GRANULAR MODELLING OF EXAM TO SLOT ALLOCATION

Siti Khatijah Nor Abdul Rahim[1], Andrzej Bargiela[1], Rong Qu[2]
[1] School of Computer Science
University of Nottingham, Malaysia Campus
E-mail: abb@cs.nott.ac.uk, khyx8skn@nottingham.edu.my
[2]School of Computer Science
University of Nottingham, UK Campus
E-mail: rxq@cs.nott.ac.uk

KEYWORDS
Information granulation, conflict chains, spread matrix, pre-processing, exam-to-slot allocation.

ABSTRACT

In this paper, we are introducing a new method of granular exam-to-slot allocation based on the pre-processing of the basic student-exam information into a more abstract (granulated) entity of conflict chains. Since the conflict chains are designed to capture the mutual dependencies between exams, they enable us to reason about the exam-to-slot allocation for all exams in a chain rather than just one exam at-a-time. The initial exam-to-slot allocation, generated through the processing of conflict chains, is then refined by considering the spread of the exams in the examination session so as to minimize the appropriately defined cost function. The granular pre-processing of problem data has been shown to enhance the efficiency of the exam scheduling task and has led to the identification of very competitive exam schedules.

INTRODUCTION

Exam timetabling represents a difficult computational problem due to the strong inter-dependencies between exams caused by the many-to-many relationship between students and exams. Timetabling researchers have investigated various methods of exploring the large combinatorial search space to generate timetables. These methods were typically formalized as heuristic assignment algorithms. However, it has been observed that no single heuristic that can be used to solve all timetabling problems because of the incorporation of problem-specific features in the heuristics (Burke et. al, 1994). In their survey (Qu et. al, 2006) point to the specialization of the timetabling research into sub-areas of educational timetabling, nurse scheduling, transport scheduling, sports timetabling, etc. However, according to the authors the most studied and researched timetabling problem is the educational timetabling and in particular, exam timetabling. The survey highlights families of related heuristics deployed in the solution of timetabling problems which include: graph heuristics, meta-heuristics, constraint based methods, multi-criteria techniques, hybridizations, and methods that focus on the investigation of neighbourhoods in the solution space.

In exam timetabling problems, the constraints are normally different from one institution to the other, which makes it difficult to define the “standard timetabling problem”. However, at a general level the exam timetabling can be thought of as a process of ensuring that all students are able to take their exams and that the schedule of examinations for each student is designed so as to maximize the gap between consecutive examinations.

In a more formal way, the timetabling literature defines two types of constraints:

a) Hard Constraints
These are the constraints that must be satisfied at all times. The principal hard constraint is the requirement that not a single student is enrolled for two exams scheduled in the same timeslot. Another hard constraint that needs to be obeyed is the room capacity; i.e. there must be enough spaces in a room to accommodate all students taking a given exam. A timetable, which satisfies hard constraints, is called a feasible timetable.

b) Soft Constraints
Soft Constraints are not critical but their satisfaction is beneficial to students and/or the institution. An example of a soft constraint is the requirement to spread out exams taken by individual students so that they have sufficient revision time between the exams they are enrolled on. Typically one cannot satisfy all soft constraints thus there is a need for a performance function measuring the degree of satisfaction of these constraints.

Since the complexity of the timetabling methods reported in the literature seems to originate from the many-to-many relationship between the students and the exams we are proposing, in this paper, to cast the timetabling problem as a granular modeling problem in which the detailed student-exam data is pre-processed into some granular entities before these are used for expressing the timetabling problem itself. Apart from generating feasible timetable, our focus is to produce a good quality timetable, which satisfies the soft constraints to the greatest possible extent. To demonstrate the general applicability of the proposed granular modeling we produce solutions for both uncapacitated and capacitated versions of the timetabling problems, reported in the literature. In is worth noting here that the capacitated version of the exam timetabling problem means that the problem is subject to additional constraints on total room capacity in each timeslot. Consequently the expected quality of the timetables, as measured by the value of the cost function, is necessarily inferior compared to the uncapacitated version of the problem. For this paper, we have used the University of Nottingham Semester 1, 1994 dataset that can be downloaded from the URL: http://www.cs.nott.ac.uk/~rxq/files/Nott.zip.

The quality of the timetable is measured using the cost function originally proposed by Carter (Carter et al, 1996) and also used by other researchers (Burke et al, 2004). The function is:

$$\frac{1}{T}\sum_{i=1}^{N}\sum_{j=i+1}^{N} s_{ij}\, w_{|\mathrm{pj}\ -\ \mathrm{pi}|} \tag{1}$$

where N is the number of exams, s_{ij} is the number of students enrolled in both exam i and j, pj is the time slot where exam j is scheduled, pi is the time slot where exam i is scheduled and T is the total number of students. Based on this cost function, a student taking 2 exams that are t slots apart, where t={1, 2, 3, 4, 5, 6}, will give rise to the costs of 16, 8, 4, 2, 1 and 0 respectively.

