2015 International Conference on Computational Science and Computational Intelligence

Improving Curriculum Timetabling Models Using Clustering

Thomas Philip Runarsson
School of Engineering and Natural Science
University of Iceland, Iceland
Email: tpr@hi.is

***Abstract*—This work describes how clustering can aid in the modelling of the curriculum timetabling problem. The practical timetabling problem cannot be solved to proven optimality in any reasonable time. A clustering technique is used to construct additional constraints, that reduce the size of the feasible search space, and improves the quality of the time-tables found within a reasonable computational time. The approach is illustrated using on a real world timetabling problem and a state-of-the-art commercial solver.**

***Keywords*-Clustering; timetable problem; building mathematical models;**

I. Introduction

This work describes the curriculum university timetabling problem at the School of Engineering and Natural Science, University of Iceland. As with any real world timetabling problem there are many issues that cannot be addressed, since the data used is typically unreliable and dynamic. For this reason the timetables are constructed by hand by two or more persons in the start of each semester, with varying results. Here an attempt is made to solve this timetabling problem as a mixed integer programming problem. Clustering is used to discover pseudo curricula. These are curricula defined by a set of courses taken by a large number of students and are not part of the regular curricula. Finding these and enforcing them as constraints reduces the feasible search space significantly and so the mixed integer programming (MIP) model becomes more tractable.

The approaches to the school timetabling problem are perhaps as many as the schools. Each school or educational system has its own set of rules that must be adhered to. The timetabling problem studied here is no exception. In the recent months there have been a number of papers devoted to an overview or survey of this important problem domain [1], [2], [3], [4]. In essence the problem is about assigning classes to timeslots and rooms. Common requirements are that courses should not clash, especially courses for the different curricula offered by the school. The utilization of rooms should be maximized. They should not be overfilled but all should be utilized. The workload of both teachers and students should be balanced. The timetables should be compact for the students but not for the teacher. Various custom preference are then added to these conditions specific for the school and may even vary between semester. The timetabling problem investigated here has all of these elements, but is tackled very differently to that described in the literature. However, the approach presented for reducing the size of the search space, using clustering, may be applied to the other formulations in the literature.

It is anticipated that many conditions will change during the timetabling process in the time leading up to the start of the semester. This is due to the volatility of the data used. Teachers availability, student registrations, and room requirements may all be subject to change. The process is an iterative one and for this reason it is desirable that the timetabling problem deliver quality solutions in a reasonable time.

The paper is organized as follows. In section II a MIP model for the curriculum based university timetabling problem is described. It has similar feature to models seen previously in the literature, but uses a timeslot system specific to the University of Iceland. The system eliminates the need to balance specifically the workload of the students and organizes the lectures in a balanced manner. In section III the clustering method used to discover pseudo curricula is described and the accompanying constraints needed for the MIP model. This is followed by an experimental study on real work timetabling data from the University of Iceland. The paper concludes with a discussion and summary of main findings.

II. Timetabling at the University of Iceland

The courses at the University of Iceland are assigned to pre-defined time-slots which are in total seven. Each time-slot is split up into two continuous blocks on two separate days. One of the blocks will hold two 40 minute lectures with one break and the other three lectures with two breaks. Hence a total of five 40 minute lectures or tutorials can be held in any time slot. Five of these time- slots are before lunch while a further two are immediately after lunch on Monday through to Thursday. The seven time-slots are shown in figure 1. The idea behind this scheme is that students have at least a one day break for any given course. Typically a course will

978-1-4673-9795-7/15 $31.00 © 2015 IEEE
DOI 10.1109/CSCI.2015.94

require five forty minute lectures/tutorials. Two courses or more could be placed within any time-slot, as long as the number of lectures does not total to more than five. Some courses may require more than five lectures, in this case putting these classes in slots six and seven would be best as these time-slots are in the afternoon and could therefore be easily extended. If the first five time slots were extended they would go into the lunch break, which is possible but not desirable. Any course, that does not fit into this scheme, could also be placed in the afternoon within a dummy time-slot labelled the eight time-slot in figure 1. Typically tutorials are held in the afternoon for large courses and will be assigned to time-slot eight[1]. In general it is preferable that these classes be in the first five slots before the lunch break.

