$$U := \frac{\text{Seat-hours used}}{\text{Total seat-hour capacity}} = \frac{Obj}{\left(\sum_{i=1}^{n} \sum_{j=1}^{m} c_j \right)} \tag{2}$$

So, for a given set of events, maximising the objective corresponds to maximising the utilisation.

3.4 Constraints

Maximisation of the utilisation is subject to the following constraints.

Given that partial allocation is not allowed, we enforce

$$\sum_{j=1}^{m} v_{ij} = S_i x_i \qquad \forall i \tag{3}$$

Room capacities cannot be exceeded, and so we impose

$$v_{ij} \leq c_j y_{ij}, \quad \forall i \in N, j \in M; \tag{4}$$

This also links the v and y decision variables.
Only one group can be allocated to a given roomslot:

$$\sum_{i=1}^{n} y_{ij} \leq 1, \quad \forall j \in M \tag{5}$$

The location penalty must be less than the upper limit B_L^{up}:

$$\sum_{i=1}^{n} \sum_{j=1}^{m} L_{ij} y_{ij} \leq B_L^{up} \tag{6}$$

When $B_L^{up} = 0$, hard location penalty is enforced.

To limit the group size (GZ) penalty for workshops we impose

$$\sum_{i=1}^{|E^2|} \sum_{j=1}^{m} \left| v_{ij} - G_i^t y_{ij} \right| \leq B_{GZ}^{up} \tag{7}$$

Constraint 7 is not required for tutorials because hard limits are imposed on their group size and it is also not required for lectures since they do not split.

To enforce the spacetype mixing penalty we impose:

$$\sum_{i=1}^{n} \sum_{j=1}^{m} \Gamma_{ij} y_{ij} \leq B_{SP}^{up} \tag{8}$$

We also impose an upper and lower limit on the group sizes for tutorials using:

$$v_{ij} \leq G_i^{up} y_{ij} \qquad \forall i \in E^3, j \in M \tag{9}$$

$$v_{ij} \geq G_i^{low} y_{ij} \qquad \forall i \in E^3, j \in M \tag{10}$$

Given that partial allocation of a module is not allowed, classes of a module should either all be allocated or none, we enforce:

$$\sum_{i=1}^{|F^k|} x_i = |D| z_k, \quad \forall k \in Q; \tag{11}$$

We impose upper limits on the number of groups per class:

$$\sum_{j=1}^{m} y_{ij} \leq G_i^{nb}, \quad \forall i \in E^2 \tag{12}$$

Since in our model, lectures do not split, we impose, $G_i^{nb} = 1$ for all $i \in E^1$. Note that if $G_i^{nb} = 1$ for all $i \in N$, the problem becomes the pure teaching space allocation problem, without splitting [Beyrouthy et al., 2007d].

If a roomslot is used then the given fraction O^l of room seats needs to be filled:

$$v_{ij} \geq O^l c_j y_{ij}, \quad \forall i \in E^2, j \in M \tag{13}$$

11

This is relevant only for `Workshops`. In this paper we use $O^l = 0.3$. (We investigated other values and found that smaller values, or turning off this constraint altogether, do not significantly change the utilisation results we present here, but retain it as in general it will lead to avoiding the placement of events in uncomfortably large rooms.) The following constraint is entailed by the other constraints but we added it as it lead to a considerable reduction in computation times.

$$\sum_{i=1}^{n}\sum_{j=1}^{m} y_{ij} \leq rp \tag{14}$$

Finally we impose timetable conflict constraints. Lectures, workshops and tutorials from within a given module should not be allocated to the same timeslot:

$$\sum_{i=1}^{|F^k|}\sum_{j=1}^{|T^\theta|} y_{ij} \leq 1 \quad \forall k \in Q, \theta \in P; \tag{15}$$

This is natural for lectures, but also assumed for tutorials, on the grounds that they are all likely to be be given the same teacher, who cannot be in two places at the same time.