We also used another cost function that considered only the number of students having exams on the same day and students having exams separated by overnight break.

INFORMATION GRANULATION IN THE ALGORITHM

The proposed granular modeling of the timetabling problem starts with the abstraction of the detailed student-exam data into an aggregated (granular) entity describing mutual dependence of individual exams (arising from students taking both exams). We refer to this granular construct as a "conflict chain". The rationale for building conflict chains is that once an interrelated set of exams has been identified, allocating any of the exams in the conflict chain to a specific time slot implies a unique relative allocation of all other exams in the conflict chain to their respective time slots. In other words, we take advantage of the fact that the potential conflict between exams does not depend on the absolute allocation of exams to time slots. On the other hand the all-important relative allocation of exams to time slots is captured within one entity thus making any subsequent reassignment very straightforward. The generation of conflict chains is accomplished as a simple pre-processing of the original problem data. The resulting re-definition of the timetabling problem, away from the allocation of individual exams to time slots and towards the allocation of conflict chains to the corresponding sets of time slots is rooted in the Granular Computing research (Bargiela, et. al, 2002, 2008).

The first pre-processing step is to determine potential clashes between examinations and to count the number of students that cause these clashes. Since the potential clashes (unlike the actual ones) do not depend on the subsequent allocation of exams to time slots they represent useful information granules that can be utilized in the timetabling process. Based on these information granules we determine the minimum number of time slots that are necessary for timetabling the given set of examinations. We refer to this stage as the construction of conflict chains. The algorithm deployed at this stage can be summarized as follows:

1. Initiate the algorithm by allocating all exams to time slot one.
2. Select the first exam as "current" and initiate the counter for the current conflict chain.
3. Label the current exam as "allocated to the current chain" and note all the exams that are in potential conflict with the current exam.
4. If the list of potentially conflicting exams is non-empty, re-allocate those exams to the next available time slot. Otherwise label the current chain as complete and proceed to Step 6.
5. If the list of potentially conflicting exams is non-empty, select the first exam from the list and repeat from Step 3 with the currently selected exam.
6. Check if all exams allocated at Step 1 are belonging to one of the conflict chains; if YES than the algorithm terminates; if NO, then the conflict chain counter is incremented and the unallocated exam is taken as "current" for processing starting from Step 3.

The outcome of the above algorithm is a set of conflict chains that represent mutually dependent exams that need to be scheduled in different time slots so as to avoid violation of hard constraints. However, the algorithm implies that it is possible to have one exam belonging to more than one conflict chain (albeit the algorithm will ensure that the allocation of this exam to time slot is consistent in both chains). For this reason we perform additional step of merging these conflict chains, which happen to have common exams. The merged conflict chains represent independent subsets of the examination set that can be dealt with one-at-a-time.

It is worth noting that the length of the conflict chain, measured as the number of necessary time slots to schedule the exams in this chain, provides an immediate indication of the difficulty of the specific timetabling task. If the number of available time slots is equal or only slightly greater than the length of the longest conflict chain than the problem is heavily constrained and we might expect that the quality of the resulting timetable, measured by the cost function, will be low. On the other hand, if the number of the available time slots is significantly greater than the length of the longest conflict chain than we can expect high quality (low cost) solutions. Of course, if the number of available time slots is less than the length of the conflict chain then the timetabling problem is infeasible.

Consequently, the information granule of conflict chain provides a useful insight into the timetabling problem from a more general perspective compared to the one afforded by considering individual exams.

	1	2	3	4	5	6	7
309	11	0	11	0	0	0	0
310	2	0	0	0	0	0	0
311	7	0	0	0	7	0	0
312	6	0	0	0	0	0	0
313	1	0	0	0	0	0	0
314	1	0	1	0	0	0	0
315	3	0	0	0	0	0	0
316	19	0	19	0	0	0	0
317	0	0	0	0	0	0	1
318	6	0	0	0	0	0	0
319	11	0	0	0	0	0	0
320	10	0	10	0	0	0	0
321	0	0	0	0	0	0	2
322	11	0	0	0	0	0	0
323	0	0	0	0	0	8	0
324	10	0	10	0	0	0	0
325	11	0	11	0	0	0	0
326	2	0	0	0	0	0	0
327	9	0	0	0	0	0	0
	:	:	:	:	:	:	:
439	4	0	4	0	0	0	0
440	1	0	0	0	0	0	1
441	13	0	13	0	0	0	0

Figure 1: Conflict Chains Before Merging

Figure 1 illustrates a representative section of the output of the conflict chain generation algorithm. The chain label is given in the first row and the exam number is indicated in the first column. It can be seen that conflict chains 1, 3 and 7 share some of the exams between themselves. For example, exam 309 is shared between conflict chain 1 and 3, exam 311 is shared between conflict chain 1 and 5 and exam 440 is shared between conflict chains 1 and 7. This means that conflict chains 1, 3 and 7 can be justifiably merged into one conflict chain. It is worth noting that merging conflict chains 1 and 7 implies that the merged conflict chain will include also exams 317 and 321 even if they appeared only in conflict chain 7.

