New Constraints and Features for the University Course Timetabling Problem*

S. Applebee, M. Aschinger, A. Bucur[†] H. Edmonds, P. Hungerländer and K. Maier

12th December 2016

Abstract

The university course timetabling problem deals with the task of scheduling lectures of a set of university courses into a given number of rooms and time periods, taking into account various hard and soft constraints. The goal of the International Timetabling Competitions ITC2002 and ITC2007 was to establish models for comparison that cover the most frequently found use cases.

Our model, motivated by a project with University College London (UCL), builds on the standard model from track 3 of ITC2007. Compared to the standard model from the literature, we cover several new constraints and extra features. For example, we expand the ITC2007 framework to generate a timetable for several weeks of the term instead of only one and introduce the corresponding timetable regularity metric, which measures the consistency of time and room assignments for a course throughout the term.

We suggest an Integer Linear Programming approach for solving this expanded timetabling problem and introduce a corresponding new benchmark library. Finally we conduct computational experiments and discuss the results obtained with respect to solution quality and practical suitability for UCL.

Keywords: University course timetabling; integer linear programming; University College London; timetable regularity; activity durations.

1 Introduction

The 2nd annual Timetabling Competition (ITC-2007) established a standardized framework for the timetabling problem in terms of problem formulation and test instances [5]. With knowledge of this, University College London (UCL) and Satalia (NPComplete Ltd) funded the research presented in this paper to investigate the value of optimization-based timetabling and to provide implementable results. The UCL timetabling problem

*This research was supported by the third party project *Satalia*, funded by Satalia (NPComplete Ltd) and conducted at the Alpen-Adria-Universität Klagenfurt.

[†]Corresponding Author: Department of Mathematics, Alpen-Adria Universität Klagenfurt, Austria, pabucur@edu.aau.at.

was modelled as closely to true complexity as possible, using actual data from 2015 and new features and constraints not previously considered in the literature.

Reviewing the relevant literature we notice that while over the years local search techniques have dominated the field of timetabling research [4], recently new approaches based on SAT [1], Constraint Programming [6], Mathematical Programming [2] and Metaheuristics [3] have successfully entered the field. Our approach is based on the Mathematical Programming methodology, as we suggest an Integer Linear Programming (ILP) algorithm to solve the optimization problem at hand.

This paper is structured as follows. In Section 2, we describe the curriculum-based timetabling problem as it was proposed in the ITC-2007 competition and extend it to include some of the extra constraints and features required by UCL. In Section 3, we introduce our ILP approach and the way it encodes the different constraints. In Section 4 we introduce several relevant metrics and we present and discuss the results of our computational experiments. Section 5 concludes the paper.

2 Problem Formulation: UCL Extended Framework

The challenge of the curriculum-based course timetabling problem (CB-CTT) is to schedule lectures belonging to a set of courses $C = \{c_1, c_2, \ldots, c_v\}$ to k periods $P = \{p_1, p_2, \ldots, p_k\}$ and m rooms $R = \{r_1, r_2, \ldots, r_m\}$, accounting at the same time for certain hard and soft constraints. In the CB-CTT the timetable is generated based upon a set of s university curricula $I = \{i_1, i_2, \ldots, i_s\}$, to which the courses belong.

Next let us introduce assignment vectors q whose entries are set to 1, if a course-period-room combination is contained in timetable. Otherwise the entries are set to 0. Now a feasible solution of the problem is a binary vector q that satisfies all hard constraints. Finally the task of the CB-CTT is to find a vector q^* such that $f(q^*) \leqslant f(q)$ for all $q \in \tilde{q}$, where $f(\cdot)$ is an evaluation function summing up all violations of the soft constraints and $\tilde{q}$ denotes the set of all feasible assignment vectors. The exact problem formulation of the ITC-2007 framework and further details can be found in [5].

