International Journal of Operations Research

International Journal of Operations Research Vol. 4, No. 4, 189−198 (2007)

Mathematical Programming Models for Solving Two Conference Planning Problems

E. Kozan* **and R. L. Burdett**

School of Mathematical Sciences, Queensland University of Technology, GPOBox 2434, Brisbane, Qld 4001, Australia

Received January 2006; *Revised September 2006*; *Accepted March 2007*

Abstract—Planning academic conferences is often a difficult, tedious and time-consuming task. This is because there are many possible designs that can be considered and created. Additionally many of the associated tasks are currently performed manually in an inefficient manner when they could easily be automated. The planning process is particularly important because high quality sessions can lose their value if the program is not planned properly. Similarly in the reviewing process each paper should be reviewed properly in order to achieve a high quality conference proceeding. For this reason, mathematical models are developed for the paper reviewing process and the program of presentations construction process. They involve the assignment of reviewers to papers, and the assignment of topics and papers to session and rooms, and papers to session slots. These two models are implemented and tested on real data associated with the Asia Pacific Industrial Engineering and Management Systems (APIEMS) 2004 conference.

Keywords—Conference planning activities, Course timetabling, Assignment problems

1. INTRODUCTION

Academic conferences of international or national status are regularly held in many countries each year to which scientific or other papers are usually written, submitted and lastly presented. The size of each conference usually varies from tens of people to thousands of people. Conference organisation is always a difficult and time consuming process for members of the organising committee, and this is particularly so for the larger international refereed conferences. There are many decisions that need to be made and many technical constraints that must be satisfied. Methods for reducing the manual effort, speeding up the entire process, and improving the solution quality are highly desirable.

Two generic mathematical programming models, namely a review process model and a program of presentations construction model are therefore developed in this paper. As conferences do not always have the same requirements different components of these models can be easily altered and or ignored if they are not applicable. Both models result in assignment problems, the second of a more difficult nature. Vast amounts of literature exist on solving assignment problems. For a review of recent developments on the solution of these problems Burdett and Kozan (2004) may be consulted. The aim of the paper however, is not to specifically develop new solution techniques for solving mathematical programming problems of this type although this is a necessary element of the paper. Rather it is to demonstrate how two conference planning activities can (or should) be performed efficiently, to give an overview of important features and issues, and lastly to report the application of the models to a real conference. Additionally this paper demonstrates what parameters are required and how they are computed, and discusses how effectively the processes can be modelled and solved.

To our knowledge these processes have not been significantly addressed before using OR techniques. This is perhaps not surprising since most conference organisers are not OR practitioners and are not aware of Operations Research or its techniques. There are also so many rules and regulations, constraints and decisions, and data that only a manual approach would be considered.

The program of presentations construction process is similar to course and examination timetabling in schools and universities. It has been shown that finding a timetable in a minimum number of periods is NP-complete (Burke et al., 1995). It has also been shown that the problem of deciding how long the timetable period is in order to schedule all exams is also NP-complete. For recent research and for a review of past research Burke and Newall (1999), Burke and Newall (2004), Burke et al. (2006a,b) may be inspected.

The format of the paper is as follows. In section two and three mathematical models are developed for both problems respectively. A case study is then presented in section four, which demonstrates the application and implementation of these models. Lastly conclusions and further research directions are given in section five.

* Corresponding author's email: e.kozan@qut.edu.au

2. THE REVIEW PROCESS MODEL

For a refereed conference, papers are accepted for publication in the proceedings based upon the outcome of the reviewing process. In the reviewing process each paper is usually reviewed by two academics in the same field as the author. For medium to large conferences this process is quite demanding. Particularly working out who the eligible reviewers are, and which papers and topics they can review. To our knowledge this task is often performed manually and or in an inefficient manner. This section therefore provides an alternative automated approach.

2.1 Parameters

The index *p*, *a* and *t* are firstly introduced and are used throughout to signify papers, authors and topics. Papers, authors and topics are labelled by integers for simplicity. The data requirements for this problem are quite large, and because of the sparseness of the data, set descriptions/definitions have been used. Alternative binary parameters could be used but the storage of large numbers of zeros (i.e. in the order of tens of thousands) is inefficient and unwarranted. The input parameters are as follows:

P = Set of submitted papers (accepted papers in the second problem).
A = Set of authors.
T = Set of topics/keywords.
PT = Set of paper-topic relationships, $PT \subset \{(p,t) \mid p \in P, t \in T\}$.
AT = Set of author-topic relationships, $AT \subset \{(a,t) \mid a \in A, t \in T\}$.
PA = Set of direct paper-author relationships, $PA \subset \{(p,a) \mid p \in P, a \in A\}$.
α_p = Principal (or primary) author of paper *p*.
nr_p = Number of reviewers required for paper *p*, usually two.
st_a = Status of author *a*.

