Daily Pattern Formulation and Valid Inequalities for the Curriculum-based Course Timetabling Problem

Niels-Christian Fink Bagger · Guy Desaulniers · Jacques Desrosiers

Abstract In 2007 the Second International Timetabling Competition was held containing the Curriculum-based Course Timetabling Problem. Since the competition the problem has been widely studied to solve it. A few studies have been dedicated to obtain lower bounds. We will follow this direction and focus on the lower bounds by only scheduling the courses into periods and ignoring the room assignments. The method we use is to formulate a mixed integer program based on a pattern enumeration for each course and each day. The pattern formulation is able to improve four lower bounds from the benchmark data and closing one of them.

Keywords Course Timetabling · Pattern Enumeration · Preprocessing · Valid Inequalities

1 Introduction

In this work we will focus on the Curriculum-based Course Timetabling Problem (CB-CTP) from Track 3 in the second International Timetabling Competition in 2007 (ITC2007) as described by Di Gaspero et al (2007). In this problem we need to generate a weekly schedule for a set of courses $\mathcal{C}$. A week

N.C. Bagger
mORetime Research Group, Management Science, Department of Management Engineering, Technical University of Denmark, Produktionstorvet, Building 426, 2800 Kgs. Lyngby, Denmark
Tel.: +45 45 25 45 42
E-mail: nbag@dtu.dk

G. Desaulniers
GERAD – HEC Montréal 3000, Côte-Sainte-Catherine, Montréal QC H3T 2A7, Canada
École Polytechnique de Montréal, Montréal, Québec, Canada H3C 3A7

J. Desrosiers
GERAD – HEC Montréal 3000, Côte-Sainte-Catherine, Montréal QC H3T 2A7, Canada
École des Hautes Études Commerciales, Montréal, Québec, Canada H3T 2A7

is divided into a number of days, e.g. 5, and each day is divided into a number of time slots, e.g. 6. We denote the set of days $\mathcal{D}$ and the set of time slots $\mathcal{T}$. A day and time slot combination is referred to as a period. So the task is to determine the periods and rooms that each lecture of each course is to take place in while ensuring some constraints are fulfilled. Each course has a predetermined number of lectures and all lectures must be scheduled and they must be scheduled in different periods. A set of lecturers is also defined and each course has a lecturer assigned. Courses taught by the same lecturer are not allowed to have any lectures scheduled at the same time. The courses are divided into curricula $\mathcal{Q}$, hence the name of the problem. For a curriculum $q \in \mathcal{Q}$ all the courses $\mathcal{C}_q$ are not allowed to have lectures scheduled at the same time. For each room it is not allowed to schedule more than one lecture in any period. Any feasible timetable must fulfil the before mentioned requirements.

To measure the quality of a solution a weighted some of some *soft constraints* is computed. For each course $c \in \mathcal{C}$, a minimum number of days $D_c^{\min}$ that the lectures should be spread across is specified. Every day below this number is penalised by five points. For each curriculum, the schedule should be as compact as possible. This is done by considering *isolated lectures*. A lecture is isolated if it is scheduled in period $(d, t) \in \mathcal{D} \times \mathcal{T}$ but no lectures are scheduled in an adjacent period. For each isolated lecture a penalty of two is added to the objective value. Two periods (d_1, t_1) and (d_2, t_2) are considered adjacent if they are on the same day, i.e. $d_1 = d_2$, and if the time slots are contiguous, i.e. $t_1 = t_2 + 1 \vee t_1 = t_2 - 1$. There are also room related soft constraints. However, we will only be looking at the time aspect, i.e. ignoring the rooms, to obtain lower bounds. Lower bounds for this problem have been studied before, see e.g. Hao and Benlic (2011) and Cacchiani et al (2013). Hao and Benlic (2011) generate lower bounds from a model by Lach and Lübbecke (2012). The model only schedules the courses into time periods, but incorporates some of the room-related penalties. A tabu search based heuristic selects a decomposition of the courses into subsets, where each subset is solved independently. A lower bound is then derived by summing up the lower bounds obtained for the individual subproblems. Cacchiani et al (2013) divide the problem into two parts; one that focuses on the period related soft constraints and one that focuses on the room related soft constraints. The sum of the lower bounds of these two problems is then a lower bound of the overall problem. In our work we will also focus on the time period assignments in the model.

2 Pattern Formulation

As it was mentioned that our model is based on a pattern enumeration scheme we need to define what a pattern is. A pattern describes the set of time slots that lectures are scheduled in, e.g. if a pattern is the set $\{0, 3\}$ then it means that if a course is assigned this pattern for some day then lectures are scheduled in time slots 0 and 3 for that course on that day. For the model we enumerate all the possible patterns. As an example consider an instance where the number of time slots for each day is 4. All the patterns are illustrated in Table 1.

