# Title Page:

**Multiple criteria mixed-integer programming for incorporating multiple factors into the development of master operating theatre timetables**

M. L. Penna,\*, C. N. Pottsb, P. R. Harperc

a National Institute for Health Research Collaboration for Leadership in Applied Health Research and Care, University of Southampton, Southampton General Hospital, Mailpoint 11, AA72, South Academic Block, Tremona Road, Southampton, SO16 6YD UK

b Mathematical Sciences, University of Southampton, Southampton, SO17 1BJ, UK

c Cardiff School of Mathematics, Cardiff University, Senghennydd Road, Cardiff, Wales CF24 4AG, UK

\* Corresponding author: m.l.penn@soton.ac.uk, +44 (0) 23 8120 8314

**Keywords**

* (O) OR in health services
* Timetabling
* Operating theatres
* Multiple criteria Integer Programming

# Multiple criteria mixed-integer programming for incorporating multiple factors into the development of master operating theatre timetables

**Abstract**

*Operating theatres and surgeons are among the most expensive resources in any hospital, so it is vital that they are used efficiently. Many European hospitals implement block scheduling, where each surgeon is assigned blocks of time in specific operating theatres on a cyclic basis. This paper proposes a model that assists hospitals in creating new master theatre timetables, which take account of reducing the maximum number of beds required, surgeons’ availability, surgeons’ preferences, variations in types of theatre and their suitability for different types of surgery, limited equipment availability, and the ability to vary the length of the cycle over which the timetable is repeated. The weightings given to each of these factors can be altered, thereby allowing exploration of a variety of possible timetables. Novel features of the model include consideration of surgeons’ preferences for slots, smoothing of bed usage during the generation of master theatre timetables and the use of operating theatres with the potential for the same theatre to be belong to multiple non-nested types. These new features are considered in combination with a range of other factors that have been considered in previous studies on the development of master theatre timetables.*

## 1. Introduction

Many European hospitals use master theatre timetables to allocate blocks of operating theatre time either at specialty level or to specific surgeons. These timetables are usually repeated every week, two weeks or as a monthly cycle. They may be kept with only minor alterations such as those resulting from staffing changes until a change of set up, such as the addition of a new theatre, requires the development of a new timetable. This means that issues such as spikes in demand for a particular type of bed that result from the timetable can have significant ongoing effects.

This paper describes how a multiple criteria mixed-integer linear programming model can be used to produce new timetables quickly, taking account of a range of important factors. The model allows hospitals to explore the relative importance of the different factors by changing their weightings in the objective function. For example, if a surgeon expresses a wish for having operations only on one particular day of the week, can this request be accommodated without a noticeable effect on the effectiveness of the system? In a similar way, potential changes to the constraints can be evaluated.

A number of studies have addressed the development of master surgical timetables (or schedules). Belien and Demeulemeester (2007) define a master surgical schedule as “a cyclic timetable that defines the number of operating rooms available, the hours that the rooms will be open, and the surgical groups or surgeons who are to be given priority for the operating room times”. Similar definitions are also given by Blake et al. (2002), Blake and Donald (2002), Santibanez et al. (2007), Testi et al. (2007), Oostrum et al. (2008), Belien et al. (2006), Hans et al. (2007, 2008) and Spratt and Kozan (2016). Agnetis et al.’s (2014) introduce a modified block scheduling approach in the construction of master timetables where some blocks are flexible within the schedule and can be assigned to different surgeons depending on demand.

All of the papers on master surgical timetables consider the amount of operating time available and the amount required by each specialty (surgical group) or surgeon. The literature review of Cardoen et al. (2010) on operating theatre scheduling includes a table that lists the other aspects considered including wards, ICUs (intensive care units), equipment, surgical staff, the budget, and operating theatre overtime/undertime. The various studies relating to theatre timetabling have considered different combinations of these, usually as constraints on the timetables generated. There are other factors that are considered in some studies. For example, the only study we have found that gives explicit consideration to surgeons’ preferences for particular days or times of day is that of Santibanez et al. (2007). Thus, there are a significant number of factors involved in theatre scheduling, and we need to decide which to include in our approach to this problem.

Cardoen et al. (2010) identify a substantial number of publications that take account of the stochastic nature of at least one aspect of theatre scheduling, although they consider all levels of theatre scheduling and not just master surgical timetable development. Many of these studies incorporate stochastic aspects by using simulation techniques, such as those of Arenas et al. (2002), Bowers and Mould (2004) and Testi et al. (2007).

The majority of studies regarding the development of master surgical timetables do not mention implementation in real hospitals, or only state that data have been obtained from hospitals, so it is unclear if they have been implemented. A notable exception to this is the work of Blake and Donald (2002), which has been in use at Mount Sinai hospital since 1996. The model in question assigns theatre time to surgical divisions when changes to the current allocation are required. The model is relatively straightforward, which implies that perhaps some models that have been developed may not be implemented because they are too complex for hospital staff to understand. The reduced complexity of the model needs to be balanced between the need to model the system accurately and to take into account sufficient factors within the model.

Our study of the literature identifies various factors of importance in theatre scheduling that are not considered routinely in previous studies and should therefore be incorporated into future models that are developed. One such factor is the ability to smooth bed usage, while taking account of weekly variations in bed availability.  Other factors concern the characteristics of operating theatres that, contrary to the assumption in most studies, are not identical. Thus, surgical slots should be matched to a relevant theatre type. In addition, the limited supply of equipment that is available for use during surgery needs to be considered. Further, the soft constraints of surgeon’s preferences combined with the hard constraints on their availability should also be taken into account.

The remaining sections of this paper are organised as follows. In Section 2, we consider the problem from the perspective of a partner hospital, and then proceed to identify the key factors to be included in our model.  This is followed by a detailed description of our mixed-integer programming model in Section 3, while details of the implementation of the model are given in Section 4.  Then computational results are provided in Section 5 using both data from the partner hospital and simulated data.  After discussion of these results in Section 6, some conclusions are drawn in Section 7 including recommendations for future work.

## 2. Background

### *2.1. Hospital perspective*

Our study was undertaken with the help of a collaborating hospital (confidentiality restrictions prevent us from naming the hospital). The hospital, with 11 theatres of 6 different types, and approximately 450 beds, operates a cyclic timetable over two weeks with morning and afternoon slots in all theatres. For some theatres, the afternoon slot is extended into an afternoon/evening slot.

Interviews with staff at the hospital were conducted as part of our study. These interviews have greatly assisted the authors in understanding the constraints on the master surgical timetable. The interviews reveal that operating theatre time, surgeons’ (and other staff) availability, and the equipment required are all significant factors to consider. Notably the interviews have raised the issue of the operating theatre type required for different operations, which is not mentioned by Cardoen et al. (2010). Indeed, the literature generally considers the theatres to be identical, and this is mentioned explicitly by Fei et al. (2008), Hans et al. (2008), Lamiri et al. (2008) and Oostrum et al. (2008). Exceptions to this appear of the work of Santibanez et al. (2007), Testi et al. (2007) and Zhang et al. (2008). The former requires “compatibility between the operating room and the specialty”, while Testi et al. (2007) and Zhang (2008) consider theatres with particular characteristics, but do not allow any theatre to belong to more than one type, which can be the case in reality.

Surveys of operating theatres in Great Britain and Ireland by Humphreys et al. (1995) and Smyth et al. (2005) indicate that the majority of hospitals have different theatre types for different types of surgery. This implies that any model that is to be useful to hospitals in carrying out their theatre scheduling must take account of theatre types.