It should be noted that elements of the paper-topic, author-topic and paper-author sets are pairs and these pairs must be input in order to create the sets. These sets are also subsets of the complete set of possible pairs shown next to their definitions. In particular paper *p* is associated with topic *t* if $(p,t) \in PT$. Similarly an author *a* is associated with topic *t* if $(a,t) \in AT$. A paper *p* has co-author *a* if $(p,a) \in PA$. The status of an author is a number between 1 and 4. Each number corresponds respectively with the following categories: non-academic, pre-doctoral, post-doctoral, and special. Technical committee members and other important people are categorised as special.

From the input parameters the following sets and values may be computed.

P_a = The set of papers associated with an author *a*.
A_p = The set of authors associated with a particular paper *p*.
PA' = The set of indirect paper-author relationships.
E = The set of eligible reviewers.
PR_p = The set of potential reviewers (i.e. candidates) of paper *p*.
np_t = The number of papers associated with topic *t*.
na_t = The number of authors associated with topic *t*.

The set of authors associated with a particular paper *p* is $A_p = \{a \mid (p,a) \in PA\}$ and the set of papers associated with a particular author *a* is $P_a = \{p \mid (p,a) \in PA\}$. The number of papers and authors associated with topic *t* respectively is the number of elements in the following subsets:

$$np_t = \left|\{p \mid (p,t) \in PT\}\right|, \quad na_t = \left|\{a \mid (a,t) \in AT\}\right| \tag{1}$$

The indirect paper–author relationships signify whether an author *a* and a paper *p* are associated with a common topic. The set is computed in the following way:

$$\begin{aligned} PA' = \{&(p,a) \mid (p,a) \notin PA, \\ &(\exists t \in T \mid (p,t) \in PT, (a,t) \in AT)\} \end{aligned} \tag{2}$$

Author *a* must not be a co-author of paper *p* in order for (*p*, *a*) to be an element of this set.

2.2 Determining eligible and potential reviewers

A critical component of the review process is the identification of which authors are eligible to review and which papers they are permitted to review. An author is eligible to review if they are a principal author of at least one paper. The set of eligible reviewers is therefore as follows:

$$E = \left\{a \mid a \in A, \left|\{p \mid p \in P, \alpha_p = a\}\right| \geq 1\right\} \tag{3}$$

Non principal authors could be eligible however contact details are not usually known for these people and they are consequently omitted. Students (who are pre-doctoral) may also not be included as reviewers due to inexperience. The status parameter may be used to add a third condition to equation (3) above. An author *a* is eligible to review paper *p* if the following is true:

i) They are eligible to review, i.e. $a \in E$
ii) They are not a co-author of paper *p*, i.e. $a \notin PA$
iii) They are associated with a topic of paper *p*, i.e. $a \in PA'$
iv) The status of *a* is greater than or equal to the status of the principal author, i.e. $st_a \geq st_{\alpha_p}$

Therefore the set of potential reviewers for paper *p* is as follows:

$$PR_p = \left\{ a \mid a \in E, (p,a) \in PA', st_a \geq st_{\alpha_p} \right\} \tag{4}$$

Optionally a reviewer may not be allowed to have any bias (positive or negative) towards a paper, i.e. $a \notin Bias_p$. A bias may occur when two papers have a common co-author. This means that co-authors of one paper should not be reviewers of the other and vice versa due to positive bias. For example this occurs when two phd students have the same supervisor. Algebraically speaking, if $\exists a \mid (p,a) \in PA$ and $(p',a) \in PA$ (i.e. author a is a co-author of paper p and p') then $a' \in Bias_{p'}\ \forall a' \in A_p$ and $a' \in Bias_p\ \forall a' \in A_{p'}$. Lastly it should be noted that if the set of candidates is zero for any paper then the reviewer for that paper must be assigned differently.

2.3 Decision variables

The decision variables for the review process model are $X_{p,a}$ and N_a. The first is a binary decision variable signifying whether author *a* is assigned as a reviewer of paper *p* and the second is an integer variable signifying the number of papers reviewed by author *a*. Both variables are related in the following way, $N_a = \sum_{\forall p \in P} (X_{p,a})$. One or both variables may be used in the following model.

2.4 Mathematical Model

The proposed model is as follows:

$$\text{Minimise } z = \sum_{a \in E} \left| N_a - \mu \right| \text{ where } \mu = \sum_p (nr_p) \Big/ |E| \tag{5}$$

$$\text{Subject to: } \sum_{\forall a \in PR_p} (X_{p,a}) = nr_p \ \ \forall p \tag{6}$$

$$X_{p,a} = 0 \quad \forall p \in P, a \notin PR_p \tag{7}$$

$$N_a = 0 \quad \forall a \notin E \tag{8}$$

$$N_a \text{ int } \forall a \in E \ \text{ or } \ X_{p,a} \in [0,1] \quad \forall p,a \tag{9}$$

Constraint (6) ensures that the correct number of reviewers is assigned to each paper. Constraint (7) ensures that an author can not be assigned as a reviewer of a paper if they are not a candidate. Similarly constraint (8) ensures that no papers can be reviewed if the author is not eligible. Constraints (7) and (8) reduce the number of decision variables that must be solved for. The objective function is to balance (minimise) the workload of reviewers. The term μ is the average number of papers per reviewer. This criterion was selected as it appears to be most relevant in practice. For example most authors do not want to be hassled to review too many papers if possible. Since the objective function is non-linear, linearization may also be performed in the usual way at the expense of significantly more variables and constraints.

