SYSTEM DEMONSTRATION
Timetabling a University Dental School

Hadrien Cambazard[1], Barry O'Sullivan[1], John Sisk[1],
Robert McConnell[2], and Christine McCreary[2]

[1]Cork Constraint Computation Centre, University College Cork, Ireland
[2]Dental Surgery School, University College Cork, Ireland
{h.cambazard|b.osullivan|j.sisk}@4c.ucc.ie,
{r.mcconnell|c.mccreary}@ucc.ie

Abstract. We present a constraint model for a real-world University Dental School Timetabling problem, similar to post-enrolment timetabling. This model is used in a timetabling system we have developed for this application.

1 Introduction

We present a timetabling problem that arises in the Dental School of University College Cork (UCC). Due to an increasing number of students the school has begun considering automated approaches to generating the timetable for their five year academic programme. A diverse variety of university timetabling problems exist, but three main categories have been identified [3, 2, 5]: school, examination and course timetabling. The Dental School at UCC is dealing with a problem, which is similar to the post-enrolment university course timetabling problem [4] that occurs in a context whereby a set of courses that have been chosen by students must be scheduled into timeslots. University timetabling problems are usually solved using local search methods coupled with a wide range of meta-heuristics. Complete methods such as constraint programming (CP) are not yet scalable to real world instances. However, the problem tackled here is relatively small and we took this opportunity to develop a CP approach reusing many of the simple modeling ideas from [1]. A key feature of the resulting system is that it can prove that the standard set of constraints used by the Dental School admits no feasible timetables in the context of increasing student numbers. Therefore, the school must perform simulations to study potential remedies for this situation, e.g. by increasing capacity or by opening new timeslots. The application has been tested at UCC and has been embedded within an online timetabling system.

2 Problem Definition

The weekly timetable to be designed comprises two or three timeslots per day (morning, midday and afternoon) for five days which gives ten to fifteen timeslots. A number of events are to be timetabled. The events taking place are each characterized by a group of students and a subject. Several subjects exist, e.g "Dental Surgery", "Ortho", "OTL/Tutorial", "OTL/Pros", "Paedo", "Restorative", and "Study" (see Figure 1).

		Monday	Tuesday	Wednesday	Thurday	Friday
Morning	Dental_Surgery	[0/10]	4.2(8)[8/10]	4.3(8)[8/10]	4.1(9)[9/10]	4.4(8)[8/10]
	Ortho	4.4(8)[8/10]	4.3(8)[8/10]	4.2(8)[8/10]	5.2(9)[9/10]	1.1(10)[10/10]
	OTL_Tutorial	[0/12]	4.5(7)[7/12]	4.4(8)[8/12]	4.2(8)[8/12]	[0/12]
	OTL_Pros	3.1(11)[11/20]	3.1(11)[11/20]	3.1(11)[11/20]	3.2(10)3.3(10)[20/20]	3.2(10)3.3(10)[20/20]
	Paedo	[0/10]	4.4(8)[8/10]	4.1(9)[9/10]	4.3(8)[8/10]	[0/10]
	Restorative	5.1(10)5.2(9)5.4(9)[28/32]	5.1(10)5.2(9)5.4(9)[28/32]	5.1(10)5.3(9)5.4(9)[28/32]	5.1(10)5.3(9)5.4(9)[28/32]	4.1(9)4.2(8)4.3(8)4.5(7)[32/32]
	Study	[0/18]	5.3(9)4.1(9)[18/18]	5.2(9)4.5(7)[16/18]	4.4(8)[8/18]	[0/18]
Midday	Dental_Surgery	[0/10]			5.1(10)[10/10]	[0/10]
	Ortho	1.1(10)[10/10]			4.5(7)[7/10]	5.4(9)[9/10]
	OTL_Tutorial	[0/12]			4.1(9)[9/12]	4.3(8)[8/12]
	OTL_Pros	3.4(10)3.5(10)[20/20]			3.2(10)3.5(10)[20/20]	3.3(10)3.4(10)[20/20]
	Paedo	[0/10]			4.2(8)[8/10]	5.3(9)[9/10]
	Restorative	5.3(9)3.2(10)3.3(10)[29/32]			5.2(9)5.3(9)3.4(10)[28/32]	5.2(9)3.1(11)3.5(10)[30/32]
	Study	[0/18]			5.4(9)4.3(8)[17/18]	5.1(10)4.2(8)[18/18]
Afternoon	Dental_Surgery	[0/10]	5.2(9)[9/10]	5.3(9)[9/10]	4.5(7)[7/10]	5.4(9)[9/10]
	Ortho	[0/10]	5.3(9)[9/10]	5.1(10)[10/10]	4.1(9)[9/10]	[0/10]
	OTL_Tutorial	[0/12]	[0/12]	[0/12]	[0/12]	[0/12]
	OTL_Pros	[0/20]	3.4(10)3.5(10)[20/20]	[0/20]	[0/20]	[0/20]
	Paedo	[0/10]	5.4(9)[9/10]	5.2(9)[9/10]	5.1(10)[10/10]	4.5(7)[7/10]
	Restorative	4.1(9)4.2(8)4.3(8)4.5(7)[32/32]	4.1(9)4.2(8)4.4(8)4.5(7)[32/32]	4.1(9)4.3(8)4.4(8)4.5(7)[32/32]	4.2(8)4.3(8)4.4(8)[24/32]	4.4(8)2.1(10)[18/32]
	Study	[0/18]	5.1(10)[10/18]	5.4(9)[9/18]	5.2(9)5.3(9)[18/18]	[0/18]