The UCL timetabling problem presents a wide range of challenges, since its features and additional constraints substantially exceed the ones from the ITC-2007 framework. Many of those features and additional constraints are omitted in this short paper due to space limitations and will be provided in a forthcoming paper. We include though the extensions that are most interesting from an academic viewpoint:

1. Our courses consist of activities with different durations, which relaxes the indistinguishability assumption of lectures from the literature. Therefore we define the set of all activities $A = \{a_1, a_2, \ldots, a_n\}$ with corresponding durations of activities $d_a,\ a \in A$.

2. The UCL framework aims to generate feasible timetables for a set W of 10 consecutive weeks in a manner that guarantees the highest possible timetable regularity. This means that whenever possible, activities should be scheduled in the same period and room over the different weeks.

3. Some activities within the same curriculum can be scheduled in parallel, e.g. if students need to attend only one of the practical lessons offered.

4. The activities have a specific predefined type, which must match the room type of the assigned room.

3 Integer Linear Programming Models

The Integer Linear Programming (ILP) solver presented in this paper is based on the approach suggested by Lach and Lübbecke [2]. The problem is split into two stages. In the first stage, each activity is assigned to an appropriate set of consecutive time periods. The assignment of activities to rooms is done in the second stage. Due to space limitation, we state the mathematical formulation only for the second stage and for the first stage we solely mention the most important newly developed constraints.

While our solver is optimized for the UCL framework, we have also tested it on the original ITC-2007 benchmark set. Preliminary results are very encouraging and hence we plan to provide a detailed analysis showing the competitiveness of our solver on the standard benchmarking sets from the literature in a forthcoming paper.

First Stage: In the first stage each activity has to be scheduled in a consecutive set of time periods. The function $D(p)$ gives the day of period p. Now if activity a is scheduled at period p, then the binary variable x_{ap} is set to 1. Otherwise we have $x_{ap} = 0$. To ensure that an activity is assigned to consecutive time periods, we also need binary variables s_{ap}, which are set to 1, if activity a starts at period p. Otherwise we have $s_{ap} = 0$. Note that variable s_{ap} is only introduced, if there are at least $d_a - 1$ consecutive time periods available after period p on the same day:

$$x_{ap} - \sum_{\substack{t=p-d_a+1 \\ s_{at} \text{ exists}}}^{p} s_{at} = 0, \quad a \in A,\ p \in P, \tag{1}$$

$$\sum_{s_{ap} \text{ exists}} s_{ap} = 1, \quad a \in A. \tag{2}$$

Equalities (1) ensure that each activity is assigned to a set of consecutive time periods that all belong to the same day. Equalities (2) guarantee that each activity has exactly one start time period.

One of our main goals for the UCL Timetabling Problem is to minimize the total number of rooms required. While this goal is clearly part of the objective function of the Second Stage, we also need to consider it during the First Stage. We propose the following constraints in order to restrict the total number of activities scheduled per time period:

$$\sum_{a \in A} x_{ap} \leq M, \qquad p \in P, \tag{3}$$

where M is an integer variable that is multiplied by a penalty term p_M in the objective function of the First Stage. Without inequalities (3) arbitrarily many activities could be assigned to the same time period during the First Stage, which could leave us with no possibility to minimize the number of rooms required in the Second Stage.

Second Stage: After solving the First Stage, in the Second Stage we determine feasible rooms for the activities, where we aim to minimize the following objectives:

(a) The number of students, which have no seat during an activity.

(b) The number of empty seats in a room during an activity.

(c) The total number of rooms.

In order to build an ILP model for the Second Stage, we introduce binary variables u_r, y_{ar} and z_{arp} with the following interpretations:

- $u_r = 1$, if at least one activity is scheduled in room r. Otherwise $u_r = 0$.
- $y_{ar} = 1$, if activity a is scheduled in room r. Otherwise $y_{ar} = 0$.
- $z_{arp} = 1$, if activity a is scheduled in room r at period p. Otherwise $z_{arp} = 0$.