2.5 Solution of the model

The model is a non-linear binary or integer programming problem depending on which decision variable is used. If both variables are present then the model is a non-linear mixed integer programming problem. Branch and bound techniques are required either way. Removal of the binary requirement for $X_{p,a}$ (and the addition of the integer requirement for N_a) significantly reduces the size of the problem. This means that there will be $|E|$ integer decision variables instead of $\sum_p |PR_p|$ binary decision variables. This is a reduction of approximately $|P|$ times the number of variables. During experimentation it was found that the integer condition also enforced the binary requirement of the solution. This appears to be a general result of some importance. During our experimentation solving the NLP was also found to be easier than solving the equivalent but larger LP.

2.6 Constructive algorithm

A constructive approach may also be taken. For example one such approach is now described. It is based upon the logic that papers with the fewest candidates should be addressed before papers with a large number of possibilities. At each step of this iterative process, the candidates for a particular paper are ordered according to workload status, i.e. in ascending order. The first nr_p candidates are selected automatically as reviewers and their workloads are incremented. The next paper in the ordering is then addressed. This is a greedy approach which produces one solution of high quality. A random component may be included if alternative solutions are required.

2.7 Solving the problem again

In practice it is often necessary to solve the model without the complete set of data due to time considerations and other reasons. Secondly some authors may not be able to review their assigned paper(s) for a variety of reasons including insufficient time, laziness, etc. In each case new reviewers are potentially required. Modifications and additions to the previous model are therefore required. Firstly the previous assignment is defined as $X'_{p,a}$ and is a new input to the model. Secondly a binary parameter σ_a is added to the problem and signifies whether author *a* can (or is willing to) review other papers or not. Thirdly a binary parameter $\vartheta_{p,a}$ is also introduced and signifies whether reviewer *a* did not review a paper *p* that was assigned to them. The number of additional reviewers required is also defined by η_p and calculated by $\eta_p = \sum_{\forall a \mid X'_{p,a}=1} (\vartheta_{p,a})$. It should be noted that $\vartheta_{p,a} = 1 \Rightarrow X'_{p,a} = 1$ however $\vartheta_{a,p} = 0$ does not imply the reverse. The following constraints are added to the original model and are used to reduce the number of decision variables.

$$\sum_{\forall a|\sigma_a=1}(X_{p,a})=\eta_p \quad \forall p \tag{10}$$

$$X_{p,a}=(1-\vartheta_{p,a})X'_{p,a} \quad \forall a,p \mid X'_{p,a}=1 \tag{11}$$

Constraint (11) consists of two parts which are shown below:

$$X_{p,a}=0 \quad \forall a,p \mid X'_{p,a}=1, \vartheta_{p,a}=1 \tag{12}$$

$$X_{p,a}=X'_{p,a} \quad \forall a,p \mid X'_{p,a}=1, \vartheta_{p,a}=0 \tag{13}$$

In the first part, the original assignment is undone while in the second part, the original assignment is retained.

3. PROGRAM DEVELOPMENT

Developing a program of presentations is primarily the process of assigning topics to sessions, sessions to rooms, papers to sessions and papers to session slots. In order to accomplish this, the following assumptions were made:
i) Each session is held at the same time in each room on each day
ii) Each slot is of the same duration

3.1 Parameters

The index *y*, *s*, *r* and *l* are introduced and refer to days, sessions, rooms and slots respectively. A slot is a position in a session where a presentation may be given. An additional topic t^* is also introduced, namely the no-topic (or miscellaneous) topic for reasons that will be later discussed. The label of the no-topic is $|T|+1$. Additional parameters related to this problem that must be input are as follows:

Y = The number of days that the conference runs.
S_y = The number of sessions on day *y*.
$D_{y,s}$ = The duration of session *s* on day *y*.
R_y = The number of rooms available (i.e. parallel sessions) on day *y*.
UR = The set of unavailable rooms, $UR \subset \{(r,y,s) \mid y \le Y, s \le S_y, r \le R_y\}$.
US = The set of unavailable sessions, $US \subset \{(y,s) \mid y \le Y, s \le S_y\}$.
C_r = The capacity of room *r* in terms of the number of people that can be seated.
ξ_p = The presenter of paper *p*, $(p,\xi_p)\in A_p$.
LP = The length of a presentation (i.e. the duration of a slot) in minutes.

From the additional input parameters the following sets and values may be computed:

NS, ns_y = The number of sessions overall and the number of sessions on day *y*.
NSL, nsl_y = The number of slots overall and the number on day *y*.
$L_{y,s}$ = The number of slots in session *s* on day *y*.
PE_t = The potential or expected audience size for topic *t*.
$\pi_{r,t}$ = The priority value or benefit associated with assigning topic *t* to room *r*.
Ξ_a = The set of papers presented by author *a*, $\Xi_a=\{p \mid \xi_p=a\}$.