Fig. 1. An example of a solution to the UCC Dental School timetabling problem.

Each has a specific maximum *capacity*, limiting the number of students who can attend the class at the same time. In other words, the same subject is always taught in the same room because each subject needs specific equipment. In a pre-processing step performed by the school, the students of each year are allocated to groups, which simply correspond to sets of students who are following the same set of subjects. A student group is defined by its *curriculum*, i.e., a multiset of subjects, and a *size*, i.e. the number of students in the group. Notice that the same subject can occur several times in a weekly period. For example, if group G_1 comprises ten students who have to attend one "Ortho" session twice during the week, one "Restorative" session and one "Paedo" session, $size(G_1) = 10$ and the curriculum, $c(G_1), is\{Ortho, Ortho, Restorative, Paedo\}$.

The events allocated in the timetable are basically defined by the curriculum of the groups. Event $e = (i, j)$ is associated with group i and the jth element of the curriculum of i. For example event $e = (1, 2)$ represents Group 1 attending "Ortho". The goal is to allocate all events to a timeslot of the timetable knowing that:

1. a group can only attend one subject in a given timeslot;
2. a group must attend all the subjects of its curriculum;
3. the number of students assigned to a given subject and timeslot must be smaller than the capacity of the subject;
4. some pairs of groups cannot follow the same subject at the same time;
5. some timeslots are initially forbidden for some groups and subjects.

Figure 1 presents an example of a solution to the problem. The groups are labelled $X.Y(s)$ in this example where X denotes the year of the group, Y its index and s is the size of the group. With each subject, we also indicate in square brackets the amount of space used compared to the space available. Restorative the monday morning is filled with 28 students out of the 32 seats available.

3 The Constraint Model

We use the following notation:

- S: the set of subjects. For a subject $s \in S$, $capa(s)$ is its capacity.
- G: the set of groups. For a group $g \in G$, we denote by $c(g)$ the curriculum of g and $size(g)$ the number of students in g.
- E: the set of events. For a given event $e \in E$, its group and index of the subject of the corresponding group's curriculum will be written $grp(e)$ and $sub(e)$. For $e = (1, 2)$, $sub(e) = 2$ and $grp(e) = 1$.
- I: the set of pairs of events with the same subject that are incompatible because the corresponding pair of groups cannot attend the subject at the same time.
- F : the set of events, timeslots pairs expressing the initial forbidden timeslots for some specific group and subject.
- nbt : the number of timeslots.

The timetabling is done using a Java open source constraint programming (CP) system, *Choco*[1]. The CP model is based on a variable x_{ij} per event expressing the timeslot in which the event is scheduled: $\forall i \leq |G|, j \leq |c(i)|, x_{ij} \in \{0, \ldots, nbt - 1\}$. The constraints are the following:

$$\begin{array}{ll}
C_1 : \forall\, i < |G| & \textsc{AllDifferent}(x_{i1}, \ldots, x_{i,|c(i)|}) \\
C_2 : \forall\, t < nbt, s < |S| & \sum_{e=(i,j)\in E \text{ with subject s}} (x_{ij} = t) \times size(i) \leq capa(s) \\
C_3 : \forall (e_1, e_2) \in I & x_{grp(e_1),sub(e_1)} \neq x_{grp(e_2),sub(e_2)} \\
C_4 : \forall (e_1, s) \in F & x_{grp(e_1),sub(e_1)} \neq s
\end{array}$$

Constraint C_1 states that a group cannot be in two rooms at the same time whereas C_2 enforces the total number of students attending a given subject to be lower than the capacity of corresponding subject/room; recall that subjects are taught in uniquely equipped rooms. C_3 states the incompatible groups and C_4, the forbidden timeslots.