When building a timetable the set of courses that define a *curriculum* may not clash, these *curricula clashes* are hard constraints. However, it is quite common that students complete their three year bachelor degree in four years. This means that there will exist curriculum clashes between semesters. Furthermore, during the last two semesters students will typically take elective courses also. These electives and the fact that students take the degree over a longer period, create potential *course clashes*. *Teacher clashes* are also plausible, but can be treated equivalently to a *curriculum clash*. That is, a set of lectures given by a teacher may not clash. In general this is not a problem. Teachers usually teach only one or two classes. The third possibility for a clash are *room clashes*, no two classes can be taught in the same room at the same time. However, more than one class may share a timeslot and room as long as the total lecture hours does not exceed the T_t hours available in time-slot t.

Let us now formulate the problem as a mixed integer programming problem (MIP). Consider now the binary

[1]This is considered to be a separate timetabling problem for now.

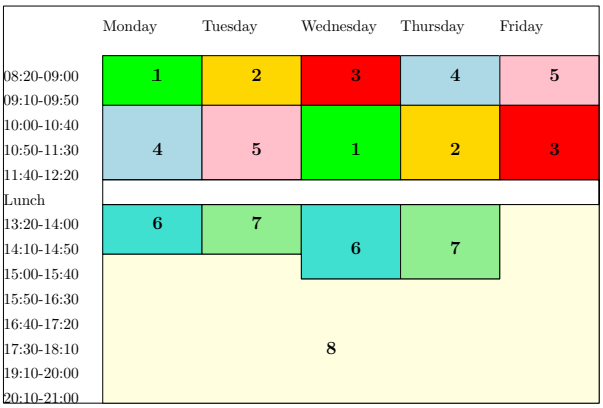

Figure 1. The seven time-slots and time-slot eight.

variable $x_{c,t,r}$ indicating whether a class c occupies room r in time-slot t. It will take the value 1 when true. Furthermore, let L_c be the number of lectures/tutorial needed for course c. Then we have, given the number of lectures T_t held by slot t,

$$\sum_{c\in\mathcal{C}} L_c x_{c,t,r} \leq T_t, \qquad t \in \mathcal{T}, r \in \mathcal{R} \setminus \{r_d\} \tag{1}$$

It may not be possible to allocate a room to all courses. To compensate for this classes can be assigned room r_d, implying that a room has yet to be found. The hard condition that no set of courses $\mathcal{C}_q \subset \mathcal{C}$ within a curriculum q should clash is formulated in a similar manner, that is

$$\sum_{c\in\mathcal{C}_q} L_c \sum_{r\in\mathcal{R}} x_{c,t,r} \leq T_t, \qquad t \in \mathcal{T}, q = 1, \ldots, n_q \tag{2}$$

where n_q is the total number of curricula. Furthermore, one must make sure that a course fits within a given timeslot, as follows

$$\sum_{r\in\mathcal{R}} L_c x_{c,t,r} \leq T_t, \qquad c \in \mathcal{C}, t \in \mathcal{T} \tag{3}$$

Each class must necessarily be assigned to some timeslot and room, but only once, that is

$$\sum_{r\in\mathcal{R}} \sum_{t\in\mathcal{T}} x_{c,t,r} = 1, \qquad c \in \mathcal{C} \tag{4}$$

This also means that any given class must fall within a single time-slot. If the required lecturing hours for a class is greater than T_t it will not fit within that timeslot. The total number of lecturing hours held by timeslot eight is, nevertheless, such that any class size can be placed there. Thus guaranteeing that a feasible solution will be found. A similar issues arises when assigning rooms, as the number of rooms is scarce. The schools within the University will initially assign as many classes as they can to the rooms within their buildings, after that one must "compete" for the remaining rooms within the entire University campus. Classes with few students, say S_c, are typically not assigned to a room at first for this reason. In order for constraint (4) to be feasible the dummy room r_d can effectively be used to assign any class. However, it should be made undesirable to assign classes to this room, which will be reflected in the objective function. Indeed assigning a class to this room means that the class has yet to be assigned a room. For this reason it would be necessary to assign all classes with few students to this room. In other words

$$\sum_{c\in\mathcal{C}, t\in\mathcal{T}: S_c \leq \underline{S}_c} x_{c,t,r_d} = 1 \tag{5}$$

where by $S_c > 0$ is the number of students enrolled in class c.