The number of sessions on day *y* is the number of rooms multiplied by the number of sessions, i.e. $ns_y=R_yS_y$. The total number of sessions is therefore, $NS=\sum_{y=1}^{Y}(ns_y)$. The number of slots on day *y* is $nsl_y=R_y\sum_{s=1}^{S_y}(L_{y,s})$ and the number of slots overall is $NSL=\sum_{y}(nsl_y)$. The number of slots available in session *s* on day *y* is the session duration divided by the length of a presentation, i.e. $L_{y,s}=D_{y,s}/LP$. It should be noted that session and presentation duration are usually chosen so that they are integer values and no leftover time occurs.

There are various indices to keep track of in this problem and it is possible to convert some of these into one in order to simplify the visual appearance of the model. Two examples are shown below.

Sessions: Over all days and rooms there are *NS* sessions. Indices *s*, *r* and *y* may be changed to one index using function, $F(y,s,r)\in[1,NS]$ as follows:

$$F(y,s,r)=\sum_{y'=1}^{y-1}(ns_{y'})+(s-1)R_y+r \tag{14}$$

If S_y and R_y are the same on each day *y* then:

$$F(y,s,r)=(y-1)S+(s-1)R+r \tag{15}$$

Note that the *y* subscript is ignored for *R* and *S*.

Slots: There are 1 to $\sum_y\left(R_y\sum_{s=1}^{S_y}(L_{y,s})\right)$ slots for presentations. Indices *y*, *s*, *l* and *r* may be changed to one index using function, $G(y,s,l,r)$ or $G(y,s,r,l)$.

$$G(y,s,l,r)=\sum_{y'=1}^{y-1}\left(R_{y'}\sum_{s'=1}^{S_{y'}}(L_{y',s'})\right)+R_y\sum_{s'=1}^{s-1}(L_{y',s'}) +R_y(l-1)+r \tag{16}$$

$$G(y,s,r,l) = \sum_{y'=1}^{y-1}\left(R_{y'} \sum_{s'=1}^{S_{y'}} (L_{y',s'}) \right) + R_y \sum_{s'=1}^{s-1} (L_{y',s'}) + L_{y,s}(r-1) + l \tag{17}$$

3.2 Decision variables

As previously mentioned program development has four components. The decision variable for each of these is as follows:

$Z_{y,s,r,t}$ = Binary variable that signifies whether the topic of session s on day y in room r is t.

$Y_{p,y,s,r}$ = Binary variable that signifies whether paper p is assigned to room r and session s on day y.

$W_{p,y,s,r,l}$ = Binary variable that signifies whether paper p is assigned to slot l of session s on day y in room r.

$V_{y,s,r}$ = The number of papers assigned to session s and room r on day y.

The number of papers assigned is related to the paper assignment in the following way, $V_{y,s,r} = \sum_p (Y_{p,y,s,r})$.

3.3 Constraints

The following constraint groupings define the complete relationships between the different components of the program development process. Which particular constraints are required in the model(s) will be discussed later.

General Constraints

$$Z_{y,s,r,t} = 0 \quad \forall y,s,r,t \mid (r,y,s) \in UR \vee (y,s) \in US \tag{18}$$

$$Y_{p,y,s,r} = 0 \quad \forall p,y,s,r \mid (r,y,s) \in UR \vee (y,s) \in US \tag{19}$$

$$W_{p,y,s,r,l} = 0 \quad \forall p,y,s,r,l \mid (r,y,s) \in UR \vee (y,s) \in US \tag{20}$$

$$Z_{y,s,r,t} \in \{0,1\},\ Y_{p,y,s,r} \in \{0,1\},\ W_{p,y,s,r,l} \in \{0,1\} \tag{21}$$

$$V_{y,s,r} \text{ integer } \forall y,s,r \tag{22}$$

Constraint (18)-(20) ensures that topics and papers can not be assigned to any room that is unavailable at the time. These constraints reduce the number of variables that must be solved for.

Topic-Session Assignment Constraints:

The constraints associated with topic to session assignment are as follows:

$$\sum_t (Z_{y,s,r,t}) = 1 \quad \forall y,s,r \tag{23}$$

$$Z_{y,s,r,t^*} \geq 1 - \sum_p (Y_{p,y,s,r}) \quad \forall y,s,r \tag{24}$$

$$\sum_r (Z_{y,s,r,t}) \leq 1 \quad \forall y,s,t \mid t \neq t^* \tag{25}$$

Constraint (23) ensures that each session must be assigned a topic. Constraint (24) ensures that a session with no talks (i.e. $\sum_p (Y_{p,y,s,r}) = 0$) must be assigned the no-topic topic. Constraint (25) ensures that each parallel session has a different topic, i.e. a topic can occur only once unless it is the no-topic.