Figure 2, illustrates the resulting merged conflict chain under the chain label 1. Conflict chains 2, 4 and 6 are unaffected by this post processing as they happen not to have any common exams with those from chain 1. In particular, it is worth noting that the exam 323 is assigned to time slot 8 in conflict chain 6 but, since it is not in conflict with any of the exams in chain 1, it has label 0 in the merged conflict chain 1.

	1	2	3	4	5	6	7
309	11	0	0	0	0	0	0
310	2	0	0	0	0	0	0
311	7	0	0	0	0	0	0
312	6	0	0	0	0	0	0
313	1	0	0	0	0	0	0
314	1	0	0	0	0	0	0
315	3	0	0	0	0	0	0
316	19	0	0	0	0	0	0
317	1	0	0	0	0	0	0
318	6	0	0	0	0	0	0
319	11	0	0	0	0	0	0
320	10	0	0	0	0	0	0
321	2	0	0	0	0	0	0
322	11	0	0	0	0	0	0
323	0	0	0	0	0	8	0
324	10	0	0	0	0	0	0
325	11	0	0	0	0	0	0
326	2	0	0	0	0	0	0
327	9	0	0	0	0	0	0
:	:	:	:	:	:	:	:
439	4	0	0	0	0	0	0
440	1	0	0	0	0	0	0
441	13	0	0	0	0	0	0

Figure 2: Conflict Chains After Merging

Besides generating the independent conflict chains, as outlined above, we evaluate the number of students who take exams allocated to time slots that are 1, 2, .., 5 time slots apart. Since we are dealing with information granules that represent a potential conflict between all exams in one time slot and all exams in another time slot, regardless of what are the actual time slot numbers, we create a framework for efficient optimization of the cost function (quality of the timetable). In what follows we will describe the proposed scheme for re-numbering of the time slots using the background knowledge about the structure of the cost function. We refer to this optimization stage as maximizing the spread of examinations.

The spread information is represented as an [sxs] matrix, where s is a number of time slots in the given timetabling problem. Figure 3 illustrates a section, representing the first six time slots, of the spread matrix evaluated for the Nottingham Dataset. We can observe that there are 1454 students who take an exam in time slot 1 and 2, 1360 students taking exams in time slot 1 and 3, etc. Clearly, the spread matrix is symmetrical with diagonal elements being omitted from the calculation.

Taking into account the format of the cost function that assigns the weight "16" to the exam spread of 1 (i.e. entries in the spread matrix (1,2), (2,3), (3,4) etc.) and assigns weight "8" to the exam spread of 2 (i.e. entries in the spread matrix (1,3), (2,4), (3,5), etc.), and so on, we can reformulate the task of optimization of the exam spread as a task of re-arranging the time slots so as to achieve the smallest sum of elements on the first minor diagonal in the re-arranged spread matrix. Taking this method to a logical conclusion we aim at making sure that some particularly large numbers in the spread matrix are shifted to minor diagonals that are of order 6 or above (thus not imposing penalties in the cost function). Of course, re-numbering of time slots is subject to strong mutual dependencies so one needs to define some compromise strategies.

	1	2	3	4	5	6
1	0	1454	1360	1717	1276	1006
2	1454	0	1355	1634	1085	997
3	1360	1355	0	1392	1158	947
4	1717	1634	1392	0	1529	1446
5	1276	1085	1158	1529	0	1120
6	1006	997	947	1446	1120	0

Figure 3: Initial Ordering of the Spread Matrix for the First 6 Slots from the Nottingham Dataset Based on the Largest Degree (LD) Decreasing Ordering

We propose here two strategies for optimization of the exam spread named as Method 1 and Method 2. A brief explanation of each method is given below:

Method 1

The first method is focused on extracting the smallest element in each row of the original spread matrix and re-numbering the relevant time slots so as to place the smallest element on the first minor diagonal. While implementing such re-numbering it is possible that higher order minor diagonals will have some greater elements associated with them. However, if the primary concern is to minimize the number of adjacent exams the method provides the optimum solution.

Method 2

The second method takes another approach of identifying the smallest elements in both rows and columns and shifting them towards the first minor diagonal. This corresponds to arranging simultaneously from the first slot and from the last slot towards the middle one. It turns out that by taking this approach a more balanced re-numbering is achieved that attempts to minimize the sum of higher minor diagonals.