Table 1 The 16 possible patterns when $|T| = 4$. The first column is the time slot and the remaining columns are the patterns. An "X" in a cell means that the pattern contains the corresponding time slot. The last row (L_k) counts how many lectures each pattern contains

t	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1		×				×	×	×				×	×	×		×
2			×			×			×	×		×	×		×	×
3				×			×		×		×	×		×	×	×
4					×			×		×	×		×	×	×	×
L_k	0	1	1	1	1	2	2	2	2	2	2	3	3	3	3	4

There is an exponential number of patterns, but as the number of time slots for a day in the data sets from ITC2007 is usually 5 or 6 then this gives us around 32 and 64 different patterns, so it is manageable. All the patterns constitute the set denoted $\mathcal{K}$. Then for each course $c \in \mathcal{C}$ and day $d \in \mathcal{D}$ we create the set $\mathcal{K}_{c,d}$ of feasible patterns for that course and day. Some courses have a set of specific periods where it is not allowed to schedule lectures in. So the patterns from $\mathcal{K}$ are added to the set $\mathcal{K}_{c,d}$ only if assigning that pattern to course $c \in \mathcal{C}$ on day $d \in \mathcal{D}$ does not violate this requirement. Furthermore each course $c \in \mathcal{C}$ has a specified number of lectures to be scheduled L_c. If a pattern contains more lectures than this number then this pattern is not added to any of the sets $\{\mathcal{K}_{c,d}\}_{d \in \mathcal{D}}$.

For the model let the binary variable $\lambda^k_{c,d} \in \mathbb{B}$ take value one if course $c \in \mathcal{C}$ has been assigned pattern $k \in \mathcal{K}_{c,d}$ for day $d \in \mathcal{D}$. Let $w_c \in \mathbb{N}_0$ be a variable for the number of days below $D^{\min}_c$ that course $c \in \mathcal{C}$ has lectures scheduled. The binary variable $s_{q,d,t}$ is set to one if curriculum $q \in \mathcal{Q}$ has an isolated lecture on day $d \in \mathcal{D}$ in time slot $t \in \mathcal{T}$. Let a^k_t be one if pattern $k \in \mathcal{K}$ contains a lecture in time slot $t \in \mathcal{T}$. Let L_k be the number of lectures contained in pattern $k \in \mathcal{K}$, R be the number of rooms available and let $\overline{a}^k_t$ be defined as follows:

$$\overline{a}^k_t = \begin{cases} 1 & \text{if pattern } k \in \mathcal{K} \text{ has a lecture assigned in time slot } t \in \mathcal{T} \\ & \quad \text{and no lecture adjacent to } t \\ -1 & \text{if pattern } k \in \mathcal{K} \text{ does not have a lecture assigned in} \\ & \quad \text{time slot } t \in \mathcal{T}\text{, but at least one adjacent to } t \\ 0 & \text{otherwise} \end{cases}$$

The pattern formulation is illustrated in Model 1.

In this model the constraints (1b) ensure that all lectures are scheduled. Constraints (1c) ensure that exactly one pattern is selected for each course $c \in \mathcal{C}$ and day $d \in \mathcal{D}$. We do not consider the rooms directly in this model but it is still ensured that each room is at most occupied by one lecture for each

$$\begin{aligned}
\min \quad & \sum_{q\in\mathcal{Q},d\in\mathcal{D},t\in\mathcal{T}} 2\cdot s_{q,d,t} + \sum_{c\in\mathcal{C}} 5\cdot w_c && && \text{(1a)}\\
\text{s.t.} \quad & \sum_{d\in\mathcal{D},k\in\mathcal{K}_{c,d}} L_k\cdot\lambda^k_{c,d} = L_c && \forall c\in\mathcal{C} && \text{(1b)}\\
& \sum_{k\in\mathcal{K}_{c,d}} \lambda^k_{c,d} = 1 && \forall c\in\mathcal{C}, d\in\mathcal{D} && \text{(1c)}\\
& \sum_{c\in\mathcal{C},k\in\mathcal{K}_{c,d}} a^k_t\cdot\lambda^k_{c,d} \leq R && \forall d\in\mathcal{D}, t\in\mathcal{T} && \text{(1d)}\\
& \sum_{c\in\mathcal{C}_l,k\in\mathcal{K}_{c,d}} a^k_t\cdot\lambda^k_{c,d} \leq 1 && \forall l\in\mathcal{L}, d\in\mathcal{D}, t\in\mathcal{T} && \text{(1e)}\\
& \sum_{c\in\mathcal{C}_q,k\in\mathcal{K}_{c,d}} a^k_t\cdot\lambda^k_{c,d} \leq 1 && \forall q\in\mathcal{Q}, d\in\mathcal{D}, t\in\mathcal{T} && \text{(1f)}\\
& D^{\min}_c - \sum_{d\in\mathcal{D},k\in\mathcal{K}_{c,d}:L_k\geq 1} \lambda^k_{c,d} \leq w_c && \forall c\in\mathcal{C} && \text{(1g)}\\
& \sum_{c\in\mathcal{C}_q,k\in\mathcal{K}_{c,d}} \overline{a}^k_t\cdot\lambda^k_{c,d} \leq s_{q,d,t} && \forall q\in\mathcal{Q}, d\in\mathcal{D}, t\in\mathcal{T} && \text{(1h)}\\
& \lambda^k_{c,d}\in\mathbb{B} && \forall c\in\mathcal{C}, d\in\mathcal{D}, k\in\mathcal{K}_{c,d} && \text{(1i)}\\
& w_c\in\mathbb{N}_0 && \forall c\in\mathcal{C} && \text{(1j)}\\
& s_{q,d,t}\in\mathbb{B} && \forall q\in\mathcal{Q}, d\in\mathcal{D}, t\in\mathcal{T} && \text{(1k)}
\end{aligned}$$