Paper-Session Assignment Constraints:

The constraints associated with paper to session assignment are as follows:

$$\sum_y \sum_s \sum_r (Y_{p,y,s,r}) = 1 \quad \forall p \tag{26}$$

$$V_{y,s,r} = \sum_p (Y_{p,y,s,r}) \leq L_{y,s} \quad \forall y,s,r \tag{27}$$

$$Y_{p,y,s,r} \leq \sum_{\forall t \mid (p,t) \in PT} (Z_{y,s,r,t}) \quad \forall p,y,s,r \tag{28}$$

Constraint (26) ensures that each paper is assigned and only once. Constraint (27) ensures that the number of presentations in a session is valid. Thirdly constraint (28) ensures that a session of a particular topic can only contain papers of that topic. This constraint was derived by noting that if $Z_{y,s,r,t} = 1$ and $(p, t) \in PT$ then $Y_{p,y,s,r} \leq 1$ and if $Z_{y,s,r,t} = 1$ and $(p, t) \notin PT$ then $Y_{p,y,s,r} = 0$.

Paper-Slot Assignment Constraints:

The constraints associated with paper to slot assignment are as follows:

$$\sum_p (W_{p,y,s,r,l}) \leq 1 \quad \forall y,s,r,l \tag{29}$$

$$\sum_{y=1}^{Y} \sum_{s=1}^{S_y} \sum_{r=1}^{R_y} \sum_{l=1}^{L_{y,s}} (W_{p,y,s,r,l}) = 1 \quad \forall p \tag{30}$$

$$\sum_{l=1}^{L_{y,s}} (W_{p,y,s,r,l}) = Y_{p,y,s,r} \quad \forall p,y,s,r \tag{31}$$

$$\sum_{\forall r} \sum_{\forall p \in \Xi_a} (W_{p,y,s,r,l}) \leq 1 \quad \forall y,s,l,a \mid |\Xi_a| \geq 1 \tag{32}$$

Constraint (29) ensures that a slot can be assigned one paper or no papers. Constraint (30) ensures that a paper can only be assigned to one slot. Thirdly, constraint (31) defines the equivalence relationship between the paper-session and slot-session assignments. That is, a paper assigned to a session must be assigned to a slot within that session. This constraint also automatically enforces the previous constraint, thus making it redundant. Constraint (32) ensures that a presenter can only give one talk in one room at one time.

An incomplete session should have empty spaces at the end and not in the middle to ensure the continuity of the session. After solving this can be easily fixed manually for example, however the following constraint may be used instead.

$$\sum_p (W_{p,y,s,r,l}) \ge \sum_p (W_{p,y,s,r,l'}) \quad \forall y,s,r,l,l' \mid (y,s,l,l') \in adjSL \tag{33}$$

This constraint ensures that a slot can only be empty if all the slots after it are also empty. If this is not true then the slot must be assigned a presentation. Note that enforcing the criteria for adjacent slots reduces the number of constraints while maintaining the condition over the entire session. Adjacent slots are given by the following set:

$$adjSL = \left\{ \begin{array}{l} (y,s,l,l') \mid l' = l+1, l \le L_{y,s}, \\ l' \le L_{y,s}, s \le S_y, y \le Y \end{array} \right\} \tag{34}$$

Variations and Simplifications

The model utilises alternative paper keywords. However in some circumstances one keyword is sufficient. This keyword may also best describe the paper over all others. This mimics the manual process a human would take to solve the problem. For example a human would balance the number of papers in each topic (stream) by choosing a single keyword from the list of alternatives. The following constraint could then be used to enforce the correct number of sessions of a given topic if np_t is given.

$$\sum_y \sum_s \sum_r (L_{y,s} Z_{y,s,r,t}) \ge np_t \quad \forall t \tag{35}$$

It is also highly desirable for topic streams to occur in the same room in the program. This is enforced by defining a new binary decision variable $RA_{t,r}$ for the room assignment. In particular this variable signifies whether room r is assigned topic t. The following constraints are required:

$$\sum_r (RA_{t,r}) = 1 \quad \forall t \mid np_t > 0, t \ne t^* \tag{36}$$

$$RA_{t,r} = 0 \quad \forall t,r \mid np_t = 0, t \ne t^* \tag{37}$$

$$Z_{y,s,r,t} \le RA_{t,r} \quad \forall y,s,r,t \tag{38}$$

$$RA_{t,r} \in \{0,1\} \tag{39}$$

The first constraint enforces that each topic except the no-topic that has papers must be given a room. The third constraint in particular enforces that $Z_{y,s,r,t} = 0$ if $RA_{t,r} = 0$. Forcing topic streams to be held in the same room makes it likely that continuity will occur, i.e. a stream is held continuously over many sessions. To explicitly enforce this however, each topic stream may be given a starting session. For the case where the number of papers of a given topic is known this implies a fixed ending session, otherwise the ending session is also variable. This problem may alternatively be viewed as a type of parallel machine scheduling problem. For example each topic is a job and each room is a machine. The jobs must be performed on one machine only but any machine may be chosen.

3.4 Objectives

Several factors affect the construction of the program of presentations and not all are required or necessary in every problem. However in many scenarios it is very likely that more than one will be necessary thus complicating the solution process. The leading objective criteria are discussed below.