The list of constraints from Cardoen et al. (2010) given above includes wards and ICU, the latter being a specialised type of ward. Discussions with hospital staff highlighted the constraints on the availability of beds, particularly in ICUs, as important in reducing the number of cancellations, and therefore in enabling hospitals to meet targets set by the government.

Discussions with surgeons and other surgical staff have revealed that they find it significantly reduces pressure on them if the same surgeon has an all-day slot, i.e. the same surgeon is assigned the morning and afternoon slots for a particular day. On the other hand, having the same surgeon assigned a morning slot in one theatre followed by an afternoon slot in another theatre is undesirable, as an over-run for the morning slot results in the afternoon surgery for two theatres starting late, thus propagating the disruption. It is also preferable to have the timetable repeating weekly if possible, which is one of the objectives suggested by Blake et al. (2002) and Belien et al. (2009) for tactical scheduling.

The hospital recently redesigned their master theatre timetable to take into account changes in staffing. This was a substantial project for the hospital managers, which took place over a four-month period, and culminated in a small number of managers spending a weekend rearranging a paper timetable to achieve the final version. At the start of this project, the intention was to ask surgeons for their preferences and incorporate these into the final schedule, but this was not possible due to the complexity of finding a feasible timetable by hand. However, it does raise the desirability of taking account the preferences of the key personnel when developing timetables. None of the literature on operating theatre scheduling reviewed by Meskens et al. (2013) considers surgeon preferences with regard to operating room allocations. Of the studies we have identified in the literature that consider preferences, four of them consider preferences while generating master surgical timetables. Specifically, Ozkarahan (2000) and Belien et al. (2009) consider preferred operating rooms, Blake and Carter (2002) attempt to ensure that “physicians are able to generate a preferred level of income”, and Testi et al. (2007) define surgeon preference based on length of stay, which is more a preference relating to wards than to surgeons. Thus, some surgeons’ preferences are considered in the literature, but they are not asked to provide a full list of their preferences by theatre, day of the week and time of day.

The difficulties experienced by the hospital with which we have collaborated in obtaining a feasible timetable highlights the extent of the constraints on the timetable. This difficulty is largely due to the desirability of using theatres as much as possible and constraints on the availability of surgeons.

The staff at the collaborating hospital were also unable to take account of the potential effects on bed usage of the new timetable because computerised support was not used. This highlights the need for a model that can take into account the relevant factors to assist hospital surgical divisions in devising new timetables.

### 

### *2.2. Identifying the important factors*

Based on the interviews with hospital staff and the areas mentioned by Cardoen et al. (2010) and other literature, in order to develop useful master surgical schedules the following factors should be taken into account:

* the amount of theatre time available;
* the amount of theatre time to be assigned for each specialty or surgeon;
* the types of the theatres, as some procedures will require equipment that is not available in all theatres;
* the availability of beds in wards;
* the availability of surgeons and if possible their preferences;
* the availability of other resources, such as other staff and equipment;
* the desirability of all-day slots and repeating slots weekly;
* the desirability that the same surgeon does not have a morning slot in one theatre followed by an afternoon slot in a different theatre on the same day;
* the timetable should be robust against uncertainty.

The first three of these relate to the theatre time, the next three to resources, while the remainder relate to desirable features of the timetable. Other factors considered in the theatre scheduling literature include patient waiting times, the number of operations not taking place for clinical reasons, and precedence constraints, release dates and due dates for operations. These are all aspects of theatre scheduling that relate to individual patients, and are therefore not relevant at the tactical level of scheduling where individual patients are not considered.

Holding areas for patients before surgery starts and post anaesthesia care are also not included in our model because they do not affect theatre scheduling according to our discussions with hospital staff. While space in ICUs is not explicitly included in this list, an ICU is a type of ward and can be included as such if wards are considered.

Staff other than surgeons can be considered by regarding them as items of equipment with limited availability. The scheduling of such staff is not modelled explicitly as within the collaborating hospital this is done on a weekly basis independent from the creation of the master theatre timetable.

Some of the factors considered will form hard constraints on the model, while others are objectives that should ideally be optimised in the timetables produced. As discussed in Section 1, there are other studies that have included various combinations of these factors in the model, but we are unable to find a study incorporating all of them.

## 

## 3. Integer programming model

As discussed above, our problem has multiple objectives: it is desirable to smooth bed usage, give surgeons slots they prefer, have all-day slots and repeat assignments weekly. This can be addressed using multiple criteria mixed-integer programming (Ehrgott, 2005), where the objective function is a sum of weighted objectives (Gass, 1986, Ehrgott and Gandibleux, 3003, Jones, 2011). The ability to weight the importance of the various objectives is particularly desirable in this case as different hospitals may have different priorities and the objective function can therefore be adapted accordingly. This section sets out how such a formulation can address the factors identified in Section 2 to produce a cyclic timetable, where the cycle length in weeks can be selected by the user.

### *3.1. Parameters*

The following notation defines the parameters used in our model. We start with the definitions of sets.

*I* = the set of surgeon

*D* = {1,…,*C*} is the set of days in a cycle, where *C* is the cycle length

*T* = the set of individual theatres

*S* = {1,…,*S*′}is the set of individual theatre slots, where *S*′ is the total number of slots

S- = {1,…,*S*′1} is the set of individual theatre slots excluding the last

*H* = the set of theatre types

*E* = the set of individual equipment types

*J* = {0,1,…,*J*max} is the set of days for which a patient may require a bed, where *J*max is set to be sufficiently large that the number of patients with longer stays is negligible and can be ignored.

*K* = the set of individual wards

In order to simplify the notation for all parameters and variables having an index for slot *s*, a theatre *t* and a day *d* will also be indexed, and therefore it is not necessary to index slots with the relevant day and theatre. Whenever a theatre slot is non-existent, then no surgeon should be set as available for that slot, thereby ensuring that sessions will not be scheduled inappropriately.

Whenever a summation over one of these indices is used, it should be assumed, unless specified otherwise, that it is over all values within the ranges given above.

Other parameters are as follows.

** = the desired frequency of repeating within the cycle: it is set to 0 if no repeats are considered desirable; otherwise, it must be a divisor of the cycle length.

For example, the set of types that can be used for day-case surgery includes most of the theatre types, but the set of types that can be used for major orthopaedic surgery is limited to a particular type of main theatre with special air filtration systems.

 the number of slots of theatre type *h* required by surgeon *i* over the whole cycle (these values are determined by strategic theatre scheduling decisions). 

** = a non-negative value below which a preference score is considered low.

 a non-negative preference score specified by surgeon *i*, for slot *s* on day *d* in theatre *t*



 the number of units of equipment of type *e* available. 

 the proportion of patients in ward *k* still occupying beds *j* days after an operation by surgeon *i* in theatre *t* (allowing for a cyclic theatre schedule, see Section 3.4.4). 

 the number of beds available on day *d* in ward *k.* 

 the maximum number slots to which surgeon *i* can be assigned in a day. 

Also, *α1,…, αK, β*1, *β*2, *β*3, *β*4 and *β*5 are non-negative weighting parameters for use in the objective function.

### *3.2. Variables*

The binary variables that we use can be defined as follows.