The number of students attending a course must not exceed the room capacity by a large number. Typically the number of students attending class will be less than that enrolled, although most will show up for lectures the first week. As a rule of thumb it is generally possible to overfill a room by up to 20%, or by a factor $f_r = 1.2$. If the room capacity is C_r, we say that

$$x_{c,t,r} \leq f_r C_r / S_c, \quad c \in \mathcal{C}, t \in \mathcal{T}, r \in \mathcal{R} \setminus \{r_d\} \tag{6}$$

and will work to reduce the size of the search space.

The objects of the timetabling problem are more than one. One would like to assign classes before lunch if possible and so create a tight morning schedule. The utilization of the rooms should be maximal and attempts should be made to assign as many courses as possible to the available rooms at the school. The number of course clashes should be minimal. Consider now the following objective function

$$\min_{\boldsymbol{x}}$$

$$W_1 \sum_{c \in \mathcal{C}} \sum_{r \in \mathcal{R}} \left(x_{c,6,r} + x_{c,7,r} \right) + W_2 \sum_{c \in \mathcal{C}} \sum_{r \in \mathcal{R}} x_{c,8,r} \tag{7}$$

$$+W_3 \sum_{c \in \mathcal{C}} \sum_{r \in \mathcal{R} \setminus \{r_d\}} \sum_{t \in \mathcal{T}} (f_r C_r - S_c) x_{c,t,r} \tag{8}$$

$$+W_4 \sum_{c \in \mathcal{C}, t \in \mathcal{T}: S_c > S_c} x_{c,t,r_d} \tag{9}$$

$$+W_5 \sum_{t \in \mathcal{T}} \sum_{c_1 \in \mathcal{C}, c_2 \in \mathcal{C}: c_1 < c_2} z_{t,c_1,c_2} M_{c_1,c_2} \tag{10}$$

Objective (7) forces courses to be assigned to time-slots $1-5$, with $W_1 < W_2$, timeslots $6-7$ are preferable to timeslot 8 (the dummy timeslot). The objective (8) will attempt to maximize room utilization and objective (9) will make if undesirable to assign a class to no room (the dummy room r_d). The part of the objective function that deals with the *course clashes* is given by (10). It uses a new variable z_{t,c_1,c_2} which will indicate that courses c_1 and c_2 don't fit in the same time-slot t. The binary indicator variable z_{t,c_1,c_2}, will be set to one in this case. This may be determined by introducing the constraint

$$L_{c_1} \sum_{r \in \mathcal{R}} x_{c_1,t,r} + L_{c_2} \sum_{r \in \mathcal{R}} x_{c_2,t,r} \leq T_t + z_{t,c_1,c_2}(L_{c_1} + L_{c_2} - T_t), \tag{11}$$

$$t \in \mathcal{T}, c_1 \in \mathcal{C}, c_2 \in \mathcal{C} : c_1 < c_2, M_{c_1,c_2} \geq 1$$

where $0 \leq z_{t,c_1,c_2} \leq 1$ is a continuous variable and will naturally tends towards zero, due to the objective function. When the courses c_1 and c_2 clash z's value will take its upper limit, which is one. The parameter M_{c_1,c_2} gives the total number of students taking both courses c_1 and c_2. Clearly we need only consider the cases when $c_1 < c_1$ and when $M_{c_1,c_2} > 0$. There is one potential problem with this last constraint. This is the possibility of placing more than two course in any time-slot and room, these constraints do not cover this situation. For this reason the following additional constraint is needed,

$$\sum_{c \in \mathcal{C}} x_{c,t,r} \leq 2, \quad t \in \mathcal{T}, r \in \mathcal{R} \setminus \{r_d\} \tag{12}$$

This way only a maximum of two courses are allowed at any time-slot and room.

