Standardization of Constraints for the Second International Timetabling Competition Problem Instances

Atish Chand
School of Computing, Information and Mathematical Sciences,
Faculty of Science and Technology,
University of the South Pacific,
chand_at@usp.ac.fj, Phone +679 – 3232219, Fax +679 - 3231527

Abstract

University timetabling problem hardness and solution acceptability has been difficult to compare due to the variety of constraints, data formats and categories of university timetabling problems. This paper shows the commonality in the various university timetabling tracks' constraints and specifies how a generic standard model can represent the various constraints.

Keywords: University Timetabling, Standardization, Constraint, Modeling

Introduction

Success of the first International Timetabling Competition (Mc Collum et al 2007), led to the Second International Timetabling Competition Committee in three tracks (Lewis et al 2007, Gasperol et al 2007, Mc Collum et al 2007): examination timetabling, curriculum timetabling and post-enrolment timetabling, to allow for better comparison of timetabling solutions. The Second International Timetabling Competition has endeavored to provide university timetabling constraints that are closer to real world timetabling data (Lewis et al 2007, Gasperol et al 2007, Mc Collum et al 2007). This paper shows how the constraints of various tracks of university timetabling problems can be represented by a standard model that has been proposed by Chand (2004, 2005). Various methods have been devised to compare university timetabling problem hardness (Burke et al 2004), (Kostuch et al 2004), (Lewis et al 2007), (Gasperol et al 2007), (Mc Collum et al 2007) but these apply to specific timetabling environments.

Standardization of University Timetabling Constraints

The proposed standard format consists of two types of constraints: domain and spread. Domain constraints specify which timetabling sessions (which may be exams, lectures, tutorials or any other events) are restricted to which domain elements (such as days, periods, timeslots, rooms or timetable slots). . Domain constraints are specified by the generic representation: Domain_Constraint((Session_List), (Domain_List), Hardness). The name of the actual domain replaces Domain such as day, hour, room etc. Session_List is the list of sessions that that must be allocated to a domain element that must be in the Domain_List. Hardness is a numeric value corresponding to the hardness of the constraint. A Hardness value of absolute 100 indicates the constraint is hard. Absolute values of Hardness between 100 and zero indicate the relative softness of the constraint.

The generic representation of a spread constraint is:

Domain_Spread ((Session_List), (Spread_List), Direction, Hardness) where the domain is one of the timetabling resources or cross product of the resources. Session_List is the list of sessions that need to be spread over the domain. Spread_List is a list of numbers that the sessions in Session_List should be spread by. That is, the difference between the domain element allocated to a session in Session_List and the domain element allocated to any session appearing later in the list, should be equal to a value in the Spread_List.

Standardization of Exam timetabling Constraints

The ITC examination track (Mc Collum et al 2007) specifically defines two categories of hard constraints in its input file: 'Period Related Hard Constraints' and 'Room Related Hard Constraints'. Though it does not specifically define soft constraints in its input file similar to its specific definition of hard constraints, it clearly outlines the soft constraints in the section 'Institutional Model Weightings' in the input file. These constraints are enumerated E (i) to E (ix). Its period hard constraints are E (i) Exam Coincidence E (ii) Exclusion and E (iii) After. The room hard constraint E (iv) is that an exam must be timetabled in a room by itself. The soft constraints are: E (v) "Two in a row", E (vi) Two in a day, E (vii) Period spread, E.(viii) Non-mixed durations and E(ix) Frontload.

The following section describes how the standard model can represent each of the above-mentioned exam timetabling constraints. E (i) Exam Coincidence Time_Spread ((S1, S2), (0), 0, 100) specifies that the two exams should be held at the same time.