Our other variables are as follows.

 the expected number of beds require2d in ward *k* on day *d.* 

 the minimum difference for ward *k* between the expected number of beds required and the beds available over all of the days of the cycle *k.* 

### *3.3. Objective Function*

Maximise



The weighting parameters *α1,…, αK*, *β*1, *β*2, *β*3, *β*4 and *β*5 can be adjusted reflect the values appropriate to the hospital and/or for the user to explore how to balance the various objectives (it is suggested that initially equal weights are used and then adjusted depending on the changes desired to the resulting timetable). This overall objective function is to be optimised subject to constraints (1) to (18) as specified below.

is the sum of the weighted minimum difference between the number of beds available and the number required for each ward *k*, so by maximising this term we make the smallest difference between beds used and beds available as large as possible. Therefore, this is a measure of how well the bed usage is smoothed in the timetable. Individual weights for the wards allow ward size and the importance of particular wards (such as ICUs) to be taken into account.

is the sum of the preference scores for the slots that have been assigned, which is a measure of how well the preferences of surgeons are met by the timetable.

is the total number of times surgeons are assigned to slots for which they have low preference scores. This allows an additional penalty for assigning a slot to a surgeon who views it as very undesirable.

is the number of times that the same surgeon is assigned to consecutive slots in the same theatre on the same day.

 is the number of times that the same surgeon is assigned to consecutive slots in different theatres on the same day.

 is the number of times that any surgeon is assigned to repeat the same slot in the same theatre at the specified interval ** for such repeats (usually weekly).

### *3.4. Constraints*

#### 3.4.1. Constraints linking other variables to x

The constraints on the values of the binary variables *y, u, v* and *w* in terms of *x* are as follows.

If the preference score for a slot that is assigned to a surgeon is low then assign a value of one to the appropriate *Y* variable:

  (1)

For the last slot in each day no surgeon can be in the same theatre for that slot and the next slot so assign a value of zero to the appropriate *u* variables:

  (2)

For all other slots if the same surgeon is in the same theatre for two consecutive slots assign allow the objective to push the appropriate *U* variables up to a value of one:

  (3)

For the last slot in each day no surgeon can be in the same theatre for that slot and a different one for the next slot so assign a value of zero to the appropriate *v* variables:

  (4)

For all other slots, if the same surgeon changes theatres between two consecutive slots, ensure that the appropriate *v* variables have a value of one:

  (5)

Where the same surgeon has the same theatre slot in consecutive weeks allow the objective to push the appropriate *W* variables up to a value of one:

  (6)

  (7)

The inclusion of both constraints (6) and (7) ensures that each repeat is counted once for each time it repeats within the cycle. For a 2-weekly cycle with weekly repeats desirable, these constraints count each repeat twice; for a 3-weekly cycle with weekly repeats, it counts each repeat three times, if it occurs in all 3 weeks. In cycles of 3 or more weeks, these constraints allow counting of repeats that occur in some pairs of weeks, but not all pairs of weeks of the cycle.

#### 3.4.2. Other resource and availability constraints

Some straightforward resource and availability constraints are given below.

Slots can only be used if they are available:

  (8)

Each surgeon is in at most one theatre at any time:

  (9)

Surgeons’ availability constraint:

  (10)

Limit on number of slots each surgeon can do per day:

  (11)

Equipment constraint:

  (12)

#### 3.4.3. Demand constraints

Since the theatres could theoretically belong to any mix of sets of types and the demand is by types of theatre, it is not straightforward to ensure that each surgeon’s theatres are meeting their demand for different sets of types of theatre, without adding additional variables to a problem already having a large number of variables. Additionally, if surgeons have two slots in a theatre which can deal with cases of the types they would do in two theatre types, then they have more flexibility if they can schedule either type of case into either slot, so specifying the type for each slot would be unrealistic. Hence, introducing additional variables would not be a good match for what is likely to occur in practice. Consequently, there is a need for the following three constraints.

The demand for each set of types of theatre for each surgeon *i* is met:

  (13)

The overall demand for each surgeon *i* is met exactly:

  (14)

Each surgeon *i* does not use any theatre on more occasions than this surgeon’s total demand for theatres of this type:

  (15)

As the demand for each set of types of theatre for surgeon *i* must be met or exceeded, we require constraint (13).

Suppose that we just count the number of theatres a surgeon has of each set of types against their demand for that set of types (just constraint (13)). This could allow surgeons to be assigned more slots than needed, which would not fit with the objective of using theatre time as efficiently as possible. Thus, for each surgeon *i*, an additional constraint is needed to specify that the number of theatre slots assigned to surgeon *i* is equal to the total demand for theatre slots for that surgeon; hence constraint (14).

We now present an example that demonstrates the need for constraint (15). Suppose we have sets of theatre types A, B and C, with theatre 1 in set A and theatre 2 in sets B and C. If a surgeon requires two slots in set of types B and one in set of types C, then assigning them one slot in theatre 1 and two slots in theatre 2 will satisfy constraints (13) and (14), as the two slots in theatre 2 mean there are two potential slots of set of types B or C available (meeting constraint (13)) and the total number of slots assigned is 3 (meeting constraint (14)), but it is impossible for one slot in theatre 1 and two slots in theatre 2 can be assigned to be two slots of set of type B and one of set of type C because theatre 1 can only be of type A. Hence there is a need for constraint (15) that forbids a surgeon to be assigned more slots in a theatre than the surgeon’s total demand for sets of types. Thus, in this example, theatre 1 could not be assigned and this difficulty is resolved.

The above argument demonstrates the need for the constraints given, but it does not exclude the possibility that there are other possible situations where the demand is not met as intended, while these constraints hold, i.e. another constraint may be required. A detailed proof that these constraints are sufficient is given in the Appendix.

Along with the consideration of types of theatres, this set of constraints addressing the possible combinations of ways of meeting the requirements for theatres of different types provides a new addition to the literature on theatre timetabling.

#### 3.4.4. Constraints relating to use of beds

The constraints relating to the availability of beds are:

  (16)

  (17)

where *N* = {(*m* - *d*)/*C*, …, (*J*max + *m* – *d*)/*C*}.

  (18)

This method of smoothing bed usage is based on that used by Gallivan and Utley (2005) in their work on booking patients into a treatment centre. In their study, individual procedures are scheduled into a cyclic timetable and the probability that a patient is still in a bed a number of days after their operation is used to calculate the expected number of patients in beds on each day of the cycle.

For the master timetable, we are scheduling slots to surgeons, rather than individual procedures. As the case mix of each surgeon and the number of cases they treat in a slot can vary, it is not possible to work at the level of individual patients. Thus, our variation of the bed usage model proposed by Gallivan and Utley (2005) includes changes to adapt it to the new context, including moving from the probability of a patient being in a bed, to the expected number of beds used on each day of the cycle resulting from a surgeon’s use of a theatre slot.

Using historical data, the expected number of patients in beds in ward *k* for each of the *j* days after each surgeon *i* has had a theatre slot in each theatre *t* is evaluated, for input into the model in the form of values. Constraint (17) uses this information since the right-hand side represents the sum of the expected number of patients from each assigned theatre slot to each day of the cycle. This is similar to the method of Gallivan and Utley’s (2005) for specifying the expected contribution of each timetabled procedure to the number of patients in beds on each day of the cycle.