The issue of conflicts between events from different modules is more difficult. Firstly, the conflict matrix $C_{i_1 i_2}$ between different classes of a lecture i_1, i_2 is generated using the student enrollment generator described in [Beyrouthy, 2008, Beyrouthy et al., 2007c]. Constraint 16 imposes that no two lectures with common students be allocated to the same timeslots:

$$C_{i_1 i_2}(y_{i_1 j_1} + y_{i_2 j_2}) \leq 1, \quad \forall \theta \in P, \forall j_1 \in T^\theta, \forall j_2 \in T^\theta,$$
$$\forall i_1 \in E^1, \forall i_2 \in E^1$$
$$i_1 \neq i_2, j_1 \neq j_2; \tag{16}$$

However, we do not impose conflicts between groups associated with tutorials and workshops. Generally, it is expected that splitting for these event-types has the property of reducing conflicts between modules, and so we do not need to include them. This issue, which we refer to as *"partial conflict inheritance"*, is discussed elsewhere [Beyrouthy et al., 2007c, Beyrouthy, 2008].

The model above is what previous work [Beyrouthy et al., 2007d] refers to as "free choice" mode because the solver has the option of whether or not to allocate an event [Beyrouthy et al., 2007d]. Although, not used in this
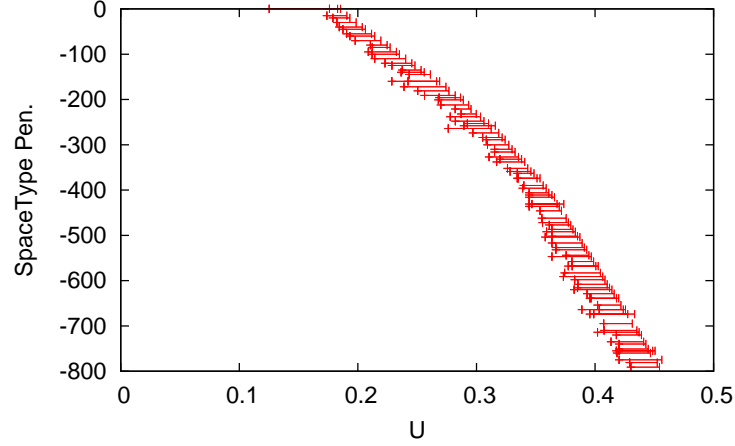
12

Figure 1: Spacetype mixing effects on utilisation on a single instance and obtained using SPM. Error bars represent the integer solutions and their respective upper bound from the linear relaxation.

paper, we could also employ a "fixed choice" mode in which all events must be allocated. In this case, we simply enforce $\forall i.\ x_i = 1$, giving

$$\sum_{j=1}^{m} v_{ij} = S_i \qquad \forall i \tag{17}$$

The value of the objective Obj is then fixed, and the problem is only the determination of feasibility.

4   The Effects of Spacetype Mixing

The model SPM can be immediately used to explore spacetype mixing before we move on to the issue of changing the space-type profile. Figure 1 is a plot of utilisation versus spacetype penalty for a typical "real-world" instance. We use a small dataset with only 15 timeslots, with large events and small rooms, creating a large number of splits per module. The error bars represent the actual integral value of the utilisation and the upper bound on the utilisation value resulting from the linear relaxation of model SPM.

Relaxing the upper limit on mixing violations allows the utilisation to increase from about 18% to about 43%. In this case, the large number of tutorial groups generally require more tutorial roomslots then are available.

13

As we force groups to match their respective spacetypes, there is a consequent effect on the utilisation, as seen in this figure. In practice, most universities allow some mixing of spacetypes to counter balance this loss.

5   Adapting Space-profiles

The model, SPM, of section 2 used a fixed space-profile determined by parameters SP_j. In this section, we give two methods that let the solver choose the space-type profile: They both select the space-types of the rooms in such a way as to maximise utilisation. The first method uses a model that is a modification of the integer programming model of the previous section. The second method relies on local search to change the space-type profile. Notice that neither method changes the set of room sizes at all. In practice, this might correspond to remodelling that does not change the walls (neither external nor internal), but can instead change the internal features and layout of the rooms.

