A CONSTRAINT PROGRAMMING APPROACH FOR A CONFERENCE TIMETABLING PROBLEM

Emrah B. Edis, Department of Industrial Engineering, Dokuz Eylul University, DEU, Endustri Muh. Bol., Bornova, 35100, Izmir, TURKEY, + 90-232-3881047, emrah.edis@deu.edu.tr
Pinar Mizrak Ozfirat, Department of Industrial Engineering, Dokuz Eylul University, DEU, Endustri Muh. Bol., Bornova, 35100, Izmir, TURKEY + 90-232-3881047, pinar.mizrak@deu.edu.tr
Irem Ozkarahan, Computer Science Department, Troy University, Gunter Annex, 100 Turner Blvd. Bldg. 826 Montgomery, AL 36103-4419, USA, ++1-334-8327293, iozkarahan@troy.edu

ABSTRACT

This study provides a solution approach to prepare the timetable of 27th National Conference on Operations Research/Industrial Engineering (OR/IE 2007) which was held in July 2007 at Dokuz Eylul University, in Izmir, Turkey. The problem is constituted of nearly 300 presentations to be held in 10 different time slots and in six parallel tracks. For this reason, 30 different topics are identified and the presentations are listed accordingly. The problem is to assign simultaneously the topics and presentations to sessions. Since the timetable becomes too complex to be solved by classical integer programming methods in a reasonable time, constraint programming approach of artificial intelligence is used to produce the timetable. The proposed approach has produced practical solutions in a reasonable time satisfying not only the conference organization committee but also preferences of attendees.

INTRODUCTION

Conference timetabling involves scheduling presentations into time periods and rooms. When doing so, two feasibility issues must be satisfied. These are to avoid individual schedule conflicts and to assign all presentations to a certain time slot and room. There may also exist additional requirements due to the type and activities of the conference such as arranging tracks according to the presentation topics etc. Conference timetabling has not received a great deal of attention in prior research literature [1]. An earlier paper by Eglese and Rand [2] introduces preference-based conference scheduling problem. They collected session preference data from participants and used simulated annealing to solve the problem. Sampson and Weiss [3] [4], and Sampson [1] extend the concept of preference-based conference scheduling. Biba *et al*. [5] present a rule-based expert system to handle the scheduling activities of conference papers. Recently, Nicholls [6] proposes a simple heuristic which allocates conference papers to sessions considering both presenter and attendee requirements. As seen from the literature, none of the papers use constraint programming (CP) as a solution approach. CP is an alternative method for this type of combinatorial problems where the feasibility is major issue. Defining global constraints in a declarative framework of a CP model makes a dramatic reduction in the problem size.

In this study, a CP approach is proposed which works out the requirements of the attendees' with the feasibility constraints of the timetable. The solution approach is employed to prepare the timetable of 27th National Conference on Operations Research / Industrial Engineering (OR/IE 2007) which was held in July 2007 at Dokuz Eylul University, Turkey. The proposed iterative CP approach has produced practical solutions in a reasonable time satisfying not only the conference organization committee but also the attendees. In the following sections, the definition of the conference timetabling problem with required notation, the details of the proposed approach and the summary are given respectively.

PROBLEM DEFINITION

The problem is constituted of nearly 300 presentations to be held in 10 different time periods and in six parallel sessions. Also, the presentations are required to be arranged according to their topics. For this reason, 30 different topics such as mathematical modeling, simulation, ergonomics etc. are identified and the presentations are listed accordingly. Considering the number of presentations with same topic and maximum number of presentations that can be assigned to a session, 60 session-names (e.g., Simulation I, Simulation II, Simulation III, Mathematical Modeling I, Mathematical Modeling II, Ergonomics, etc.) are defined to be assigned to six sessions through 10 time periods. Then, the problem is to assign simultaneously the topics and the presentations to sessions. The notation is given below:

Indices:

i	index for session-names to be assigned to sessions through time periods, $i = 1,..,60$
j	index for topics, $j = 1,.., 30$
p	index for periods, $p = 1,..,10$
k	index for presentations, $k = 1,.., 293$
a	index for registered authors that have more than one presentation, $a = 1,\ldots,20$

Parameters:

undesired[*k*,*p*]	takes value one if *p* is an undesired period for presentation *k*, and zero otherwise
pres_topic[*k*]	topic index of presentation *k*
pres_of_authors[*k*,a]	takes value one if *a* is one of registered authors for presentation *k*, zero otherwise
topic_of_session [*i*]	topic index of session-name *i*

Decision variables:

*assign*1[*i*]	the period index of session-name *i*
temp[*p*,*j*]	takes value one if period *p* covers topic *j* and zero otherwise
*assign*2[*k*]	the period index of presentation *k*