E (ii) Exclusion. Time_Spread ((S1, S2), (0), 0, -100) specifies that the two exams should not be held at the same time. E (iii) After. Time_Spread ((S1, S2), (0), 1, -100) specifies that the S2 should be scheduled after S1. E (iv) An exam must be timetabled in a room by itself. Room_Spread ((S1), (0), 0, 100) specifies that the exam S1 should be scheduled in the same room. E(v) Two in a row. : Time_Spread ((S1, S2), (0,1), 0, -90) specifies that the two exams should be preferably not scheduled at the same time or one after the other. E(vi) Two in a day.: Day_Spread ((S1, S2), (0), 0, -90) specifies that the two exams should be preferably not scheduled on the same day. E(vii) Period spread. If exams S1 and S2 should be spread over 5 periods, then the constraint: Period_Spread ((S1, S2), (5), 0, 90) specifies that the two exams should be preferably spread over 5 periods. E (viii) Non-mixed durations. If S1 and S2 are two exams of different durations then the constraint Time_Spread ((S1, S2), (0), 0, -90) specifies that the two exams should be preferably not scheduled at the same time. E(ix) Front load. If S1 and S2 are two exams that should be held earlier in the examination period, then the constraint Time_Constraint((S1, S2), (T1, T2, T3), 90) indicates that .the exams S1 and S2 should be preferably scheduled in the first three timeslots or other initial timeslots as required.

.**Standardization of Curriculum Timetabling Constraints**

Whereas the examination timetabling input file described all of its constraints, the curriculum data input file (Gasperol et al 2007) describes only one type of constraint: unavailability constraint that specifies that a particular course cannot be scheduled in a particular period. This constraint is a domain constraint that restricts certain courses (or exams or sessions) to certain time slots. Unlike the examination-timetabling problem that specifies the constraints within its input file, the curriculum problem specifies other constraints in its technical report. These constraints consist of hard constraints C (i) to C (iv) and soft constraints C(v) to C(viii) as follows.

C (i) All lectures of a course must be scheduled, and they must be assigned to distinct periods. C(ii).Two lectures cannot take place in the same room in the same period.

C(iii): Lectures of courses in the same curriculum or taught by the same teacher must be all scheduled in different periods. C (iv). If the teacher of the course is not available to teach that course at a given period, then no lectures of the course can be scheduled at that period. C(v). For each lecture, the number of students that attend the course must be less or equal than the number of seats of all the rooms that host its lectures. C(vi). The lectures of each course must be spread into the given minimum number of days. C(vii). Lectures belonging to a curriculum should be adjacent to each other (i.e., in consecutive periods). C(viii). All lectures of a course should be given in the same room.

The following section describes how the standard model can represent each of the abovementioned curriculum timetabling constraints.

C(i). This constraint is a combination of two constraints: a domain constraint that specifies the lectures must be allocated to one of the timeslots in time domain and a spread constraint specifies that the difference between the each and every pair of timeslots should not be nil. Timeslot_Constraint((S1, S2, S3,…), (T1, T2, T3, T4…), 100) specifies that all the lectures S1, S2 etcetera of a course should be allocated to one of the timeslots T1, T2, etcetera. Timeslot_Spread ((S1, S2, S3, …), (0), 0, -100) specifies that the difference between the timeslots allocated to lectures S1, S2, S3 etcetera should not be zero. C(ii). Timetableslot_Spread ((S1, S2, S3, …), (0), 0, -100) specifies that the difference between the timetableslots allocated to lectures S1, S2, S3 etcetera should not be zero. C(iii). If S1, S2, S3 etcetera are the lectures in the same curriculum are taught by the same teacher than the constraint Timeslot_Spread ((S1, S2, S3, …), (0), 0, -100) specifies that the difference between the timeslots allocated to lectures S1, S2, S3 etcetera should not be zero, that is, it should be one or more. C (iv). If a teacher teaching the lectures S1, S2, S3 etcetera is not available at time timeslots T1, T2, T3, etcetera, then the constraint Timeslot_Constraint((S1, S2, S3,…), (T1, T2, T3, T4…), -100) specifies that those timeslots should not be allocated to the given lectures.

C (v). If the lectures S1, S2, S3 of a course can be accommodated only by the rooms R1, R2, R3, etcetera that have sufficient seating capacity then the constraint Room_Constraint((S1, S2, S3,…), (R1, R2, R3,…), 90) specifies that those lectures should be preferably allocated to one of the given rooms. C (vi). T If S1, S2, S3 etcetera are the lectures of a course that should be spread into a minimum of say 5 days than the constraint Day_Spread ((S1, S2,), (0, 1,2,3,4), 0, -90) ensures that those two lectures will preferably have a day spread of 5 or more. C(vii).If S1 and S2 are two lectures of a curriculum, than the constraint Timeslot_Spread ((S1, S2,), (1), 0, 90) specifies that the two lectures should preferably be allocated consecutive timeslots (periods).