Constraint (17) is constructed as follows;

* To calculate the number of patients in beds in ward *k* on day *d* of the cycle resulting from surgeon *i* having a slot in theatre *t* on day *m* of that cycle if *m* ≤ *d*, then we require *j = d* *m*, giving . If *m* > *d* then day *m* occurs after *d* in the cycle so its occurrence in the current cycle does not make any contribution to the number of patients in beds on day *d* of that cycle.
* As patients can stay in the hospital for longer than the cycle length, to obtain the total contribution to the number of patients in beds in ward *k* on day *d* of the cycle resulting from surgeon *i* having a slot in theatre *t* on day *m*, it is necessary to sum the contributions from all of the previous cycles.
* To calculate the number of patients in beds in ward *k* on day *d* of the cycle resulting from surgeon *i* having a slot in theatre *t* on day *m* of the previous cycle, we require *j = C + d*  *m* giving. This applies to any day *m* of the previous cycle.
* Similarly, for the contribution from day *d*, *n* cycles previously; we calculate the number of patients in beds in ward *k* on day *d* of the cycle resulting from surgeon *i* having a slot in theatre *t* on day *m* of that cycle, then *j = nC + d*  *m* giving .
* The maximum length of stay to be considered is *Jmax* so only values of *n* such that *nC + d*  *m* ≤ *Jmax* should be considered. Similarly, operations from future days do not contribute so *nC + d*  *m* ≥ 0. Rearranging these equations gives the set *N* of values for *n* as specified above.
* Therefore, the sumis the number of patients in beds in ward *k* on day *d* of the cycle resulting from surgeon *i* having a slot in theatre *t* on day *m* of the cycle.
* The summation in the right-hand side of (17) ensures that the contribution is only counted if the slot is assigned by setting  = 1. Summing over *i*, *s*, *t* and *m* ensures that the contribution from every scheduled theatre slot is included.
* Therefore, constraint (17) sets the value of  to be the expected number of beds required in ward *k* on day *d*.

Constraint (18) then sets the value of  to be the minimum difference between the number of beds available on day *d* in ward *k* and.

*3.5 Model Overview*

### The *xi*,*t*,*d*,*s* variables, together with constraints (8) – (15), define the underlying structure of the master theatre timetable. Since indices *d* and *s* jointly correspond to a time period, we can regard our model as a generalisation of the 3-index assignment problem (see Burkard et al. (2009) for a review of assignment problems). Special cases of the 3-index assignment problem are NP-hard such as the axial version (Karp, 1972) and the planar version (Frieze, 1983).

### Our model, as presented above, is a linearized version of a quadratic model. For example, it is clear from the definitions of the *xi,d,t,s* and *ui*,*d*,*t*,*s* variables that *ui*,*d*,*t*,*s* = *xi*,*d*,*t*,*sxi*,*d*,*t*,*s*+1, which implies that variables *ui*,*d*,*t*,*s* can be removed at the expense of introducing a quadratic term into the objective function. Thus, the model also has some of characteristics of a quadratic assignment problem.

### The problem is also similar in nature to types of educational timetabling problems. For example, a surgeon is analogous to a class to be scheduled, which has to be assigned to rooms and to time slots. This similarity can be observed from an integer programming model for a university timetabling problem that is proposed by Daskalaki et al. (2004).

### *3.6. Cutting Planes*

In order to gain further insight into the running of the model, the values of variables in the solution to the LP relaxation of the MIP can be considered. It is reasonable to expect that adding constraints which reduce the feasible region for the LP relaxation, but not the feasible region of the original problem, would speed up the solution process as the solution to the LP relaxation should then be closer to the solution of the MIP problem.

The variables *ui,t,d,s* and *wi,t,d,s* have positive coefficients in the objective function, so the optimisation process aims to make their values as big as possible. The upper values of these variables are limited by constraints (3), (6) and (7), which exploit the binary nature of *ui,t,d,s* and *wi,t,d,s* with inequalities that relate them to *xi,t,d,s*. In the linear relaxation of test instances of problem, non-binary values of *ui,t,d,s* and *wi,t,d,s* are generated, so it is worth finding cutting planes whose addition provides a tighter formulation than with only (3), (6) and (7) constraining *ui,t,d,s* and *wi,t,d,s*.

If *xi,t,d,s* = 0, then both *ui,t,d,s* and *wi,t,d,s* should also be zero, as if a surgeon is not operating in a slot in a particular theatre then this surgeon cannot be operating in that slot in that theatre and the one after it in the same theatre or in the same slot and theatre with the desired repeat frequency in the timetable. Thus, the following constraints can be added to the problem.

Tighter constraints from (3):

  (3a)

  (3b)

Tighter constraints from (6):

  (6a)

  (6b)

Tighter constraints from (7):

  (7a)

  (7b)

Our findings to date indicate that the addition of these constraints increases the speed at which the solver optimises the MIP, with the solution gap (between the best bound and best solution) after 100 seconds going from 9.44% without the extra constraints, down to 4.04% with them. A similar improvement is observed after 200 seconds, with the solution gap decreasing from 6.27% to 3.99% when adding the constraints. These results suggest that these cutting plans are useful in reducing computation time and should be included.

Note that the original constraints are still required to ensure that the *u* and *w* variables are assigned values in accordance with their definitions. These additional constraints are also included in the model as implemented below.

In summary, we have defined the problem and formulated it as a multiple criteria MIP: the objective function provided in Section 3.3 is to be optimised, subject to constraints (1) to (18), including (3a), (3b), (6a), (6b), (7a) and (7b). The next stage is to implement the model and use both real and randomly generated data to evaluate its performance.

## 

## 4. Implementation

This section explains how the formulation developed in the previous section is implemented. Firstly, the data requirements are discussed. This is followed by consideration of the software to be used and how the data is entered into the software. The validation and verification of the model is then discussed. Lastly, the reasons for not considering any stochastic elements of the problem are explained.

### *4.1. Data*

The hospital staff working on the timetable would know the values to enter for the majority of the data that are required. This information for our example case is obtained from the interviews with staff. This includes the cycle length required, the repeat frequency that is desirable (weekly), the number of theatres, the types of those theatres, the maximum number of theatre slots available on any day in the schedule, the availability of the theatres, the number of surgeons, the availability of equipment/resources that needs to be considered at this level of scheduling, the maximum number of days that patients spend in beds after surgery (ignoring outliers), the number of wards, and the maximum number of beds available in each ward.

Other information such as the surgeons’ availability and preferences requires consultation with individual surgeons. However, in our tests, a mixture of random sample data and data devised to test specific aspects of the model is used for these values.

We use the existing hospital timetable to give the number of slots in each type of theatre required by each of the surgeons. This information would usually come from the decisions made within the hospital in their strategic planning.

The values of , the expected number of patients in ward *k* still occupying beds *j* days after being operated on by surgeon *i* in theatre *t*, are less straightforward to assign, but are obtained by collating the hospital records of previous cases. They are calculated based on the historical number of patients in the beds of each ward on the days following operations by each surgeon in theatre *t*. If no such data exist, then the data from a theatre of the same or of a similar type as *t* are used.

### *4.2. User interface spreadsheet*

Of the software available for solving MIP problems, FICOTM Xpress Optimization Suite is used for this study. This software is selected because other applications in the literature show that the Xpress optimiser is powerful and user friendly.

Hospital staff members are not expected to be familiar with optimisation software such as Xpress. Also, entering the data directly into the arrays of Xpress would be difficult to do accurately, since there is such a large quantity of data to load. However, hospital staff are typically familiar with Excel, and setting out the data entry for the arrays in a spreadsheet makes it much easier to avoid errors with data input. Therefore, Excel is used for the data entry in our model.