5.1   Extended Model: EXT-SPM

The model EXT-SPM modifies SPM by dropping the fixed parameters SP_j and the matrix Γ_{ij} and by introducing two decision variables:

$$w_{\rho\sigma} = \begin{cases} 1 & \text{if room } \rho \text{ is assigned spacetype } \sigma. \\ 0 & \text{Otherwise} \end{cases}$$

and the derived variable:

$$u_{j\sigma} = \begin{cases} 1 & \text{if roomslot } j \text{ is assigned spacetype } \sigma \text{ .} \\ 0 & \text{Otherwise} \end{cases}$$

We also add the following constraints.

Precisely one spacetype is assigned to any room:

$$\sum_{d=1}^{|D|} w_{bd} = 1, \quad \forall b \in R \tag{18}$$

If a room is assigned a given spacetype, then all of its respective roomslots are assigned that spacetype:

$$\sum_{j=1}^{|A^b|} u_{jd} = p w_{bd}, \quad \forall b \in R, \forall d \in D; \tag{19}$$

14

That is, when adapting the space-profile, we alter the spacetype of all room-slots of a room, e.g. in this model, we do not allow a room to be of tutorial type half of the week and a workshop the other half of the week.

Linking variables u to y, imposes that if a roomslot is assigned a given spacetype d, then that roomslot can only be assigned classes of type d.

$$y_{ij} \leq u_{jd}, \quad \forall d \in D, \forall i \in E^d, \forall j \in M; \qquad (20)$$

That is, no space-type mixing is permitted.
Only one spacetype is assigned to any roomslot:

$$\sum_{d=1}^{|D|} u_{jd} \leq 1, \quad \forall j \in M \qquad (21)$$

All other constraints in SPM remain unchanged in the extended model.

5.2 LS-SPM: Local search in Space-profiles

For smaller instances and for finding upper bounds on a given solution, the integer programming approach will be sufficient. However, we also start to introduce heuristic methods so as to move towards tackling large instances that are intractable by an exact solver.

Here, the space-type alteration is performed using a local-search-like method. The neighbourhood operators we use are:

- **Swap-type:** OP^1, Randomly select 2 different rooms with two different spacetypes, and swap their spacetypes. For example, if selected room 1 is a lecture room and room 2 is a tutorial room, then after applying this operator, room 1 becomes a tutorial room and room 2 a lecture room.

- **Alter-type:** OP^2, Randomly select a room and alter it's spacetype.

The following procedure, LS-SPM, is a straightforward hill-climbing that uses these operators to change the space-type profile in order to increase the utilisation.

Proc: LS-SPM
1 Set $Obj_1 = 0$;
2 Load Rooms vector : Ro_i
3 load current Rooms vector : $Ro_c = Ro_i$
4 Load empty room vector Ro_d
5 Load modules vector E_i

15

6 LOOP 1 to `Tcriteria` or `No improvement`
7   Randomly select operator OP^X, with $X \in \{1,2\}$
8   apply operator to rooms vector(Ro_c): $Ro_d = OP^X (Ro_c)$
9   run CPLEX with Model `SPM` on (E_i,Ro_d)
10   get $Obj^*= Max(U)$
11   IF $Obj^* \geq Obj1$
12          $Obj_1 = Obj^*$
13   assign Ro_d vector to Ro_c: $Ro_c = Ro_d$
14   END IF
15 END LOOP
16 Return space-profile: Ro_c

Starting with a fixed instance of events and rooms, the procedure applies a perturbation/move using operators OP^{1-2} to the space-type vector for the rooms. Then using the current event instance, it solves model `SPM` (we use CPLEX 10[1]), and finds the maximum utilisation Obj^*. If Obj^* is larger or equal to incumbent utilisation then the move is accepted, otherwise it is rejected and another move tried. Also, of course, it could easily be extended to use a meta-heuristic such as simulated annealing. Note that there is an implicit assumption that it will be given a set of events that is large enough that the maximum utilisation will be less than 100%; otherwise the landscape will be flat, precluding meaningful hill-climbing.