III. Clustering common courses

The model presented in the previous selection can vary in difficulty. If less emphasis is put on (9), how often the dummy room is used, the easier the problem. When little emphasis is put on avoiding timeslot eight, so too the problem becomes easier. Better utilization of rooms and timeslots makes the problem challenging for the MIP solver. Furthermore, the addition of the course clash objective (10), with the introduction of variable z_{t,c_1,c_2}, makes the problem difficult to solve. One approach to making the problem more tractable for the MIP solver is to reduce the size of the search space. One obvious approach would, for example, be to set a minimal tolerance for the number of students, for any pair of courses, that can be in a clash. That is, add the condition that $M_{c_1,c_2} \geq \underline{M}$, some lower bound $\underline{M}$. This would be an additional condition in (10) and (12). For example, $\underline{M} = 3$ would imply that one or two students in any two courses that clash may be ignored. However, even this low value of 3, if acceptable, still makes the problem difficult to solve in a reasonable time. For larger values the objectives will end up being ignored, and so the solution becomes suboptimal.

The approach taken here to make the problem more tractable is to reduce the search space by finding *pseudo curricula* and introducing them in an equivalent manner to the regular curricula constraints. This is achieved by applying a clustering method to discover automatically *pseudo curricula* from the student registrations. For this purpose it was found that a centroid based clustering techniques formed better clusters than connectivity based methods such hierarchical clustering. Furthermore, the well known k-means clustering method [5] will be used in our study.

In the extreme case all pairs of courses c_1 and c_2 with $M_{c_1,c_2} \geq 1$ could be set as a pseudo curricula. As these would be hard constraints, they would potentially force courses to be placed in timeslot eight and the dummy room. This is an undesirable side-effect. Indeed, only courses with many required lecture/tutorial courses should be placed in timeslot eight. This extreme constraint, written as

$$L_{c_1} \sum_{r \in \mathcal{R}} x_{c_1,t,r} + L_{c_2} \sum_{r \in \mathcal{R}} x_{c_2,t,r} \leq T_t, \tag{13}$$

$$t \in \mathcal{T}, c_1 \in \mathcal{C}, c_2 \in \mathcal{C} : c_1 < c_2, M_{c_1,c_2} \geq 1$$

may, however, actually be useful if the condition $M_{c_1,c_2} \geq 1$ would be relaxed a little, to allow for potentially more clashes, say for example $M_{c_1,c_2} \geq 8$.

There is a further benefit to using clustering over the minimizing only object (10), this is that a group of students will observe fewer clashes and so also the individuals there within. This is because the clustering will discover subsets of courses where more than a pair of courses may not clash, as opposed to just the pair c_1 and c_2. In effect the individual student will experience fewer clashes. This is an important aspect, since for the students, being in one clash may be acceptable whereas two or more impossible.

The automatic technique used to discover pseudo curricula works as follows. First of all remove all students from the data that fit perfectly to their schools set course curricula $\mathcal{C}_q$. Then repeat the following process:

1) Perform the k-means clustering on the remaining students, with n_k clusters.
2) Treat each of the n_k clusters as a new curriculum.
3) Examine how many students actually take exactly this pseudo curricula, there should be at least one. If so, add this new pseudo curriculum $\mathcal{C}_p$ to the set of new curricula.
4) Remove all students that fit this new set of pseudo curricula.
5) If no new pseudo curricula was found then stop, else return to step 1).

The pseudo curricula is now added to the model described in the previous section, in a similar way to the regular curricula, that is,

$$\sum_{c \in \mathcal{C}_p} L_c \sum_{r \in \mathcal{R}} x_{c,t,r} \leq T_t, \qquad t \in \mathcal{T}, p = 1, \ldots, n_p \quad (14)$$

where n_p is the total number of pseudo curricula discovered.