In addition to facilitating the input of data, the spreadsheet also contains sheets for displaying the results of the optimisation, including the suggested timetable, and of analysis of that timetable including a graphical representation of the number of beds required compared with the number available over the course of the cycle of the timetable. Figure 1 illustrates an example of the worksheet that gives the analysis of the suggested timetable.

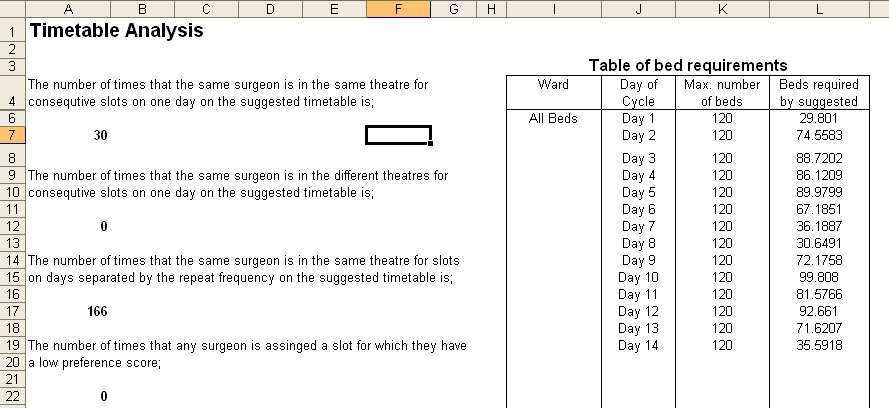


Figure 1: Example of the Timetable Analysis worksheet

Visual basic for applications (VBA) coding is used to load all of the data from the spreadsheets, format it and save it as a text document ready for use as input into Xpress. Similar code is also used to read data from an output file produced by Xpress, to import the results back into Excel and to display them to the user in the output worksheets described above. To reduce the scope for error in entering the data, Excel’s validation functions have been used on the majority of the data entry cells, to specify the type of data that can be entered.

### *4.3. Feasibility testing*

Given the complexity of the problem and that hospitals are expected to be using their theatres at close to capacity (and hence close to infeasibility), it is possible that the user may specify a problem that is infeasible.

The problem of diagnosing the cause of infeasibility in MIP and IP has been studied by a number of researchers; Chinneck and Greenberg have a number of papers on the subject (Chinneck, 1997 and 2001; and Greenberg and Murphy, 1991). The methods they describe for diagnosing infeasibility or subsets of constraints that are infeasible all require considerable knowledge of MIP/IP and the methods used to solve such problems. As the intention is that our model will be used by hospital staff with at best limited knowledge of these areas, it would be unreasonable to expect them to be able to interpret the results of the type of analysis conducted in these studies. Therefore, a more basic approach to resolving infeasibility is required.

Basic feasibility testing has been built into the VBA code that uploads the data from the spreadsheet and formats it for the solver. This compares the overall demand with the availability of surgeons and theatre slots and raises an appropriate error message to the user for errors such as: “The number of sessions that [surgeon] is available to operate in is less than the number of slots they are to be assigned overall”. Similar messages are generated if any surgeon requires more slots of any type than they can be available for, if the total demand exceeds the number of slots available, or if the demand for any type of slots exceeds the availability of that type. These types of quick checks are used to identify the more obvious ways in which a problem could be infeasible, thus avoiding the user wasting time trying to run the solver on a problem that has a data entry error or is infeasible for straightforward reasons.

There are numerous more complex ways in which the problem could be infeasible. Thus, to enable users to find feasible solutions, we recommend that they firstly ensure that their current timetable would be feasible for the data that they enter (where any additional surgery to be added can be scheduled in currently empty slots). This ensures that a feasible solution will be found. Analysis of the resulting timetable, and gradual insertion of the additional constraints that are required, will enable the user to identify compromises to their original problem that will yield a feasible timetable.

### *4.4. Validation of the model*

For the master surgical timetable, the partner hospital has an existing timetable, so the conditions can be set so that only the existing timetable is feasible and the values of the objectives can be checked against the real values. As expected, this produces the current timetable, with surgeons assigned their current slots, and the objective function value as expected. Thus, the model is validated against the real-life system.

### *4.5. Stochastic considerations*

In Section 1, the stochastic elements of the problem are raised as an important factor in theatre scheduling, but this aspect is not addressed in the formulation given in Section 3.

For the master surgical timetable, the most significant area of variability in the results concerns the number of beds required. This arises from both variability in the length of time patients stay in beds after their operations, and in the number of patients each surgeon treats in a theatre slot, which in turn arises from the stochastic nature of demand. Combining the effects of these two factors over all theatre slots would add significantly to the complexity of the model and thus considerably increase computational time, thereby making the use of the model less attractive to hospital managers.

Furthermore, as no consideration of the effects of the surgical timetable on bed usage is currently made by the hospital in producing the timetable, accounting for the average usage will provide a considerable improvement on the current situation. Fugener et al. (2014) have made significant steps towards a full consideration of the stochastic elements in a downstream bed usage model in master theatre timetable development, although they use an ‘approximated objective function’. Although they include significantly fewer aspects of theatre scheduling in their model than is the case in ours, their work suggests that the prospects for exploring the stochastic nature of this type of problem in the future are positive.

The most significant argument for not considering the stochastic nature of the problem at this level is that, once a master surgical timetable is selected and implemented, the variations in length of stay can be considered during day-to-day scheduling, when combinations of individual patients to book into slots are considered. In the latter problem, the expected number of patients for each slot will be known and the stochastic elements of the problem can then be considered much more effectively. Astaraky and Patrick (2015) demonstrate how this can be achieved when scheduling specific cases.

## 5. Computational results

In this section, we first compare the results obtained from the model with those of the current timetable for the collaborating hospital. We then investigate how random variations in the data affect the computational results,

### *5.1. Collaborating hospital*

The test case for the collaborating hospital has 11 operating theatres, which can be split into 6 sets of types, 54 surgeons, with up to 3 slots per day (the evening slot is only available on certain days), 9 equipment types to consider, and a two-weekly repeated cycle. The types of theatre is an important consideration because less complex surgery can be conducted in an operating theatre that is better equipped than necessary, but not vice versa, and there is more than one type of fixed equipment that may be required. Table 1 illustrates how the mix of surgery types and theatre types is configured in the data used. The differences between theatre types relate to theatre size and fixed facilities available within the operating theatres, as well as hospital policies. Surgery types 1 to 6 are generally carried out in theatres A to F respectively, but as the table shows there is variability in the flexibility to conduct the surgery in other theatre types.

Table 1: Data showing which types of surgery can be performed in the various theatre types (“x” indicates that a surgery type can take place in a theatre type).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Surgery Type** | **Theatre Types** | | | | | |
| **A** | **B** | **C** | **D** | **E** | **F** |
| **1** | x | x | x | x |  |  |
| **2** |  | x |  | x |  |  |
| **3** |  |  | x | x |  |  |
| **4** |  |  |  | x |  |  |
| **5** |  |  |  |  | x |  |
| **6** | x | x | x | x | x | x |

The equipment types considered are those with limited availability that can move between theatres, but not within a single theatre slot. The required number of theatre slots of each type for each surgeon is taken from the existing timetable, so the results will show the extent to which the model can improve on the current system. The information on bed requirements by ward is not available, so we treat all of the beds available as belonging to a single ward and use a maximum length of stay of 30 days. The bed usage resulting from a particular surgeon having a slot in a particular type of theatre is derived from the previous years’ data.