Model 1: The pattern formulation for the curriculum-based timetabling problem

period in constraints (1d). These constraints are sufficient due to the fact that it is allowed to schedule any course into any room. Constraints (1e) and (1f) ensure that no conflicting lectures are scheduled in the same period, i.e. courses taught by the same lecturer or belonging to the same curriculum. In (1g) and (1h) the values of the variables w_c and $s_{q,d,t}$ are calculated respectively. (1i), (1j) and (1k) are the variable domains.

The main benefit of the pattern formulation is the possibility to preprocess the model and add valid inequalities. In the preprocessing phase the focus is to reduce the set of patterns $\mathcal{K}_{c,d}$ for each course $c \in \mathcal{C}$ and day $d \in \mathcal{D}$. It can also be shown that some of the other variables in the model can be substituted by the pattern variables. After this, some valid inequalities are derived by constructing a graph that contains a node for each pattern variable. This is denoted the *pattern conflict graph*. Two nodes are connected by an edge if both of the corresponding patterns cannot be selected in a feasible solution. Cliques in this graph are then enumerated and for each clique a constraint is added, which takes the sum of the corresponding patterns and set them to be less than or equal to one. The clique graph can also be used to add other valid inequalities which are focusing on the isolated lectures. All the preprocessing, how to identify edges in the pattern conflict graph and the added valid inequalities are too extensive to include in this extended abstract, but will follow in an upcoming article.

3 Results

To test the pattern formulation we used the MIP solver from Gurobi Optimization, Inc. (2015) version 6.5.0 and the 21 data sets from the ITC2007 competition. The default settings were used for most of the parameters except the parameters *Threads* and *Presolve*. Threads was set to 1 since it was only allowed to use a single core for the competition. Presolve was set to 2, the most aggressive setting, to allow the solver to reduce the model even further than we did. The computer used for the tests was a 3.40GHz Intel® Core™ i5-3570K CPU with 16GB memory running Windows 10 Pro 64-bit. In the competition a benchmarking tool was provided to report the amount of time the algorithms were allowed to run on the computers of the contestants. On our computer this was given as $208s$. We have then tested the model with a time limit of $40 \times 208s$ to compare with results from the literature; Lach and Lübbecke (2012), Burke et al (2010), Hao and Benlic (2011) and Cacchiani et al (2013). We also ran the model with a time limit of $100 \times 208s$ to see if we could improve the results further. Table 2 compares the result with those reported in the literature for the first fourteen data sets given a time limit of $40 \times 208s$. The reason for not comparing for the last seven data sets is that these data sets were not made publicly available until after some of the articles were written.

In Table 2 it can be seen that the pattern formulation obtains a lower bound which is at least as good as the lower bound obtained by the other approaches for most of the data sets, except one. The reason that the lower bound for the first data set is worse than that of the other approaches is that the cost of the optimal solution is from the room-related soft constraints which we did not consider in our model. The pattern formulation shows competitiveness in terms of obtaining lower bounds and finds a better bound in most cases.

The result of running the pattern formulation with a time limit of $100 \times 208s$ is reported in Table 3. There it can be seen that the pattern formulation is able to improve four of the currently best known bounds and proving that the best know solution value is optimal.