There are two algorithm settings needed for the k-means clustering, the number of clusters n_k and the distance metric used. The data used in the clustering are binary vectors whose length is equal to the total number of courses held. Each bit in the vector corresponds to a course and when set to true the student is registered in the corresponding course. The cityblock (L1 distance) and Hamming distances are the most suitable measures for this representation. The centroid is then the component-wise median of the points in that cluster. A centroid then represents a pseudo curriculum. We will now illustrate the technique on a typical semester in what follows.

IV. Experimental Study

The data used in the study is from the autumn semester 2014 for the School of Engineering and Natural Science at the University of Iceland. There are 2166 students are pursuing their studies in their respective fields, which result in a total of $n_q = 89$ curricula depending on speciality tracks and year of study. The classes on offer are 204. The number of students that take only a given curricula and no other courses are depicted in the black colour in figure 2. This is around one third of all students. These students will not experience any course clashes, due to constraint (2). The figure also depicts students that fall entirely within a typical set of pseudo curricula, in the dark gray colour, found using clustering. This particular set contains in total $n_p = 286$ specific pseudo curricula. These are around one fifth of the students. A further one tenth of the students take only a single course and so experience no clashes. Given then that a feasible solution is found one can then estimate that two thirds of the students will experience no clashes. The rest of the students, depicted in the lighter colour, taking two or more classes rely on the minimization of objective (10) to avoid clashes.

The k-means clustering algorithm is initialized randomly with $n_k = 10$ clusters. For this reason we have investigated the performance of the technique, described in the previous section, by performing 30 independent experiments. The number of pseudo curricula n_p versus the number of students that fit these pseudo curricula is shown in a 2D boxplot in figure 3. As one can see the performance varies. In our experiments we will simply use the result of one of these experiments, the one used in figure 2.

The courses with four students or less are forced into the dummy room as discussed previously. For this particular semester these are 30 courses in total. The total number of rooms used in the study is 25 (plus

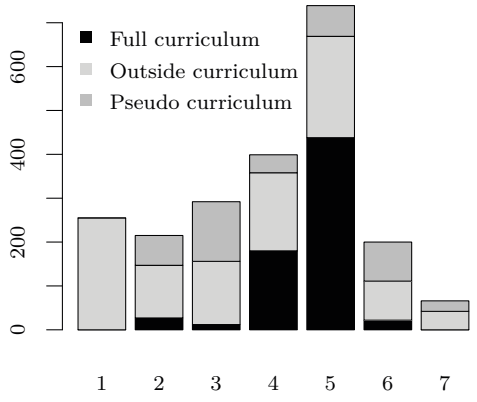

Figure 2. The number of students registered in one to seven courses. Students are put in one of three groups, those following strictly a full-, pseudo- and outside curricula.

the dummy room). These are the actual rooms preferred by the school. There are 8 courses which have over 195 students registered, which exceeds the capacity of the largest room. In figure 4 the room capacity of the 25 rooms are plotted as horizontal lines. The figure also plots on the horizontal axis the courses, those with more than 8 students, against the number of students registered for the course. There one can see that eight courses are above the 195 student limit on the far right side. At the level below there are 21 courses that would potentially be assigned to the 195 student room. At least seven can be accommodated, or more depending of weekly hours of lecturing. So in total at least $30 + 8 + (14) \approx 52$ classes will be forced to the dummy room, or slightly less.

A. General setup

All the computational experiments were conducted using the MIP solver Gurobi [6]. The runs were performed on a desktop computer with a 16 core Intel(R) Xeon(R) CPU E5-2650 0 @ 2.00GHz, using sixteen threads. The internal memory on the machine is 32GB which is more than sufficient for the problem, the runs below consume up to around one third of this memory. The weights for the objectives were set to $W_2 = W_4 = 100$, $W_5 = 10$ and $W_1 = W_3 = 1$. These were arbitrarily chosen, but reflect the importance of each objective. The runs were terminated after a time limit of 3 hours.