C (viii). If lectures S1, S2, S3 etcetera belong to one course, than the constraint Room_Spread ((S1, S2, S3, …), (0), 0, 90) specifies that those lectures should preferably be held in the same room.

Standardization of Post-Enrolment Timetabling Constraints

The post enrolment input file contains the following constraints: P (i) An event requires certain features; P (ii) An event cannot be assigned to a particular timeslot and P (iii). An event should be scheduled before another event. The following section describes how each of the abovementioned post-enrolment timetabling constraints can be represented by the standard model. P(i). If the lectures (events) S1, S2, S3 of a course can be accommodated only by the rooms R1, R2, R3, etcetera that have the required features then the constraint Room_Constraint((S1, S2, S3,…), (R1, R2, R3,…), 100) specifies that those lectures should be allocated to one of the given rooms. P(ii). This domain constraint excludes timeslots from being allocated to those events (say lectures). If the lectures S1, S2, S3 etcetera cannot be allocated timeslots T1, T2, T3, etcetera, then the constraint Timeslot_Constraint((S1, S2, S3,…), (T1, T2, T3,…), -100) specifies that those timeslots should not be allocated to the given lectures. P(iii). This is a spread constraint that specifies the order in which events should be scheduled. If S1 and S2 are two lectures (or events) where S2 should be scheduled after S1, than the constraint Timeslot_Spread ((S1, S2,), (0), 1, -100) specifies that the S1 and S2 should not be scheduled at the same time and that S2 should be scheduled after S1. Lewis et al (2007) also state some real world constraints that were not included as part of the Second International Timetabling Competition, but it was recognized that these varying constraints (enumerated X(i) to X(ix), in the following section) as well as many other constraints form a part of real world timetabling problems. The following section shows how these constraints can be represented in the standard model.

X (i) "Inter-site travel times: in some practical cases, a university might be split across a number of campuses, and students and staff may require some commuting-time in order to travel from one site to another. Thus, if two events i and j have common students, but need to take place in different sites, then the constraint 'if event i is scheduled to occur in timeslot x, then event j cannot occur in timeslot x + 1 if this timeslot is on the same day'". If the lectures (events) S1, S2, S3 , etcetera are limited to different sites then the constraint Timeslot_Spread((S1, S2, S3,…), (0,1) 0, -100) specifies that the given lectures should not be scheduled at the same time or one after another. X (ii) "Providing a Lunch-break: many universities will also want to ensure that all staff and students have the opportunity to eat lunch. Thus constraints such as the following might be imposed: 'if a student is attending an event in a 12:00pm timeslot, then he-or-she must not be required to attend and event in a 1:00pm timeslot on the same day, and vice-versa' ". This is a domain constraint requiring the introduction of an event LBx (Lunch Break). LBx is allocated to each and every staff and student. x represents a number for the various lunch breaks for each group of students and staff. If T1 and T2 represent the timeslots 12 noon and 1.00 pm respectively, the constraint Timeslot_Constraint((LB1, LB2, LB3, …), (T1, T2), 100) specifies the various lunch breaks should be either at 12 noon or at 1 pm. X (iii) Relative Timing of Events: universities may also wish to impose other types of constraint on their timetabling problem such as "events i and j must be assigned to the same/different timeslots", "events i and j must take place on the different days", "there must be at a least one day gap between events i and j", and so on. These are spread constraints similar to P(iii). X (iv) "Events without Rooms: in certain constraints such as those describe for some events may not actually require a room, because they may take place outdoors, involve trips to off-site locations, and so on". This is a room domain constraint where dummy rooms are created for those events do not require a physically existing room. X (v) "Room availability: In some cases, certain rooms might not be available in certain timeslots. This could be caused by, say, the room being used by another faculty, or because the key-holder of the room might not be present at certain times during the week". This is a timetableslot domain constraint. The constraint Timetable_slot_Constraint((S1, S2, S3,…), (TB1, TB2, TB3,…), 100) specifies the given lectures(events) Li, S2 etcetera can be allocated the rooms at the given timeslots, the rooms in combination with the timeslots, constituting the timetable slots. X (vi) "Room Hierarchies: in many institutions, a large room may have a number of movable partitions within it, so that the room can be effectively broken up into a number of smaller classrooms". This means that in one timeslot, the resource might be used to house a very large event, while in the next timeslot a number of smaller events might all be scheduled into this same resource. This is a room domain constraint whereby room virtualization is used to combine small room partitions into a larger room and a large room is sub-divided into smaller rooms. Each physical room is allocated a total capacity and a current capacity (Chand 2004). Let us say that Room X (total capacity 150) is composed of three smaller partitions: Room Xa, Xb and Xc each of total capacity 50. If Room X is used for a single event to house, say, 120 students, then its current capacity is updated to zero, that is no more events can be scheduled in that room at that timeslot. However, if Room X is used to house a small class of say 40 students in one of its partitions (Room Xa), the current capacity of Room X is updated to 100 (from the previous 150), meaning that up to another 100 students can be accommodated in both the other partitions as a single room, or up to 50 each in the Rooms Xb and Xc at that timeslot. X (vii) "Filling Rooms: In some cases, the university may have a policy where small events are discouraged from being put into overly large lecture theatres etc". This is a domain constraint. Larger classes have preference over smaller classes for larger rooms, hence scheduling small events on larger rooms are automatically discouraged. As well as all of these features, there are also an abundance of different constraints relating to the usability and "friendliness" of a timetable. X (viii) "Free days: in some institutions, it may be considered desirable to allow students and/or staff to have one day a week free from lectures in order to allow time for research etc". X (ix) "Lecturer Preferences: There may also be a number of individual requirements from lecturers about the allocation of their teaching hours. Some lecturers, for example, may prefer to do all of their teaching in a single day; others may prefer to have their hours equally distributed throughout the week". This is a spread constraint. If the lectures S1, S2, S3 etcetera are conducted by one lecturer than the constraint Day_Spread((S1, S2, S3,…), (0) 0, 100) specifies that the lectures should be scheduled on the same day. Alternatively, Day_Spread((S1, S2, S3,…), 1,2,3,4) 0, 100) specifies that lectures should be spread over more than one day up to five days.