PROPOSED SOLUTION APPROACH

The proposed CP approach handles planning of timetable in two phases. In the first phase, Topic To Sessions (*TTS*) model assigns the session-names (accordingly topics) to sessions along time periods:

$$alldifferent(all(i \text{ in } 1..60\text{: } topic\_of\_session[i] = j)\ assign1[i]) \qquad j = 1,\ldots,30 \qquad (1)$$

$$\sum_{i=1}^{60} (assign1[i] = p) = 6 \qquad p = 1,\ldots,10 \qquad (2)$$

Equation (1) ensures that topics assigned to parallel sessions should be different within each period. For example, two simulation sessions should not be held within the same period so that attendees would be able to follow presentations on the same topic. All time periods have exactly six parallel sessions as given in Equation (2). It should also be noted that the data including desired time periods and other requests of attendees are converted to equations and added into *TTS* model. Fixing the values of some variables in this manner accelerates the proposed procedure. As the *TTS* model is solved, the data on topics of each session are transferred to second phase through variables *temp*[*p*,*j*] by Equation (3):

$$temp[assign1[i], topic\_of\_session[i]] = 1 \qquad i = 1,\ldots,60 \qquad (3)$$

In the second phase, Presentations To Sessions (*PTS*) model (using fixed topics of each session) assigns presentations to time periods subject to a number of constraints:

$$\sum_{k|pres\_topic[k]=j} (assign2[k] = p) \geq 4 \qquad p = 1,\ldots,10;\ j = 1,..30 \mid temp[p,j] = 1 \qquad (4)$$

$$\sum_{k|pres\_topic[k]=j} (assign2[k] = p) \leq 5 \qquad p = 1,\ldots,10;\ j = 1,..30 \mid temp[p,j] = 1 \qquad (5)$$

$$alldifferent(all(k \text{ in } 1..293: pres\_of\_authors[k,a] = 1)\ assign2[k]) \qquad a = 1,\ldots,20 \qquad (6)$$

$$\text{if } undesired[k,p] = 1 \text{ then } assign2[k] <> p \text{ endif}; \qquad k = 1,\ldots,293,\ p = 1,\ldots,10 \qquad (7)$$

Equation (4-5) states that each session is made up of four or five presentations with respect to the session-topics transferred from the *TTS* model. Equation (6) ensures that the presentations of attendees who present more than one paper should be scheduled in different periods. Preferences of attendees in terms of undesired periods are satisfied by Equation (7). Finally, it should also be noted that a problem-specific search procedure based on the first-fail principle is embedded into the *PTS* model in order to prune the search tree in the earlier stages and reach feasible solutions in advance.

These two CP models are written in OPL Studio 3.7 [7] optimization software. The two phases are integrated by means of an OPL Script [7, pp.311-370] algorithm. This script runs firstly the *TTS*; takes the topic of each session from this model; inputs it into *PTS* and runs *PTS* to find the final conference schedule. However, the solution may turn out to be infeasible if the schedule of topics found in *TTS* conflicts with the constraints in *PTS*. Therefore, the script works iteratively until the constraints in the second phase are in harmony with the schedule found in the first phase.

CONCLUSION

In this study, a CP approach for a conference timetabling problem is developed. The proposed approach works in two phases in which it assigns topics to sessions in the first phase and schedules presentations to sessions in the second phase. Both timetabling feasibility constraints and attendee requirements are handled in the models. The proposed approach is used to prepare the timetable of the National Turkish OR/IE 2007 Conference. It has produced practical solutions in a reasonable time satisfying not only the conference organization committee but also the preferences of attendees. The study revealed that CP may also be a useful tool for these types of problems. The proposed procedure may be applicable to other timetabling problems with some variations.

REFERENCES

[1] Sampson, S.E. Practical Implications of Preference-Based Conference Scheduling. *Productions and Operations Management*, 2004, *13* (3), 205-215.

[2] Eglese, R. W. & Rand G.K. Conference seminar timetabling, *Journal of Operational Research Society*, 1987, *38* (7), 591-598.

[3] Sampson, S.E. & Weiss, E.N. Increasing service levels in conference and educational scheduling: A heuristic approach. *Management Science*, 1995, *41* (11), 1816-1825.

[4] Sampson, S.E. & Weiss, E.N. Designing conferences to improve resource utilization and participant satisfaction. *Journal of Operational Research Society*, 1996, *47* (2), 297-314.

[5] Biba, M., Ferilli, S., Di Mauro, N. & Basile, T.M.A. Intelligent Methodologies for Scientific Conference Management. *ISMIS 2006, Lecture Notes in Artificial Intelligence*, 2006, 258-267.

[6] Nicholls, M.G. A small to medium sized conference scheduling heuristic incorporating presenter and limited attendee preferences. *Journal of Operational Research Society*, 2007, *58*, 301-308.

[7] ILOG OPL Studio 3.7, Language Manual, *ILOG S.A.*, 2003