B. Models

The following model settings are investigated

1) The plain model presented in section II.
2) Plain model with constraint (13), but with the condition that $M_{c_1,c_2} \geq \underline{M}$ (described below).
3) Plain model with constraint (13), but with the condition that $M_{c_1,c_2} \geq 8$.
4) Plain model with the pseudo curriculum constraint (14).

All models presented in the previous section were unsolvable and terminated after 24 hours. Because of this some approximation was needed to make the problem solvable within reasonable time. The approximation is as follows, in object (10) the additional condition was added that only $\underline{M}$ students or more taking a pair of common courses need be considered. That is,

$$\ldots + W_5 \sum_{t \in \mathcal{T}} \sum_{c_1 \in \mathcal{C}, c_2 \in \mathcal{C}: c_1 < c_2, M_{c_1,c_2} \geq \underline{M}} z_{t,c_1,c_2} M_{c_1,c_2} \quad (15)$$

and set $z_{t,c_1,c_2} = 0$ for the cases where $M_{c_1,c_2} < \underline{M}$. Two different settings for $\underline{M}$ are used, 4 and 5, which obtain a reasonable gap.

C. Computational results

A suitable gap[2] or around 0.2%, or less, was achieved after three hours of computation. Runs using $\underline{M} = 4$ versus using 5 requires twice the internal memory. The key objective values for the four different settings are given in table II and I, using $\underline{M}$ equal to 4 and 5 respectively. In table I we see that model 4) requires one or two additional courses to be in timeslot 8 and the room utilization is worse than the other models. Model

[2]The gap is the objective bound minus incumbent object value all divided incumbent object value.

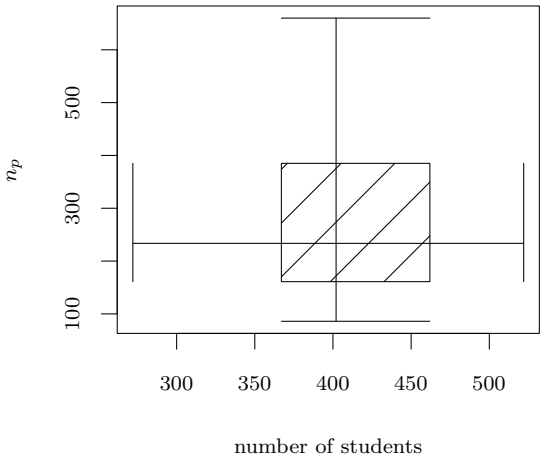

Figure 3. Boxplot of the number of students fitting exactly the pseudo curricula versus the number of pseudo curricula created, based on 30 independent clustering experiments.

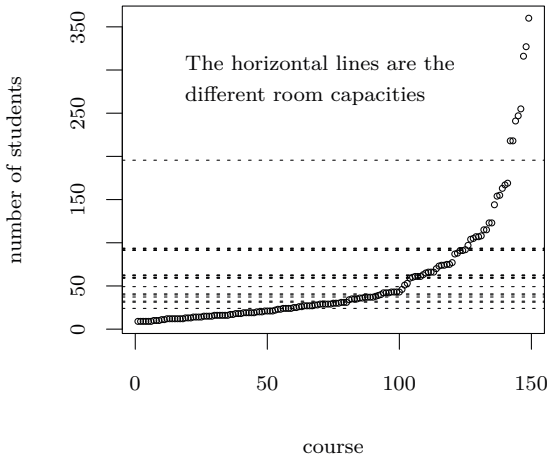

Figure 4. The number of students registered versus courses. The maximum allowable room capacity for the 25 rooms are plotted as horizontal dotted lines.

3) achieves fewer course clashes than model 1), otherwise its performance is similar. The fewest course clashes are achieved here using model 2) and 4). In general model 4) is far superior in terms of student course clashes. We would have expected to get even better results with model 4) with $\underline{M}$ equal to 4, but it may be that a longer computation time that 3 hours may be needed. When reducing the value of $\underline{M}$ the more computation time is needed and also the internal memory. The internal memory required doubles when $\underline{M}$ is reduced from 5 to 4.