This procedure is faced with the difficult task of optimising model `SPM` so that the run-time is reduced. This procedure is performed over 6 hours of computation time, owing to the difficulty of quickly solving the model with the integer programming solver. However, eventually, the exact solver used to calculate the achieved utilisation could be replaced by an appropriate meta-heuristic solver; provided that it is sufficiently robust in the sense of the performance not being sensitive to the space-types used. Otherwise, biases would lead it to favouring some space-types over others.

6 Results

In this section, we present results of studies on the effect on utilisation of using the methods of the previous section to improve the space-type profile. The instance used has 20 modules having one lecture each, and various workshops and tutorials. We do not give full details as the point of this
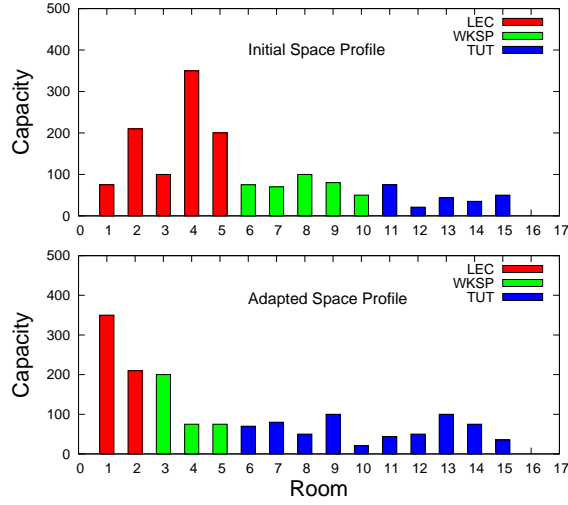
---

[1]http://www.ilog.fr

Figure 2: Results of adapting the space-type profile using procedure `LS-SPM` starting with a real-world instance.

paper is not to study any one particular case in depth, but rather to show the methodology at work and to give some illustrative results.

6.1 The Adapted Space-type Profiles

Figure 2 is a histogram comparing the initial and adapted space-profile using procedure `LS-SPM`. Notice that the algorithm "selects" small-sized rooms and changes their space-type to tutorials. Leaving workshop and lecture space-types for the largest rooms. The total number of roomslots suffice to fit all lectures in just two larger rooms. (Note that the method is rather time-consuming, owing to running the `SPM` model in every iteration. However, since space planning is only done rarely then this is not a major practical issue in practice. Similarly, in figure 3, we compare the initial and adapted space-profile using the exact model `EXT-SPM`. The adapted profile is very similar to that produced by the local search of figure 2.

As intended, the space-type profile was adapted to the available event profiles. In this case, because tutorials require smaller but more rooms as opposed to lectures which require more seats per room but fewer rooms.
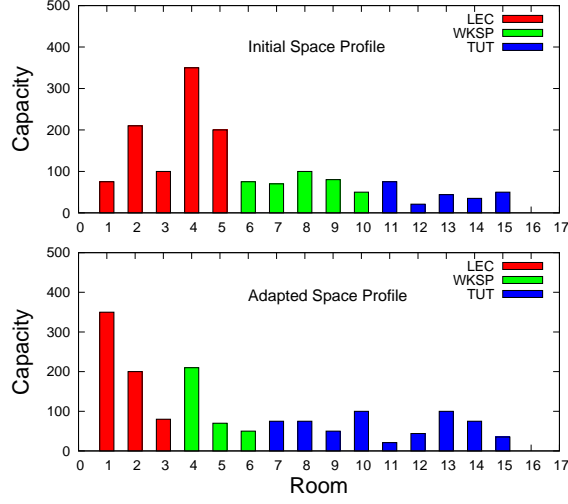
17

Figure 3: Results of adapting the space-type profile using model `EXT-SPM` and the same real-world instance as figure 2.

6.2   Achievement Curves with the Adapted Profiles

We have given methods to change the space-type profile. However, the key test of their effectiveness is to see whether or not they improve the critical utilisation. Hence, we now create and plot the achievement curves [Beyrouthy et al., 2007d] for the adapted space-type profiles.