The resulting problem has 124755 variables, of which 124740 are binary variables, and 138366 constraints.

As mentioned in Section 4.4, the model has been run with the surgeons’ availability set to ensure that the current timetable is produced, which means that the data on the current timetable are collected in the same way as the data on the proposed timetable, thus allowing them to be easily compared.

The proposed timetable discussed in this section is based on the assumption that all of the surgeons are always available. In reality, this is not the case, but as full information on each surgeon’s availability and preferences is not available (due to the time and effort in obtaining such data, as well as the raising of hopes that would result from collecting these data in a situation where the hospital has no intention of changing the timetable imminently) any other model would require further assumptions and potentially take us further from reality. It is in any case of interest to explore how much the usage of beds could be smoothed if all of the surgeons were available for all slots. The effects of changing the availability of surgeons are explored in Section 5.2.

It is hoped that in the future when the partner hospital wishes to change the theatre timetable, it will be worthwhile collecting data on each surgeon’s availability and preferences, and include this information in the model which would assist in the development of a new timetable.

Table 2 give a comparison of the original and proposed timetable as suggested by the MIP when solved with Xpress-IVE, while Figure 2 explores the benefits of the proposed model in terms of the smoothing of bed usage. These results demonstrate that considerable improvements in the smoothing of bed usage, numbers of all-day slots and numbers of slots repeated weekly can be achieved. The actual achievement of such improvements depends on the extent to which the surgeons’ availability is limited. The bed usage data in the graph show the expected averages for each day of the 14-day scheduling cycle, so increasing the gap between these and the number of beds available increases the flexibility for dealing with variations around the averages.

**Table 2: Comparison of objectives for current and proposed timetables**

|  |  |  |
| --- | --- | --- |
| **Indicator** | **Original Timetable** | **Proposed Timetable** |
| Maximum beds used | 100 | 83 |
| Number of surgeons changing theatres | 0 | 0 |
| Number of all-day slots | 30 | 55 |
| Number of slots repeat weekly | 166 | 178 |

Figure 2: Graph comparing bed smoothing by the model and the existing timetable.

These results demonstrate that considerable improvements in the smoothing of bed usage, numbers of all-day slots and numbers of slots repeated weekly can be achieved. The extent to which such improvements can actually be achieved depends on the extent of the limitations on surgeons’ availability. The bed usage shown in the graph are the expected averages for each day of the 14-day scheduling cycle, so increasing the gap between these and the number of beds available increases the flexibility for dealing with variations around the averages.

### *5.2. Data variations*

The results provided above do not include constraints on surgeons’ availability. Not only is this far from the case in real life, but also it takes so long to run the model when availability constraints are included that the computer used terminates the run due to insufficient memory before reaching a solution. Therefore, in order to explore the effects on solution time and the quality of solution (based on the gap between the best upper bound and the best solution found after 10 minutes) that can be obtained, we report on the results of tests with random data.

Our random test instances are based on data from the collaborating hospital. As above, we consider a hospital with 11 theatres that partition into 6 sets of types, with operations performed in up to 3 slots per day by 54 surgeons, using a total of 9 equipment types and working on a two-week cycle. In order to ensure that the problems are feasible, a random assignment of surgeons to slots is generated and the required numbers of slots of each type is set to fit with that random assignment. Then the availability of surgeons is set to ensure that they are available for each of their slots assigned on that timetable and for a percentage of the other slots. Each of these instances produces a model having in excess of 100,000 variables and 100,000 constraints. To explore the effects of different limitations on surgeons’ availability, each instance type was run five times with the percentage of the slots (in addition to that in the random timetable) that surgeons are available set to 100%, 90%, 75%, 50% and 25%, respectively.

Figures 3 and 4 are scatter plots showing how the solution gap and computation time vary with the availability of surgeons. Some of the values in these figures are too close to each other to be distinguishable: there are five points for each level of surgeon availability. The two figures demonstrate that as surgeons’ availability decreases the best available solutions are found faster. This is because for more limited availability of surgeons there are fewer feasible solutions to explore while searching for an optimal solution and demonstrating optimality. Note that Figure 4 contains fewer data points as it only includes completed runs for which optimal solutions are found, as there is no ‘number of seconds to reach optimal solution’ when an optimal solution cannot be found.

In reality, the availability of surgeons is quite constrained as they have outpatient clinics and other tasks that have to be conducted at specific times. For example, some surgeons work at other hospitals at certain points in the week. Therefore, it is useful that better solutions are found faster with more limited surgeons’ availability as this means that solution times for real-life instances are likely to be reasonable.

**Figure 3: How the solution gap varies with availability of surgeons**

Figure 4: How solution time varies with availability of surgeons, when optimality is proven.

### *5.3. Sensitivity to objective function weightings*

This section discusses the purpose of sensitivity analysis before showing how this approach is applied to the mixed-integer linear programming formulation for the master surgical timetable.

In linear programming, sensitivity analysis involves evaluating the extent to which data for the problem instance can change before the optimal solution changes. This is particularly useful if there is uncertainty around the values of the data for the objective or constraints. For linear programming, standard procedures are available for sensitivity analysis (Winston, 1994). However, sensitivity analysis for problems with integer variables is more complex as linear programming sensitivity analysis only applies if the solution to the LP relaxation satisfies the integrality constraints of the IP. For our master surgical timetabling problem, changes to the weights in the objective function should preferably result in changes to the optimal solution so that users can quickly obtain solutions for different prioritizations of objectives and compare the effects on the resulting timetables.

Our sensitivity analysis is applied to the test instance involving data obtained from the collaborating hospital, as described in Section 5.1. Table 3 gives the results of running the model with instances that are identical except that the objective function weightings are adjusted slightly each time. As only one ward is considered in our tests, the objective function coefficients α1,…, α*k* reduce to a single value that is denoted by α. Changes to the value of β1 are not considered as preference scores (other than distinguishing between theatre types) have not been entered for this data set.

**Table 3: Exploring the effects of small changes in the objective function weightings.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Weightings used** | | | | |
| α | 2 | 3 | 4 | 5 | Difference between beds available and max beds used | All day slots | Change theatres | Weekly repeats |
| 5 | 5 | 5 | 5 | 10 | 9.69 | 4 | 0 | 162 |
| 6 | 5 | 5 | 5 | 10 | 22.09 | 5 | 0 | 159 |
| 5 | 6 | 5 | 5 | 10 | 9.62 | 4 | 0 | 162 |
| 5 | 5 | 6 | 5 | 10 | 9.43 | 4 | 0 | 158 |
| 5 | 5 | 5 | 6 | 10 | 9.69 | 4 | 0 | 162 |
| 5 | 5 | 5 | 5 | 5 | 11.02 | 4 | 0 | 156 |
| 5 | 20 | 5 | 5 | 10 | 10.75 | 14 | 0 | 155 |

Table 3 illustrates that only small changes to the weights are required to produce different timetables, so the solution values are sensitive to the weightings used. Note that changes to the value of 4 do not alter the results as none of the timetables include any surgeons operating in different theatres in consecutive slots on the same day.

The sensitivity to changes in the objective function weightings observed above means that, in practice, hospital managers can run the model with a variety of values and then compare the resulting timetables. This will allow active consideration of the relative importance of the different criteria and their impact on the theatre timetable for their particular environment.