The problem can be seen as list coloring, where events can be mapped to nodes and timeslots to colors, combined with knapsack constraints for each color limiting the *amount* of color that can be used in the coloring. One subject defines a 0-1 multi-knapsack problem, *i.e* a bin-packing. The size of this timetabling problem is relatively small, and this constraint model turns out to be efficient in practice. Nevertheless, hard instances were met when increasing the number of timeslots to accomodate the new students. Several redundant constraints can be added to improve performance:

1. If two events cannot fit together for a given subject because of their size, a redundant inequality constraint can be added to the model. These constraints are redundant given the knapsack but the presence of maximum cliques can considerably strengthen the constraint propagation.
2. Cliques in the coloring graph can be exploited using ALLDIFFERENT constraints.
3. Among all the events that must be assigned to a given subject, we can compute the k smallest that overload the capacity of the subject. The number of occurrences of a given value (timeslot) amongst all the events that are related to the corresponding subject is, therefore, bounded by k. A GLOBALCARDINALITYCONSTRAINT can be stated over all the corresponding events enforcing an upper bound on the number of occurrences of each timeslot to k. Notice that this constraint is a relaxation of all the knapsacks associated with a given subject, but it performs reasoning on

[1] `http://choco.emn.fr`

all the timeslots together as opposed to the knapsack constraints which operate independently on each timeslot.

4. All events related to the same group and the same subject are symmetrical. Such a set of events $e_1, \ldots, e_k$ can be ordered: $x_{grp(e_1),sub(e_1)} < x_{grp(e_2),sub(e_2)} < \ldots < x_{grp(e_k),sub(e_k)}$ as all permutations of these variables in the same solution are also solutions.
5. All events corresponding to groups of the same size with the same initial domain and the same curriculum are also symmetrical and can be permuted in any solution. We can safely order them in the timetable as explained above.

Redundant Constraints 1, 2, 4, 5 turn out to be critical to prove inconsistency of some instances. The problem differs from the Post Enrolment University Course Timetabling Problem [4] in terms of "room allocation". In the Dental School problem a set of events allocated to the same timeslot is subject to a knapsack constraint rather than a matching (events-room) constraint. The model presented here augmented with the redundant constraints seems to be efficient, but the decomposition strategy presented in [1] could be applied. In this context we would postpone the resolution of the knapsacks (using only the relaxation based on the redundant Constraint 3) once a coloring has been found, and infer cuts expressing that subsets of events cannot be together.

4 Conclusion

We have developed a simple timetabling system based on constraint programming for the Dental School at University College Cork. Its ability to prove inconsistency is its main originality and offers to the school a tool to simulate various scenarios to deal with increasing student numbers. A online application was designed and proved to be very useful to the school. We plan to add the computation of explanations to our system to offer feedback to the school to help overcome situations where no consistent timetables exist. By comparison, traditional timetabling system focus on minimizing the degree of violation of hard constraints to counter inconsistency. We believe that it is interesting to evaluate both approaches on this problem from the user's perspective.

Acknowledgements. Supported by Science Foundation Ireland (Grant 05/IN/I886).

References

1. Hadrien Cambazard, Emmanuel Hebrard, Barry O'Sullivan, and Alexander Papadopoulos. Local search and constraint programming for the post-enrolment-based course timetabling problem. *accepted to Annals of Operation Research*, 2009.
2. M. W. Carter and G. Laporte. Recent developments in practical course timetabling. In *PATAT*, pages 3–19, 1997.
3. D. de Werra. An introduction to timetabling. *European Journal of Operational Research*, 19(2):151–162, February 1985.
4. R. Lewis, B. Paechter, and B. McCollum. Post enrolment based course timetabling: A description of the problem model used for track two of the second international timetabling competition. Technical report, Cardiff University, 2007.
5. A. Schaerf. A survey of automated timetabling. *Artificial Intelligence Review*, 13(2):87–127, 1999.