Summary and Future Work

It has been demonstrated how all the specified constraints of the various tracks of the Second International Timetabling Competition can be represented by the proposed standard format. The implication is that all the constraints of the various tracks are instances of a generic university timetabling problem and hence a generic function could possibly be used to determine the hardness of the timetabling problems and acceptability of the solutions. University timetabling problems consist of the same domain and spread constraints. Difference may exist in the number of domain and spread constraints, and the ratio between domain and spread constraints amongst various instances of university timetabling problems, but the problem still remains the same in terms of the nature of constraints.

References

1. Burke E. K, Eckersley A. J , McCollum B , Petrovic S , Qu R, (2004) Similarity Measures For Exam Timetabling Problems, Proceedings of the 5th International Conference of the Practice and Theory of Automated Timetabling.

2. Kostuch P, Socha K, (2004) Hardness Prediction for the University Course Timetabling Problem,

3. Lewis R, Paechter B, McCollum B, (2007), Post Enrolment based Course Timetabling: A Description of the Problem Model used for Track Two of the Second International Timetabling Competition
http://www.cs.qub.ac.uk/itc2007/postenrolcourse/course_post_index_files/problemmodel.htm

4. Gaspero L D, McCollum B, Schaerf A, (2007), The Second International Timetabling Competition (ITC-2007): Curriculum-based Course Timetabling,
http://www.cs.qub.ac.uk/itc2007/postenrolcourse/course_post_index_files/problemmodel.htm

5. McCollum B, McMullan P, Burke E K, Parkes A J, Qu R (2007), The Second International Timetabling Competition: Examination Timetabling Track,
http://www.cs.qub.ac.uk/itc2007/examtrack/exam_track_index_files/examproblemmodel.htm

6. Chand A. (2004), A Generic Constraint Based Model for Representing Complete University Timetabling Data, Proceedings of the 5th International Conference on the Practice and Theory of Automated Timetabling Pittsburgh

7. Chand A (2005) A Constraint Directed Reasoning System for University Timetabling MSc Thesis In partial fulfillment of the requirements for the degree of Master of Science
School of Computing, Information and Mathematical Sciences, Faculty of Science and Technology, the University of the South Pacific.