It is desirable to have sessions on a particular topic if possible. Hence the number of sessions with specific topics should be maximised or the number of sessions with no-topic should be minimised. Algebraically this is as follows:

$$\text{Minimise} \quad \sum_y \sum_s \sum_r (Z_{y,s,r,t^*}) \tag{40}$$

Incomplete sessions are also undesirable in a program. The number of these should also be minimised using the following expression:

$$\text{Minimise} \quad \sum_y \sum_s \sum_r (L_{y,s} - V_{y,s,r}) \tag{41}$$

Alternatively the number of empty sessions could be maximised. This objective however may be of little use in situations where the number of slots is very nearly equal to the number of presentations.

Topics that occur in the same room continuously over several adjacent sessions are often very desirable in a program. Two sessions are adjacent if no other session occurs between the end of one and the start of the other. The set of adjacent sections is therefore defined as follows:

$$adjSN = \{ (y,s,s') \mid y \le Y, s \le S_y, s' \le S_y, s' = s+1 \} \tag{42}$$

An objective criterion to maximise the number of adjacent sessions with the same topic in the same room is as follows:

$$\text{Maximise} \quad \sum_{\forall (y,s,s') \in adjSN} \left(\sum_{\forall r,t} (Z_{y,s,r,t} Z_{y,s',r,t}) \right) \tag{43}$$

To ensure sessions are assigned to adequate rooms, the following objective criterion is proposed.

$$\text{Maximise} \quad \sum_y \sum_s \sum_r \sum_{t \ne t^*} (\pi_{r,t} Z_{y,s,r,t}) \tag{44}$$

This objective ensures that topics with large numbers should not be assigned in the same session as they will compete for rooms with greater seating capacity. The benefit associated with assigning a topic to a room is defined as follows:

$$\pi_{r,t} = \frac{C_r}{PE_t} \quad \text{or} \quad \pi_{r,t} = C_r - PE_t \quad \forall r,t \tag{45}$$

$$PE_t = \left|\{a \mid (a,t) \in AT\}\right| \tag{46}$$

This is based upon the logic that topics that have the most papers/presentations (i.e. that are common to more people) should be held in larger rooms. It is also assumed that a person who researches in a current area will also be interested in research by other people in that same area.

At this stage an objective criterion associated with the assignment of papers to individual slots that is beneficial and worthwhile has not been found. Presenter preference however is one possibility of lesser importance that could be introduced. However this would require additional information to be input. It is expected that most presenters would like the first slot in a session so that their presentations are "out of the way" as soon as possible. Preference for certain days and sessions could therefore also be incorporated. In all wisdom however this feature should be probably be left for special cases rather than for all presenters.

3.5 Feasibility issues

Before attempting the construction of a program of presentations, there are several feasibility issues that should be noted. Firstly the number of slots must be greater than the number of presentations:

$$\sum_y \left(R_y \sum_{s=1}^{S_y} (L_{y,s}) \right) \ge |P| \tag{47}$$

Secondly the total room capacity should also be sufficient to seat all attendees at one time:

$$\sum_r (C_r) \ge |A| \tag{48}$$

Where a topic can only be held in strictly one room the number of papers must be less than the total sum of the sessions multiplied by their number of slots. This infeasibility however is unlikely to occur in many practical situations.

3.6 Solution

The model is very time consuming if not impossible to solve analytically due to the fact that there are many binary variables, and many non-linear and or multi-objective criteria. Therefore a decomposition strategy that solves a relaxed version of the problem in stages is discussed. The decomposition is not based upon the size of different parts but rather the relative importance of different aspects of the problem. More specifically the approach is based upon the way a human would solve the problem by selecting one keyword for each paper as previously discussed.

Stage 1: Assign topics to sessions and topics to rooms.

The following model minimises unused session capacity where possible. Completely empty sessions are defined as no-topic sessions and these do not contribute to the objective function. The model also does not allow miscellaneous sessions (i.e. a session with mixed topics) which may result in an infeasible model for some problem instances.

Minimise $\sum_{\forall t \mid t \ne t^*} (\varepsilon_t)$

Subject to: (23), (36), (37), (38), (39) and

$$\varepsilon_t = \sum_{\forall y,s,r} (Z_{y,s,r,t} L_{y,s}) - np_t \quad \forall t \tag{49}$$

$$\varepsilon_t \ge 0 \quad \forall t \tag{50}$$

$$\eta_t = \sum_{\forall y,s,r} Z_{y,s,r,t} \quad \forall t \tag{51}$$

$$\eta_t \text{ integer } \forall t \tag{52}$$

η_t is introduced as the number of sessions with topic t and ε_t is introduced as the free capacity in terms of unused slots. No enforcement of the binary condition for $Z_{y,s,r,t}$ is required in this model due to the integer condition for η_t.

In this model np_t does not necessarily have to be static. For example if $\alpha_{k,p}$ gives the kth keyword of paper p and $\kappa_{p,k}$ is a binary variable that signifies whether the kth keyword of paper p is selected then np_t may be defined by the following equation and included in the model:

$$np_t = \sum_p \sum_{\forall k \mid \alpha_{p,k} = t} (\kappa_{p,k}) \quad \forall t \tag{53}$$

Stage 2: Assign individual papers to sessions

The stage one model indirectly assigns papers to sessions. That is, it assigns general numbers of papers to sessions but does not assign specific papers. No objective is required in this stage.