We will refer to the original space-type profile as the "static" case. The space-type profiles we just produced by adapting to a single instance we will refer to as "fixed adapted". Note that the "fixed" is not meant in the sense of "repaired" but rather in the sense of "unchanging"; due to the fact that when producing the achievement curve we use the same space-type profile for all the random subsets.

Figure 4 presents the achievement curves we obtained. Firstly, it gives the achievement curve for the original static space-type profile. The critical utilisation is about 19%: that is, if set of candidate events is such that $U_R < 19\%$ then almost always they can be all allocated. On the other hand, if $U_R > 19\%$ then almost always some event(s) will need to be left unallocated. Secondly, figure 4 gives the curve for the space-type profile produced with procedure `LS-SPM`. This time, the critical utilisation is about 29%.

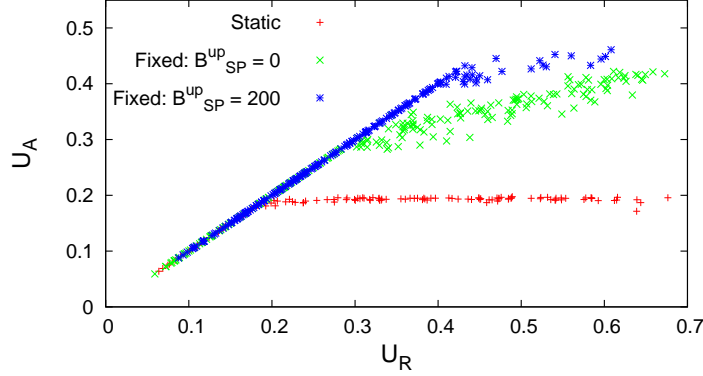In both of these cases, space-type mixing was forbidden when producing

18

Figure 4: Achievement curve of the requested utilisation U_R versus U_A, for three cases: `Static` and `Fixed-adapted` both with $B_{SP}^{up} = 0$, and `Fixed-adapted` with $B_{SP}^{up} = 200$

the curves, that is $B_{SP}^{up} = 0$. To illustrate its benefits, the last curve is when some space-type mixing is permitted, by relaxing the upper bound on the penalty to $B_{SP}^{up} = 200$. The critical utilisation improves considerably to about 40%.

Notice that, in all cases, the values of the achieved utilisation U_A, for a given utilisation request, are grouped around the mean: Changes in U_A between points near to some value of U_R are small compared to the value of U_A itself. This implies that properties of the system are statistically predictable.

The key observation is that the adapted space-profiles have improved the utilisation to about 29% up from about 18% for the static model, corresponding to an increase of over 50% in the effective capacity.

6.3 Dynamically Adjusted Profiles

In the previous section, the space-type profile was adapted by using some typically target instance, but then fixed as we produce the achievement curves. That is, each subset generated to produce each (U_R, U_A) point on the achievement curve plot was given the same space-type profile. However, we can also separately optimise the space-type profile for each subset. That is, in this "`dynamic-adapted`" case, we adapt the space-profile for every point on the achievement curve. Clearly, average utilisation achieved by the `dynamic-adapted` case, provides an upper bound on the `fixed-adapted`
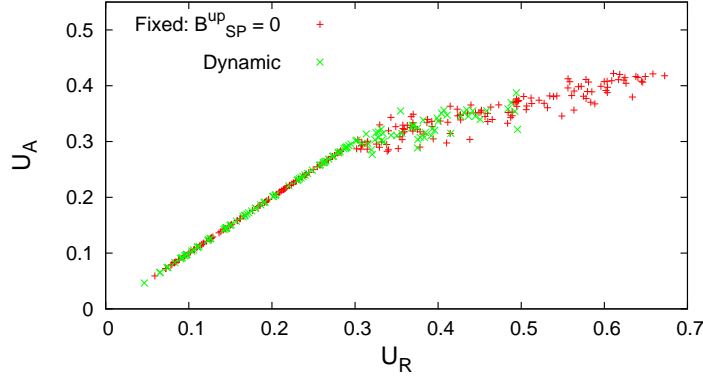
19

Figure 5: Achievement curve of the requested utilisation U_R versus achieved utilisation U_A, for `Fixed-adapted` and the `Dynamic-adapted` models, both with spacetype mixing forbidden, $B_{SP}^{up} = 0$.