Table I
SUMMARY OF RESULTS IN TERMS OF VARIOUS SOLUTION ATTRIBUTES. WITH $\underline{M} = 5$ AND OVER CONSIDERED IN THE OBJECTIVE FUNCTION.

Constraints	$\sum\sum(x_{c,6,r} + x_{c,7,r})$	$\sum\sum x_{c,8,r}$	$\sum\sum z_{t,c_1,c_2}$	$\sum z_{t,c_1,c_2} M_{c_1,c_2}$	$\sum r_d$	$\sum\sum\sum(f_r C_r - S_c)x_{c,t,r}$	Incumbent	Gap
1)	61	1	171	276	50	1903	7141	0.17%
2)	60	2	166	290	50	1904	7164	0.18%
3)	60	1	169	290	50	1932	7142	0.18%
4)	63	3	122	203	49	2022	7285	0.22%

Table II
SUMMARY OF RESULTS IN TERMS OF VARIOUS SOLUTION ATTRIBUTES. WITH $\underline{M} = 4$ AND OVER CONSIDERED IN THE OBJECTIVE FUNCTION.

Constraints	$\sum\sum(x_{c,6,r} + x_{c,7,r})$	$\sum\sum x_{c,8,r}$	$\sum\sum z_{t,c_1,c_2}$	$\sum z_{t,c_1,c_2} M_{c_1,c_2}$	$\sum r_d$	$\sum\sum\sum(f_r C_r - S_c)x_{c,t,r}$	Incumbent	Gap
1)	61	1	164	249	50	1933	7145	0.22%
2)	60	2	142	211	50	1903	7164	0.19%
3)	62	1	147	235	50	1933	7145	0.22%
4)	61	3	141	217	50	1951	7312	0.21%

V. Conclusion and discussion

The idea of using clustering to find pseudo-curricula is two fold. Firstly, to shrink the feasible search space and secondly to reduce the number of individual students having numerous clashes. It was also shown that the search space could be shrunk simply using constraint (13) with some suitable tolerance on the number of students taking common courses, that is M_{c_1,c_2} should be greater than 8 was used in the experiments. Using a value of 1 implies that no clashes are tolerated. This would mean more courses are forced into time-slot eight. The objective function that attempts to minimize all course clashes fails to solve the problem to optimality even after 24 hours. However, when this condition is relaxed to allow for more clashes it can be solved with a reasonable gap within a few hours. A suitable gap was achieved with $\underline{M}$ as low as 4.

The use of clustering not only helps in building more efficient timetabling models, but also gives insight into how the students are taking their degree. The work described is work in progress. The next steps are to investigate more thoroughly the number of clashes experienced by the individual students. In the extreme case each students can be treated as a curriculum and the object would be to satisfy as many of these student curricula as possible. This may be achieved by setting the students curriculum as soft constraints. However, such an approach may be fruitless as it will introduce even more binary variables to the problem. However, incrementally adding students that have many clashes in their timetable as a pseudo-curricula may be one answer.

References

[1] A. Bettinelli, V. Cacchiani, R. Roberti, and P. Toth, "An overview of curriculum-based course timetabling," *Top*, vol. 23, no. 2, pp. 313–349, 2015. [Online]. Available: http://link.springer.com/10.1007/s11750-015-0366-z

[2] E. K. Burke, J. H. Drake, B. McCollum, and E. Özcan, "Comments on: An overview of curriculum-based course timetabling," *Top*, vol. 23, no. 2, pp. 355–358, 2015. [Online]. Available: http://link.springer.com/10.1007/s11750-015-0362-3

[3] N. Pillay, "A survey of school timetabling research," *Annals of Operations Research*, vol. 218, no. 1, pp. 261–293, 2014.

[4] ——, *A review of hyper-heuristics for educational timetabling*, 2014.

[5] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Applied statistics*, pp. 100–108, 1979.

[6] I. Gurobi Optimization, "Gurobi optimizer reference manual," 2015. [Online]. Available: http://www.gurobi.com