## 

## 6. Discussion of flexibility and limitations

The proposed model allows consideration of variations in the availability of surgeons. This could be particularly useful in demonstrating how much difference any changes in a particular surgeon’s availability makes to possible timetables, which would facilitate discussion with the surgeon on changing availability.

The current model cannot directly suggest specific instances where changes to some surgeons’ availability would help to improve the timetable. However, comparing the timetable produced with no limitations on a surgeon’s availability with that where the surgeon’s availability constraints are included can highlight issues where negotiation with the surgeon is worthwhile. Specifically, if the two timetables are identical, then that surgeon’s constraints are not worth considering, whereas the creation of two different timetables suggests that changes to that surgeon’s availability are worth exploring.

As discussed in Section 4.4, it is possible that changes to the availability of one or more surgeons could result in the problem becoming infeasible; hence, the above recommendation to ensure that the current timetable with any additional sessions incorporated is feasible initially. Additional constraints on availability can then be incorporated gradually thereby allowing the user to identify any particular constraints that make the problem infeasible. This addresses limitations on the extent to which the model can identify the constraints that make problems infeasible.

The ability to vary the weightings of the objectives (see Section 5.3) allows users to adapt the model to their priorities and to generate and compare different timetables based on variations around these priorities.

The data entry process is structured so that the numbers of surgeons, theatres, slots per day, wards and beds can be adapted to reflect the structure of any size of hospital. Other features such as theatre types and equipment are also fully adjustable. The use of an Excel interface ensures that the data entry process is accessible to hospital staff so that the full extent of the model’s flexibility can be exploited.

The ability to include limited availability of equipment in the model allows not only equipment, but also other staff limitations to be taken into account. It is possible that there are other restrictions on the timetable that another hospital would want to include that cannot be taken into account by the model. For example, we have not considered the possibility that some equipment can only be used once on any day. However, this level of detail is better addressed at the day-to-day scheduling level when it could be considered if a particular patient’s treatment would require particular equipment.

The most significant limitation of the model is that it does not take account of variations in the case mix of each surgeon from week-to-week and the variations in patients’ lengths of stay. Users should be particularly aware of this when considering the expected bed usage, as the values given are averages and the actual usage will vary. Even so, this inclusion of bed usage is a significant forward step from previous models that have not considered it when constructing master theatre timetables.

Any limitations on the accuracy of the data for predicting bed usage, particularly when new surgeons are introduced and the data would need to be estimated, should also be considered when interpreting the results.

## 7. Conclusions

This paper illustrates that it is possible to solve (in some cases approximately rather than optimally) the MIP for the problem of finding master surgical timetables in a reasonable amount of time using a standard MIP solver, for medium sized hospitals. Therefore, no further work is planned on applying column generation or heuristic approaches.

The stochastic nature of some aspects of the problem has not been incorporated at this level as it is felt that this can be addressed more effectively in the day-to-day scheduling process. All of the other factors identified in Section 2.2 have been included in the formulation, which as far as we are aware has not been done before. The specific elements of this model that are new contributions compared with other studies appearing in the literature are:

* The consideration of the availability of beds, particularly its inclusion at ward level and in tactical rather than day-to-day scheduling.
* Relating the expected contribution to bed usage of surgeons having particular slots is new, as is allowing the number of beds available to vary over the course of the repeat cycle for the timetable.
* In the literature different theatre types are rarely considered, so our inclusion of not only different theatre types, but also the way in which sets of theatres can meet different demand is novel.
* Consideration of the availability of surgeons is limited in the literature and the ability to consider their preferences is new.
* The ability to allow for the availability of other resources, such as staff and equipment, by using the equipment availability function is new.
* The inclusion of weekly (or other) repeats in a cycle of longer than one week is unusual, as is allowing for the need to avoid the same surgeon having consecutive slots in different theatres.

It has not proven possible to implement this model within a local hospital within the timescale of this project, due to the pressures on hospital staff. However, it is hoped that such implementation will take place in due course. In addition to the hospital staff time required to implement such models, the cost of the software needed to run them is a potentially significant limiting factor to their use in hospitals.

Future work in this area could incorporate more of the dynamic and stochastic elements of the problem, as well as bringing together solutions to the problems faced at the strategic, tactical and day to day scheduling levels. For such models to be implemented in the UK, it would require a significant change for hospital staff compared to current common practice.

## Acknowledgements

This research was undertaken with funding from Engineering and Physical Sciences Research Council (EPSRC) and written up while funded by the National Institute for Health Research Collaboration for Leadership in Applied Health Research and Care (NIHR CLAHRC) Wessex. The views expressed in this publication are those of the author(s) and not necessarily those of the EPSRC or NIHR.

The authors would like to extend their appreciation to the staff at the partner hospital for both giving up time for interviews and discussions of the research and providing the necessary data.

The optimisation software used was FICO® Express-IVE.

We would also like to thank anonymous reviewers whose comments which have greatly assisted us in improving the presentational aspects of this paper.

## References

Arenas, M., Bilbao, A., Caballero, R., Gomez, T., Rodriguez, M.V. and Ruiz, F., (2002). Analysis via goal programming of the minimum achievable stay in surgical waiting lists. *Journal of the Operational Research Society 53(4)*, 387-396

Agnetis, A., Coppi, A., Corsini, M., Dellino, G., Meloni, C. and Pranzo, M., (2014). A decomposition approach for the combined master surgical schedule and surgical case assignment problems, *Health Care Management Science* *17(1)*, 49–59.

Astaraky, D. and Patrick, D., (2015). A simulation based approximate dynamic programming approach to multi-class, multi-resource surgical scheduling. *European Journal of Operational Research 245(1)*, 309-319

Belien, J., Demeulemeester, E. and Cardoen, B., (2006). Visualizing the demand for various resources as a function of the master surgery schedule: a case study. *Journal of Medical Systems 30(5)*, 343-350

Belien, J. and Demeulemeester, E., (2007). Building cyclic master surgery schedules with levelled resulting bed occupancy. *European Journal of Operational Research 176(2)*, 1185-1204

Belien, J., Demeulemeester, E. and Cardoen, B., (2009). A decision support system for cyclic master surgery scheduling with multiple objectives. *Journal of Scheduling 12(2)*, 147-161

Blake, J.T., Dexter, F. and Donald, J., (2002). Operating room manager's use of integer programming for assigning block time to surgical groups: a case study. *Anesthesia and Analgesia 94(1)*, 143-148

Blake J.T. and Donald J., (2002). Mount Sinai Hospital uses integer programming to allocate operating room time. *Interfaces 32(2)*, 63-73

Blake, J.T. and Carter, M.W., (2002). A goal programming approach to strategic resource allocation in acute care hospitals. *European Journal of Operational Research 140(3)*, 541-561

Bowers, J. and Mould, G., (2004). Managing uncertainty in orthopaedic trauma theatres. *European Journal of Operational Research 154(3)*, 599-608

Burkard, R., Dell’Amico M. and Martello, S., (2009).  *Assignment Problems*, SIAM: Philadelphia

Cardoen, B., Demeulemeester, E. and Belien, J., (2010). Operating room planning and scheduling: a literature review. *European Journal of Operational Research 201(3)*, 921-932

Chinneck, J.W., (1997). Finding a useful subset of constraints for analysis in an infeasible linear program. *INFORMS Journal on Computing 9(2)*, 164-174