	Mean			Std-dev		
	Lec	Wksp	Tut	Lec	Wksp	Tut
Static	61.5	76.2	51.5	6.0	6.3	3.2
Fixed Adapt	50.5	76.3	33.9	6.6	7.4	4.2
Dynamic Adapt	56.0	76.6	34.8	10.4	12.6	7.0

Table 1: Average occupancy (percentages) and standard deviation, for the space-type profiles: the original `Static` profile, and the two adapted profiles `Fixed` and `Dynamic`.

case. Although, it would not be practical to change the space-type profile for every set of events, it is still useful to compare these two. Hence, in figure 5 we compare the achievement curves for the `Fixed-adapted` and `Dynamic-adapted` cases. Somewhat surprisingly, the difference is not significant. That is, once the space-type profile is fixed on the initial scenario, then it generally does not help a lot to adjust it further per instance.

6.4 Occupancy versus Frequency

So far, we have only looked at the utilisation, but other secondary measures of space usage can also provide insight. As described in [Beyrouthy et al., 2007d], the occupancy, O, is measured for every solved instance as

20

the average over all rooms, of seat used over available seats. The frequency, F, is simply the fraction of available roomslots that are used. Essentially speaking, they are related to the utilisation, U, by $U = FO$.

Table 1 presents the average occupancy per spacetype, for the `static`, `fixed` and `dynamically` adapted cases, as well as the standard deviation for each of three cases. We first notice that tutorials, expectedly, have the lowest occupancy. This is presumably due to the "Group-size" constraint being in a range (15-20) less then the average room size. In contrast, the free split in the case of workshops gave a much large occupancy.

However, possibly unexpectedly, as we alter the space-profile, the occupancy tends to decrease, mainly in the case of tutorial and lectures. The results suggest that the space-profile improvement has dramatically improved the frequency of usage but not the room occupancy. Although the results might be seen as counter-intuitive, we believe this is because the fixing of the space-type profile permits more events to be scheduled, increasing the frequency, though at the cost of lower-quality match, on average, between the event and room sizes. We also remark that this implies that trying to increase the occupancy alone without considering overall utilisation could have unintended and damaging consequences. We believe that the best approach to improve occupancy is to alter the *room-size* profile, and this has been studied in [Beyrouthy et al., 2007b].

7   Conclusions and Future Work

The issues stemming from space planning are of increasing importance to universities [McCollum and McMullan, 2004, McCollum and Roche, 2004, Space Management Group (SMG), 2006]. This paper has shed light on aspects related to the planning of teaching space in academic institutions, particularly, on issues relating to the space-types of these rooms, and the "space-type profile", the relative numbers of each type of space. (A space-type represents the facility that is offered by any given room e.g. lecture theater, seminar, or tutorial room.)

The overall goal is to tailor the space-type profile to fit the uncertain demand, and to study resulting improvements in the critical utilisation. With this goal, we have presented a new model and a heuristic method to alter the space-type profile. This extends previous research and available models in space allocation [Beyrouthy et al., 2006, 2007d,a, 2008a], where the space-type profile was fixed.

We addressed the issues using a methodology that takes a typical in-

21

stance of current events, and using either of two distinct methods, alters the space-type profile in a manner that best matches that instance. That is, the procedure is basically as follows:

1. Given the initial (current) space-type profile, generate the achievement curve allowing an estimation of the initial critical utilisation.

2. Using an extended IP model or a local-search-like method, and a typical instance of teaching events we alter the current space-type profile and generate a new profile that optimises the utilisation for that specific instance. These methods are novel in that their output is an allocation of space-types to the rooms, rather than only an allocation of events.

3. Using the newly created space-type profile, we generate another achievement curve, and so estimate the new critical utilisation.