References

Burke EK, Mareček J, Parkes AJ, Rudová H (2010) Decomposition, reformulation, and diving in university course timetabling. Computers & Operations Research 37(3):582–597

Cacchiani V, Caprara A, Roberti R, Toth P (2013) A new lower bound for curriculum-based course timetabling. Computers & Operations Research 40(10):2466 – 2477

Di Gaspero L, McCollum B, Schaerf A (2007) The second international timetabling competition (itc-2007): Curriculum-based course timetabling (track 3). Tech. rep., School of Electronics, Electrical Engineering and Computer Science, Queens University SARC Building, Belfast, United Kingdom

Gurobi Optimization, Inc (2015) Gurobi optimizer reference manual. URL http://www.gurobi.com

Hao JK, Benlic U (2011) Lower bounds for the itc-2007 curriculum-based course timetabling problem. European Journal of Operational Research 212(3):464 – 472

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

Table 2 The lower bounds obtained when given 40 CPU units. The first column is the data sets. The second column is the currently best known solution values. A lower bound (lb) and the gap (%gap) between the lower bound and the best known upper bound is reported for each approach; LL (Lach and Lübbecke, 2012), B (Burke et al, 2010), HB (Hao and Benlic, 2011), C (Cacchiani et al, 2013) and P (the pattern formulation). The numbers reported in bold font denotes when the corresponding approach obtained a lower bound at least as good as the others. An underlined number denotes when the corresponding approach obtains a lower bound better than all the others. The bottom two rows counts the number of times that the approach obtained a lower bound at least as good as all the others (bold font) and the number of times that the approach obtained a lower bound better than all the others (bold font and underlined)

		LL		B		HB		C		P	
comp*	ub	lb	%gap	lb	%gap	lb	%gap	lb	%gap	lb	%gap
01	5	4	20.0	**5**	0.0	4	20.0	**5**	0.0	0	100.0
02	24	11	54.2	1	95.8	12	50.0	16	33.3	<u>**20**</u>	16.7
03	64	25	60.9	33	48.4	36	43.8	**52**	18.8	**52**	18.8
04	35	28	20.0	**35**	0.0	**35**	0.0	**35**	0.0	**35**	0.0
05	284	108	62.0	114	59.9	80	71.8	166	41.5	<u>**191**</u>	32.7
06	27	10	63.0	16	40.7	16	40.7	11	59.3	<u>**24**</u>	11.1
07	6	**6**	0.0	**6**	0.0	**6**	0.0	**6**	0.0	**6**	0.0
08	37	**37**	0.0	**37**	0.0	**37**	0.0	**37**	0.0	**37**	0.0
09	96	46	52.1	66	31.3	67	30.2	92	4.2	<u>**96**</u>	0.0
10	4	**4**	0.0	**4**	0.0	**4**	0.0	2	50.0	**4**	0.0
11	0	**0**	0.0	**0**	0.0	**0**	0.0	**0**	0.0	**0**	0.0
12	298	53	82.2	95	68.1	84	71.8	100	66.4	<u>**165**</u>	44.6
13	59	41	30.5	54	8.5	55	6.8	57	3.4	<u>**59**</u>	0.0
14	51	46	9.8	42	17.6	43	15.7	48	5.9	<u>**51**</u>	0.0
Best		**4**		**6**		**5**		**6**		**13**	
		<u>**0**</u>		<u>**0**</u>		<u>**0**</u>		<u>**0**</u>		<u>**7**</u>	

Table 3 The lower bounds obtained when given 100 CPU units. The first column is the data sets. The second column is the currently best known solution values. A lower bound (lb) and the gap (%gap) between the lower bound and the best known upper bound is reported for the best currently known (Best) and for the pattern formulation(P). The numbers reported in bold font denote when the pattern formulation obtained a lower bound at least as good as the best known ones. An underlined number denotes when the pattern formulation obtained a lower bound that improves the currently best known ones. The bottom two rows count the number of times that the pattern formulation obtained a lower bound at least as good as the best known (bold font) and the number of times that the pattern formulation obtained a lower bound which improves the best known ones (bold font and underlined)

		Best		P	
comp*	ub	lb	%gap	lb	%gap
01	5	5	0.0	0	100.0
02	24	16	33.3	**<u>24</u>**	0.0
03	64	52	18.8	**<u>54</u>**	15.6
04	35	35	0.0	**35**	0.0
05	284	211	25.7	210	26.1
06	27	27	0.0	26	3.7
07	6	6	0.0	**6**	0.0
08	37	37	0.0	**37**	0.0
09	96	96	0.0	**96**	0.0
10	4	4	0.0	**4**	0.0
11	0	0	0.0	**0**	0.0
12	298	138	53.7	**<u>175</u>**	41.3
13	59	59	0.0	**59**	0.0
14	51	51	0.0	**51**	0.0
15	62	52	16.1	**<u>54</u>**	12.9
16	18	18	0.0	**18**	0.0
17	56	56	0.0	53	5.4
18	61	61	0.0	52	14.8
19	57	57	0.0	**57**	0.0
20	4	4	0.0	**4**	0.0
21	74	74	0.0	**74**	0.0
Best				**16**	
				<u>4</u>	