Solve (26), (27), (28) and $V_{y,s,r}$ integer $\forall y,s,r$

A model for the assignment of papers to slots is unnecessary (and unimportant) after stage 1 and 2 have been performed. This is because there are no constraints other than a session continuity constraint, i.e. (33).

4. CASE STUDY

4.1 Implementation and data collection

Keyword selection is vitally important when assigning reviewers and when constructing a program of presentations. For best results, the right keywords should be defined. Authors of papers should define these directly from a given list. Some authors however will inevitably define the wrong topics or totally different topics that are not in the topic list. This means that each paper must be physically inspected and modified where necessary. Keyword priority and ordering is also important and may affect the planning processes.

Construction and maintenance of the database is also critical. Multiple entries (i.e. duplications) for example and other errors makes the application of OR techniques next to impossible. A database is necessary to store all author names, addresses, papers, and so forth.

4.2 Data

The models were applied to the 5th Asia Pacific Industrial Engineering and Management Systems Conference (APIEMS) 2004 Gold Coast, Australia. For this conference there were 460 papers associated with 51 topics (52 is the no-topic), and approximately 1000 authors and co-authors. The conference duration was three days and on each day there were 2, 5, and 3 sessions planned respectively. There were also eight parallel tracks (i.e. eight rooms). The maximum number of papers that could be presented in each session was four and each presentation was 20 minutes.

4.3 Results

The models were solved using the latest version of Lingo on a P4 with 2.2Mhz speed.

Paper-Reviewer Assignment

The problem was solved exactly due to the removal of the binary constraint for $X_{p,a}$ and the inclusion of an integer constraint for N_a. Otherwise the problem was too large to be solved in a reasonable amount of time. Consequently for this problem there were 44557 variables, 1995 integers, and 5345 constraints. The CPU time was 1 min 3 sec and the objective function value was 505.08. The solution which is a 460 x 1000 matrix and a 1 x 1000 vector however is quite sparse and is too large to display.

Program Development

After the review process there were 302 accepted papers to be placed in the program of presentations. A single keyword was chosen that best described each paper. Table 1 below shows the number of papers associated with each topic after this process.

A model with 28714 variables, 420 integers and 28976 constraints resulted for the complete problem. This model however was too large to be solved in reasonable time and therefore it was split into two parts and subsequently solved. The solution details are as follows:

Topic to Session & Topic to Room Assignment: 4554 variables, 340 integers, 4434 constraints, CPU 1 min 46 sec, 144019 iterations, 446 branch and bound steps, objective value 5.

Paper to Session Assignment: 24576 variables, 80 integers, 24543 constraints, CPU 7 sec, 166 iterations.

The stage 1 solution is shown in Table 2. This solution efficiently utilises session capacity because the sessions are fully utilised for all of the topics other than the no-topic. This is shown by the following vector which is associated with the free capacity of topics variable.

$\underset{\sim}{\varepsilon}$ = (0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1,0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 45)

In this solution a number of topic streams are not held continuously in the program. A manual reordering can correct this if required. For example session topics may be swapped if the session capacity in terms of slots is the same. An example of an efficient re-ordering is shown in Table 3. From this table it is clear that all topics that require more than one session are held continuously in their assigned room. An automated approach may also be taken to perform the reordering but requires an additional model and or procedures.

The results in Table 3 were then input into the stage 2 model and it was solved. The solution shown in Table 4 was obtained and is a solution to the complete problem. Note that the dark grey filled squares show slots not built into the program. The light grey filled squares however show the slots that are not utilised in the current solution.

Table 1. The number of papers per topic

Topic	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
#	22	0	0	4	4	4	5	0	8	0	5	4	10	5	0	0	4	19	0	10
Topic	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40
#	11	0	8	4	0	3	0	0	0	8	8	0	4	11	9	5	9	8	3	8
Topic	41	42	43	44	45	46	47	48	49	50	51	52								
#	31	13	5	0	26	0	0	4	8	4	8	0								

Table 2. Topic to room and session assignment results

Day	Session (# of paper)	Room 1	Room 2	Room 3	Room 4	Room 5	Room 6	Room 7	Room 8
1	1(5)	37	36	18	49	52	20	52	52
	2(4)	1	9	30	42	41	21	31	5
2	1(4)	1	45	12	49	41	21	51	26
	2(4)	37	9	35	50	38	4	51	34
	3(5)	7	45	35	42	41	20	52	52
	4(5)	1	11	18	43	41	13	52	52
	5(4)	17	45	18	42	41	23	6	39
3	1(4)	1	45	48	40	38	23	33	34
	2(4)	24	45	30	40	41	21	31	34
	3(5)	1	45	18	52	41	13	52	14

Table 3. Results after a manual reordering

Day	Session (# of paper)	Room 1	Room 2	Room 3	Room 4	Room 5	Room 6	Room 7	Room 8
1	1(5)	7	36	35	49	52	20	52	52
	2(4)	1	9	35	49	38	21	51	5
2	1(4)	1	9	30	42	38	21	51	26
	2(4)	1	45	30	42	41	21	6	39
	3(5)	1	45	18	42	41	13	52	52
	4(5)	1	45	18	43	41	13	52	52
	5(4)	17	45	18	50	41	23	31	34
3	1(4)	24	45	12	40	41	23	31	34
	2(4)	37	45	48	40	41	4	33	34
	3(5)	37	11	18	52	41	20	52	14