We demonstrated that such adjustment to the space-type profiles has the potential to significantly improve space utilisation. For example, in a case used in this paper the critical utilisation was improved from about 20% to 30%, that is, a 50% improvement.

With respect to future work, one can think of a full *space profile* as consisting of two components:

- **Space-type profile**. The numbers of each space type.
- **Room-size profile**. The numbers of each size of room.

Our previous work on room sizes [Beyrouthy et al., 2007b] worked in a single space-type and adjusted the room-size profile. In contrast, the work here fixed the set of room-size profile but adjusted the space-type profile. Also, in section 5, we have used just one scenario when searching for the fixed adapted space-profile. However, the work in [Beyrouthy et al., 2007b] showed that it gives better results to use a stochastic programming approach and simultaneously optimize with respect to multiple scenarios. In future work, we intend to combine these and so improve the space profile as a whole; that is, to simultaneously adjust both room-size and space-type profiles.

Also, currently, the allocation problems are solved with a straightforward integer programming (IP) model and this can only handle problems of rather limited size. In future work, we plan to improve the IP model, and also to develop meta-heuristic solvers that are tailored to solving the combination of timetabling and adjustment of space-type profiles.

Finally, another goal is for studies to include rooms that can be partitioned (e.g. using a movable partition) on demand for some specific timeslots, meeting various teaching demands. For example, a room usually used

22

for lectures could for partitioned for a few days a week to host smaller tutorials. Such partitioning might hence be regarded as a special case of spacetypes, and ultimately space planning should take account of this possibility; preliminary studies are given in [Beyrouthy et al., 2008b]. Combining all these steps leads towards understanding the full complexity of space planning and management issues.

References

C. Beyrouthy. *Models, Solution Methods and Threshold Behaviour for the Teaching Space Allocation Problem*. PhD thesis, School of Computer Science, University of Nottingham, 2008. (Submitted). http://www.cs.nott.ac.uk/∼cbb/.

C. Beyrouthy, E. K. Burke, J. D. Landa-Silva, B. McCollum, P. McMullan, and A. J. Parkes. Understanding the role of UFOs within space exploitation. In *Proceedings of the 2006 International Conference on the Practice and Theory of Automated Timetabling (PATAT 2006)*, pages 359–362, Brno, Czech Republic, 2006. URL http://www.asap.cs.nott.ac.uk/publications/pdf/ajp_patat2006.pdf.

C. Beyrouthy, E. K. Burke, D. Landa-Silva, B. McCollum, P. McMullan, and A. J. Parkes. The teaching space allocation problem with splitting. In E. K. Burke and H. Rudova, editors, *Practice and Theory of Automated Timetabling VI: Revised Selected papers from the 6th international conference, PATAT, Aug 30-Sep 1, 2006*, volume 3867 of *Lecture Notes in Computer Science*, pages 228–247. Springer-Verlag, Brno, Czech Republic, 2007a. ISBN 978-3-540-77344-3. URL http://www.asap.cs.nott.ac.uk/publications/pdf/ccb_patat2006.pdf.

C. Beyrouthy, E. K. Burke, D. Landa-Silva, B. McCollum, P. McMullan, and A. J. Parkes. Improving the room-size profiles of university teaching space. Technical Report Submitted to Annals of Operations Research, School of Computer Science & IT, University of Nottingham, 2007b.

C. Beyrouthy, E. K. Burke, D. Landa-Silva, B. McCollum, P. McMullan, and A. J. Parkes. Partial inheritance in the teaching space allocation problem with splitting. Working Paper, NOTTCS-TR-2007-6, 2007c.

C. Beyrouthy, E. K. Burke, J. D. Landa-Silva, B. McCollum, P. McMullan, and A. J. Parkes. Towards improving the utilisation of university teaching space. *J Oper Res Soc*, 2007d. ISSN 0160-5682. URL

`http://dx.doi.org/10.1057/palgrave.jors.2602523`. Available online, awaiting volume assignment.