Note that the variables y_{ar} and z_{arp} are only introduced, if it is feasible to schedule activity a in room r at period p, i.e. if the activity type matches with the room type, if the activity is assigned to period p in the First Stage and if the room is available at period p. Accordingly we define $P(a)$, $P(r)$ and $P(a,r)$ as the sets of available time periods for activity a, for room r and for their combination respectively. Analogously we specify $A(r)$ and $A(p,r)$ as the sets of feasible activities for room r at period p and $R(a,p)$ as the set of feasible rooms for activity a at period p.

For each feasible activity-room combination we introduce a penalty parameter p_{ar} that gives the absolute value of the difference between the available seats in room r and the number of students registered for activity a. We also introduce the penalty parameter p_r giving the costs for using room r. Now we can state our ILP model:

$$\min \sum_{a\in A,\ r\in R(a,p)} p_{ar}y_{ar} + \sum_{r\in R} p_r u_r \tag{4a}$$

$$\text{s.t.} \sum_{r\in R(a)} z_{arp} = 1, \quad a\in A,\ p\in P(a), \tag{4b}$$

$$d_a y_{ar} - \sum_{p\in P(a,r)} z_{arp} = 0, \quad r\in R,\ a\in A(r), \tag{4c}$$

$$\sum_{a\in A(p,r)} z_{arp} - u_r \leq 0, \quad r\in R,\ p\in P(r), \tag{4d}$$

$$u_r\in\{0,1\},\ y_{ar}\in\{0,1\},\ z_{arp}\in\{0,1\}, \quad r\in R,\ a\in A(r),\ p\in P(a,r). \tag{4e}$$

Equalities (4b) guarantee that exactly one room is assigned to an activity at each time period. Equalities (4c) ensure that the same room is assigned to all time periods of an activity. Constraints (4d) guarantee that at most one activity is assigned to room r at each time period and also ensure $u_r = 1$, if at least one activity is scheduled in r.

4 Metrics and Results

In order to evaluate the quality of a timetable and its usefulness in practice, it is necessary to introduce a wide range of metrics. In this paper we present a short selection of metrics that were deemed most important by UCL.
Space utilization: The UK's most important metric for determining how well universities use their facilities is space utilization s_u, which is defined as the sum of room frequency and average room occupancy. For computing the room frequency r_f we divide the number of time periods occupied by the number of time periods that are available in rooms, in which at least one activity is scheduled.

Next let us define the room occupancy $o_{a,r}$ of an activity a that is scheduled in room r: $o_{a,r} = \min\left(1, s_a/c_r\right),\ r \in R,\ a \in T(r)$, where s_a denotes the number of students registered for activity a, c_r gives the capacity of room r and the set $T(r)$ contains all activities scheduled in room r. Now the average room occupancy $\bar{r}_o$ is simply the mean of all room occupancies of the considered timetable.

Note that the objective function of the Second Stage of our ILP approach is tailored to minimize both space utilization and the number of students without seats.

Timetable regularity: Timetable regularity measures the consistency of time and room assignments of a timetable throughout the term, assuming that each week of the term has slightly different activities. We count the different start times $s(a,w)$ and room assignments $r(a,w)$ of an activity a in week $w \in W$ via the following function:

$$g(w_1,w_2,\overline{a}) = \begin{cases} 2, & \text{if } s(\overline{a},w_1) \neq s(\overline{a},w_2) \text{ and } r(\overline{a},w_1) \neq r(\overline{a},w_2), \\ 1, & \text{if either } s(\overline{a},w_1) \neq s(\overline{a},w_2) \text{ or } r(\overline{a},w_1) \neq r(\overline{a},w_2), \\ 0, & \text{otherwise,} \end{cases}$$