Table 4. Paper to session and paper to slots assignment solution

Day	Session (# of papers)	Room 1					Room 2					Room 3					Room 4				
		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	1(5)	44	66	79	328	432	251	330	338	380	410	57	117	247	249	466	221	282	287	317	341
	2(4)	193	295	297	390		212	234	271	435		48	49	51	53		173	182	209		
2	1(4)	302	306	309	324		6	80	102	144		168	319	326	425		409	452	460	465	
	2(4)	343	344	348	356		101	131	211	49		34	98	127	156		461	462	463	464	
	3(5)	305	386	387	389		224	245	248	252	254	60	81	87	90	430	180	225	250	342	442
	4(5)	14	74	123	146	194	256	272	312	359	445	108	116	235	299	301	13	283	411	436	447
	5(4)	4	154	395	397		370	381	415	423		307	308	310	376		331	332	335	336	
3	1(4)	7	313	437	443		255	426	427	433		133	155	267	419		226	242	311	402	
	2(4)	261	262	407	456		64	69	137	164		8	29	401	434		11	67	121	151	
	3(5)	62	95	233	259	260	177	200	231	246	399	15	16	24	36	41					

Day	Session (# of papers)	Room 5					Room 6					Room 7					Room 8				
		1	2	3	4	5	1	2	3	4	5	1	2	3	4	5	1	2	3	4	5
1	1(5)						105	265	316	318	347										
	2(4)	189	190	297	288		91	92	138			368	369	371	440		228	263	264	420	
2	1(4)	93	109	152	176		147	171	188	205		17	362	365	367		178	217	428		
	2(4)	218	453	454	455		3	77	83	84		28	118	273	441		33	112	391		
	3(5)	128	132	135	158	160	300	303	304	403	451										
	4(5)	111	163	170	184	215	37	99	294	296	298										
	5(4)	229	232	269	372		315	360	418	446		291	340	424	448		55	114	400		
3	1(4)	110	275	289	345		46	52	219	220		71	72	73	75		143	357	392	293	
	2(4)	63	227	374	439		12	38	175	213		113	134	136	414		30	39	43		
	3(5)	40	58	59	61	70	20	31	42	56	96						142	214	314	354	355

5. CONCLUSIONS

In this paper generic optimisation models were developed to help solve two significant conference planning tasks. The models were implemented and tested on real life data. The models for the paper review process were quite sufficient and could be solved immediately. The program development problem however was much larger and complex and required a decomposition approach to obtain feasible solutions analytically. The effort required to perform these activities was minimal using the proposed optimisation models in comparison to a manual approach.

A number of extensions became visible in the process of the solving the two problems and may be investigated in future work. For example the grouping of like topics may be performed. Fewer topics mean less assignments and a reduced problem size. A simple iterative procedure can be used to convert the data. Alternatively, grouped topics may be introduced as additional topics. Existing topics could also be retained.

The order of keywords in terms of importance can make a big difference. The question of which topic(s) best describes a paper however must be answered. Author's recommendations may be taken otherwise obtaining this information is difficult and or time consuming. Assigning suitable weightings (priorities) for each keyword may be used.

The complete model for the second problem could not be solved analytically because of the search space size and complexity. More efficient meta-heuristics could therefore be used in future and are recommended. Recent literature on course timetabling and other more complex assignment problems corresponds with this conclusion. A comparison between this approach and the decomposition approach may then be made.

REFERENCES

1. Burdett, R. and Kozan, E. (2004). The assignment of individual renewable resources in scheduling. *Asia Pacific Journal of Operational Research*, 3: 355-377.
2. Burke, E.K., Elliman, D.G., and Weare, R.F. (1995). A hybrid genetic algorithm for highly constrained timetabling problems. *Proceedings of the 6th International Conference on Genetic Algorithms*, San Francisco, Morgan Kaufman, pp. 605-610.
3. Burke, E.K. and Newall, J.P. (1999). A multi-stage evolutionary algorithm for the timetable problem. *IEEE Transactions on Evolutionary Computation*, 3(1): 63-74.
4. Burke, E.K. and Newall, J.P. (2004). Solving examination timetabling problems through adaptation of heuristic orderings. *Annals of Operations Research*, 129: 107-134.
5. Burke, E.K., Bykov, Y., Newall, J.P., and Petrovic, S. (2004). A time-predefined local search approach to exam timetabling problems. *IIE Transactions on Operations Engineering*, 36(6): 509-528.
6. Burke, E.K., Petrovic, S., and Qu, R. (2006a). Case based heuristic selection for timetabling problems. *Journal of Scheduling*, 9(2): 115-132.
7. Burke, E.K., MacCarthy, B., Petrovic, S. and Qu, R. (2006b). Multiple-retrieval case based reasoning for course timetabling problems. *Journal of the Operational Research Society*, 57: 148-162.