Chinneck, J.W., (2001). Fast heuristics for the maximum feasible subsystem problem. *INFORMS Journal on Computing 13(3)*, 210-223

Daskalaki, S., Birbas, T. and Housos, E., (2004). An integer programming formulation for a case study in university timetabling. *European Journal of Operational Research* 153 (2004) 117-135

Ehrgott, M. and Gandibleux, X., (2003). *Multiple Criteria Optimisation: State of the Art Annotated Bibliographic Surveys*, Kluwer Academic Publishers: New York

Ehrgott, M., (2005). *Multicriteria Optimization 2nd ed.,* Springer: Berlin

Fei, H., Chu, C., Meskens, N. and Artiba, A., (2008). Solving surgical cases assignment problem by a branch-and-price approach. *International Journal of Production Economics 112(1)*, 96-108

Frieze, A.M., (1983). Complexity of a 3-dimensional assignment problem. *European Journal of Operational Research 13*, 161-164

Fugener, A., Hans, E.W., Kolisch, R., Kortbeek, N. and Vanberkel P.T., (2014). Master surgery scheduling with consideration of multiple downstream units. *European Journal of Operational Research 239(1)*, 227-236

Gallivan, S. and Utley, M., (2005). Modelling admissions booking of elective in-patients into a treatment centre. *IMA Journal of Management Mathematics 16(3)*, 305-315

Gass, S.I., (1986). A process for determining priorities and weights for large scale linear goal programmes, *Journal of the Operational Research Society* *37(8)*, 779–785

Greenberg, H.J. and Murphy, F.H., (1991). Approaches to diagnosing infeasible linear programs. *ORSA Journal on Computing 3(3)*, 253-261

Hans, E.W., Nieberg, T. and van Oostrum, J.M., (2007). Optimization in surgery planning. *Medium Econometrische Toepassingen 15(1)*, 20-28

Hans, E.W., Wullink, G., van Houdenhoven, M. and Kazemier, G., (2008). Robust surgery loading. *European Journal of Operational Research 185(3)*, 1038-1050

Humphreys, H., Stacey, A.R. and Taylor E. W., (1995). Survey of operating theatres in Great Britain and Ireland. *Journal of Hospital Infection 30(4)*, 245-252

Jones, D., (2011), A practical weight sensitivity algorithm for goal and multiple objective programming, *European Journal of Operational Research* *213(1)*, 238–245

Karp, R.M., (1972). Reducibility among combinatorial problems. In: Miller, R.E. and Thatcher J.W. (Eds), (pp. 85-103) *Complexity of Computer Computations*, Plenum Press, New York

Lamiri, M., Xie, X., Dolgui, A. and Grimaud, F., (2008). A stochastic model for operating rooms planning with elective and emergency surgery demands. *European Journal of Operational Research 185(3)*, 1026-1037

Oostrum, J.M., Van Houdenhoven, M., Hurink, J.L., Hans, E.W., Wullink, G. and Kazemier, G., (2008). A master surgical scheduling approach for cyclic scheduling in operating room departments. *OR Spectrum 30(2)*, 355-374

Ozkarahan, I., (2000). Allocation of surgeries to operating rooms by goal programing. *Journal or Medical Systems 25(6)*, 339-378

Meskens, N., Duvivier, D. and Hanset, A., (2013) Multi-objective operating room scheduling considering desiderata of the surgical team. *Decision Support Systems, 55(2)*, 650–659

Santibanez, P., Begen, M. and Atkins, D., (2007). Surgical block scheduling in a system of hospitals: an application to resource and wait list management in a British Columbia health authority. *Health Care Management Science 10(3)*, 269-282

Spratt, B. and Kozan, E., (2016). Waiting list management through master surgical schedules: A case study, *Operations Research for Health Care 10(1),* 49-64

Smyth, E.T.M., Humphreys, H., Stacey, A., Taylor, E.W., Hoffman, P. and Bannister, G., (2005). Survey of operating theatre ventilation facilities for minimally invasive surgery in Great Britain and Northern Ireland: current practice and considerations for the future. *Journal of Hospital Infection 61(2)*, 112-122

Testi, A., Tanfani, E. and Torre, G., (2007). A three-phase approach for operating theatre schedules. *Health Care Management Science 10(2)*, 163-172

Winston, W.L., (1994). *Operations Research, Applications and Algorithms 3rd Ed*. Duxbury Press: California

Zhang, B., Murali, P., Dessouky, M.M. and Belson, D., (2008). A mixed integer programming approach for allocating operating room capacity. *Journal of the Operational Research Society 60(5)*, 663-673

## Appendix A

The following argument uses induction to show that these three constraints are sufficient, by showing that the constraints ensure that it would always be possible to assign a set of types to each theatre and thus the demand for each set of types is met. The equation numbers relate to those given in the formulation in Section 3.4.3 of the paper. The set of

Case 1: If all of the theatres are of all sets of types then   1, and effectively *H* = {1} as we only need to consider one set of types, so they can be ignored in the constraints, which become;

  (13)

  (14)

  (15)

In fact in this case constraints (13) and (15) are redundant as constraint (14) is a tighter version of either of them. In this case as all theatres are of all set of types we just need the number of theatres each surgeon requires to be able to assign the relevant types to each theatre and meet each surgeons demand. If the problem is feasible then the demand can be met and only constraint (14) is needed.

Case *n*-1: The demand has been met with the current mix of sets of types for each theatre.

Case *n*: The theatres remain of sets of types as in case *n*-1 except theatre *K’* which is of all sets of types as in case *n*-1 except *k* (when it was assigned set of types *k* for some part of the solution in case *n*-1), then;

Constraint (13) remains as in case *n-*1 except for theatre *K’* and set of types *k*, as now  0 when *h* = *k* and *t* = *K’* so one term is removed from the sum which is greater than or equal to the demand for type *k*. This ensures that where in the solution to case *n*-1 theatre *K’* was being assigned set of types *k* then in the solution to case *n* theatre *K’* cannot be treated as meeting some of the demand for set of types *k*. And the demand for the other types is met as in the case *n*-1.

Constraint (14) continues to say that the total demand for theatres for each surgeon remains equal to the total number of theatres they are assigned.

Constraint (15) becomes tighter as = 0 when *h* = *k* and *t* = *K’*, where it was 1 before, so for each surgeon the limit on then number of times theatre *K’* can be used reduced by their demand for set of types *k*, ensuring that the demand for set of types *k* is met by other theatres where a slot in theatre *K’* had been assigned as set of types *k* that demand must now be met by a different theatre.

So where in a solution to case *n*-1 the demand for type *k* was met by theatre *K’* then in case n that demand cannot be met by type *k* so the number of times theatre *K’* can be used is reduced by constraint (15) and constraint (13) ensures that there must be available capacity in other theatres of type *k* to continue to meet the demand. In other words where a slot theatre *K’* was being assigned type *k*, it must no longer be assigned as type *k* and a different theatre must be assigned instead.

So by induction we can move from case 1 to any combination of the possible sets of types of theatre and the demand constraints are sufficient to ensure that demand is met (N.B. the problem may be infeasible for some combinations of sets of theatre types, so going from case *n*-1 to case *n* may make the problem infeasible).

Along with the consideration of types of theatres this set of constraints to address the possible combinations of ways of meeting the requirements for theatres of different types is a new addition to the literature on tactical theatre scheduling.