with $w_1, w_2 \in W$, $i \in I$ and $\overline{a} \in A(i,w_1,w_2)$ is an activity that has to be scheduled both in w_1 and w_2 in curriculum i. There are b different combinations of activities, pairs of weeks and curricula and the total number of students is given by $h_t = \sum_{i \in I} h_i$, where $h_i,\ i \in I$, is the number of students registered in curriculum i. Now the timetable regularity TR can be defined as:

$$\text{TR} = 1 - \left(\sum_{i \in I} \left(\sum_{\substack{w_1,w_2 \in W \\ w_1 \neq w_2}} \left(\sum_{\overline{a} \in A(i,w_1,w_2)} g(w_1,w_2,\overline{a}) \right) \right) \cdot h_i \right) / \left(2 \cdot b \cdot h_t\right).$$

We try to maximize TR by first including activities that have to be scheduled in most weeks of the term in a base week that is solved with the two stage approach described in the previous section. Afterwards we solve an ILP for each particular week, where we maximize similarity to the base week by adding respective soft constraints.

Computational Experiments: Finally we present the results obtained by using our ILP approach on a selection of the original set of UCL curricula, available at `http://tinyurl.com/timetabling-lib`. All experiments were performed on a Linux 64-bit machine equipped with 4 × Intel(R) Xeon(R) CPU e5-2630 v3@2.40GHz and 16 GB RAM. We use Gurobi 6.5.1 as our ILP-solver.

Our benchmark set consists of around 250 activities per week with an average length of ≈ 3.5 time periods. Each week consists of 5 days with 18 time periods (a 30 minutes) per day. In each week we use around 20 of the available 279 rooms.

We obtained timetables for the whole term within 190 seconds computing time. The corresponding metrics are:

$$\text{(a) } r_f = 0.49, \quad \text{(b) } \bar{r}_o = 0.78, \quad \text{(c) } s_u = 1.27, \quad \text{(d) } TR = 0.8723.$$

The very high timetable regularity of $87,23\%$ is very important for the 35615 UCL students. With our timetables determined they do not have to adapt to frequent weekly timetable changes. Furthermore the high average occupancy metric shows that on average, used rooms are more than $\frac{3}{4}$ full, which ensures an efficient facility usage. Finally the room frequency metric indicates that rooms, which are used at least once, are occupied almost 50% of the total available time periods.

5 Conclusion

In this paper we presented a solution to the curriculum-based timetabling problem of a real-world institution, the University College London, whose requirements and specifications considerably exceed those of the ITC-2007 problem formulation. Due to space restrictions, we selected only the most significant new problem features and the most interesting metrics for this paper. An extended version of this publication will include the solution to the complete problem with 1000 activities per week and 279 rooms, as well as the full set of modeling features, requirements and metrics.

References

[1] R. Asín Achá and R. Nieuwenhuis. Curriculum-based course timetabling with sat and maxsat. *Annals of Operations Research*, 218(1):71–91, 2014.

[2] G. Lach and M. E. Lübbecke. Curriculum based course timetabling: new solutions to udine benchmark instances. *Annals of Operations Research*, 194(1):255–272, 2012.

[3] R. Lewis. A survey of metaheuristic-based techniques for university timetabling problems. *OR Spectrum*, 30(1):167–190, 2008.

[4] Z. Lü and J.-K. Hao. Adaptive tabu search for course timetabling. *European Journal of Operational Research*, 200(1):235 – 244, 2010.

[5] B. McCollum, A. Schaerf, B. Paechter, P. McMullan, R. Lewis, A. J. Parkes, L. D. Gaspero, R. Qu, and E. K. Burke. Setting the research agenda in automated timetabling: The second international timetabling competition. *INFORMS Journal on Computing*, 22(1):120–130, 2010.

[6] L. Zhang and S. Lau. Constructing university timetable using constraint satisfaction programming approach. In *CIMCA-IAWTIC'06 - Volume 02*, pages 55–60. IEEE Computer Society, 2005.