C. Beyrouthy, E. K. Burke, D. Landa-Silva, B. McCollum, P. McMullan, and A. J. Parkes. Threshold effects in the teaching space allocation problem with splitting. *Accepted for publication in the European Journal of Operational Research (EJOR)*, 2008a.

C. Beyrouthy, E. K. Burke, B. McCollum, P. McMullan, and A. J. Parkes. Evaluating the space planning benefits of partitionable rooms. In *Proceedings of the 7th International Conference on the Practice and Theory of Automated Timetabling (PATAT 2008)*, 2008b. To appear.

B. Bleau. Planning models in higher education: Historical review and survey of currently available models. *Higher Education*, V10(2):153–168, Mar. 1981. URL `http://dx.doi.org/10.1007/BF00129129`.

B. Bollobas. *Random Graphs*. Academic Press, London, England, 1985.

J. Boronico. Quantitative modeling and technology driven departmental course scheduling. *Omega*, 28(3):327–346, June 2000.

N. Bullock. Modeling the demand for teaching space. *Environment and planning*, 1:69–103, 1974.

E. K. Burke and S. Petrovic. Recent research directions in automated timetabling. *European Journal of Operational Research (EJOR)*, 140(2):266–280, 2002. URL `http://www.asap.cs.nott.ac.uk/publications/pdf/ejor02.pdf`.

M. Carter and G. Laporte. Recent developments in practical course timetabling. In *Selected papers from the second International Conference on Practice and Theory of Automated Timetabling*, pages 3–19. Springer-Verlag, 1998. ISBN 3-540-64979-4.

P. Fizzano and S. Swanson. Scheduling classes on a college campus. *Computational Optimization and Applications*, 16:279–294, 2000.

HEFCE. Estates management statistics project. Technical report, Higher Education Funding Council for England, March 1999. Report 99/18. http://www.hefce.ac.uk/pubs/hefce/1999/99_18.htm.

B. McCollum and P. McMullan. The cornerstone of effective management and planning of space. Technical report, Realtime Solutions Ltd, Belfast,

24

Jan 2004. http://www.realtimesolutions-uk.com/SpaceManagment.doc, accessed July 2007.

B. McCollum and T. Roche. Scenarios for allocation of space. Technical report, Realtime Solutions Ltd, Belfast, 2004. http://www.realtimesolutions-uk.com/Scenarios.doc.

S. K. Mirrazavi, S. J. Mardle, and M. Tamiz. A two-phase multiple objective approach to university timetabling utilising optimisation and evolutionary solution methodologies. *Journal of Operations Research Society (JORS)*, 54(11):1155–1166, 2003.

O. Rossi-Doria, C. Blum, J. Knowles, M. Sampels, K. Socha, and B. Paechter. A local search for the timetabling problem. In *Proceedings of the Fourth International Conference on the Practice and Theory of Automated Timetabling (PATAT 2002)*, pages 124–127, Gent, Belgium, 2002.

O. Rossi-Doria, M. Sample, M. Birattari, M. Chiarandini, M. Dorigo, L. Gambardella, J. Knowles, M. Manfrin, M. Mastrolilli, B. Paechter, L. Paquete, and T. Stützle. A comparison of the performance of different metaheuristics on the timetabling problem. In *Selected papers from the Fourth International Conference on the Practice and Theory of Automated Timetabling (PATAT 2002)*, volume 2740, pages 329–351, Gent, Belgium, 2003. Lecture Notes in Computer Science, Springer-Verlag, London, UK. URL `citeseer.ist.psu.edu/article/rossi-doria02comparison.html`.

A. Schaerf. A survey of automated timetabling. *Artif. Intell. Rev.*, 13(2):87–127, 1999. ISSN 0269-2821. doi: http://dx.doi.org/10.1023/A:1006576209967.

S. M. Selim. Split vertices in vertex colouring and their application in developing a solution to the faculty timetable problem. *The Computer Journal*, 31(1):76–82, 1988.

Space Management Group (SMG). Space utilisation: practice, performance and guidelines. Technical report, UK Higher Education Space Management Project, 2006.

25