The first six rows-columns of the re-numbered spread matrix using Method 1 and 2 are shown in Figure 4 and Figure 5 respectively. However, one needs to remember that some of the rows-columns represented in Figures 4 and 5 do not appear in Figure 6 because they had their corresponding time slot number greater than 6. Nevertheless the sample spread matrices serve to illustrate the main characteristics of the two methods.

	1	2	3	4	5	6
1	0	1006	1360	1276	1454	1717
2	1006	0	947	1120	997	1446
3	1360	947	0	1158	1355	1392
4	1276	1120	1158	0	1085	1529
5	1454	997	1355	1085	0	1634
6	1717	1446	1392	1529	1634	0

Figure 4: The New Arrangements of the Initial Ordering of the Spread Matrix after Applying Method 1

	1	2	3	4	5	6
1	0	1006	1717	1360	1454	1276
2	1006	0	1446	947	997	1120
3	1717	1446	0	1392	1634	1529
4	1360	947	1392	0	1355	1158
5	1454	997	1634	1355	0	1085
6	1276	1120	1529	1158	1085	0

Figure 5: The New Arrangements of the Initial Ordering of the Spread Matrix after Applying Method 2

It is worth highlighting here that our proposed solution to the timetabling problem is fundamentally different from the previous solutions based on consideration of one-exam-at-a-time (Lewis, 2008). By pre-processing the original data into suitable information granules we were able to re-formulate the problem as an optimization of minor traces of the spread matrix.

COMPUTATIONAL RESULTS

The proposed method has been evaluated on a real-life dataset representing examinations at the University of Nottingham scheduled during the 1994-5 autumn semester (Table 1).

Table 1: University of Nottingham Dataset (Nott), Semester 1, 1994-1995 Used in Our Experiment

Dataset	Number of students	Number of exams	Number of enrollments	Conflict density
Nott	7896	800	33997	0.03

In order to ensure that the conflict chains are produced as efficiently as possible, we have pre-ordered the exams using the Largest Degree (LD) method (i.e. gave priority to the exams that have the largest number of conflicting exams). The granulation of this data set into conflict chains resulted in the identification of 18 time slots as the minimum number required for the satisfaction of hard constraints. The cost function (1) evaluated on the merged conflict chains without- and with the optimization of the exam spread using Method 1 and Method 2 are presented in Table 2:

Table 2: Cost Functions Before and After Considering the Spread Information for the Uncapacitated Nott Dataset.

No of slots	18
Cost Function with original ordering of time slots	43.91
Cost Function After Rearrangement of slots using Method 1	29.03
Improvement Percentage (%)	33.89
Cost Function After Rearrangement of slots using Method 2	**24.18**
Improvement Percentage (%)	44.93

It is clear that the optimization of the spread matrix by re-numbering of the exam slots leads to a significant improvement of the cost function. It is worth noting here that we are considering the smallest number of time slots that allows generation of the feasible schedule. As such, the cost function is necessarily large because there is little room for maneuver with moving time slots around.

An alternative version of the Nottingham exam-scheduling problem involves on one hand a relaxation of the constraint on the number of time slots from 18 to 23 and the introduction of an additional constraint on the number of students taking exams in any of the time slots (maximum number 1550). Using the same cost function (1) and evaluating the results for the original ordering of time slots and the optimized ordering using Method 1 and 2 we obtain results reported in Table 3.

Table 3: Cost Functions Before and After Considering the Spread Information for the Capacitated Nott Dataset.

No of slots	23
Cost Function with original ordering of time slots	22.51
Cost Function After Rearrangement of slots using Method 1	21.29
Improvement Percentage (%)	5.42
Cost Function After Rearrangement of slots using Method 2	**19.61**
Improvement Percentage (%)	12.88

The inspection of the spread matrix generated by both methods reveals that the first method tends to over-emphasise the selection of small spread values on the first minor diagonal and by the time the optimisation gets to dealing with few remaining time slots the capacity constraints forces it to leave the high spread values on the bottom-right section of the first minor diagonal. By contrast the Method 2 takes a more balanced approach to optimizing the spread values and, in doing so, is less affected by the capacity constraint, thus producing lower overall cost.

The capacitated Nottingham exam-scheduling problem has been also evaluated using an alternative cost function as proposed in (Qu et al, 2006). In this version of the problem the objective is to minimize adjacent exams on the same day and overnight. By extracting information from the spread matrix, the values obtained are 3565 for adjacent periods and 1536 for overnights, in the original ordering of time slots. If we quantify this result using a function similar to (1) but defined only on the adjacent time slots we have: (3565 + 1536) * 16 / 7896, which equals to 10.34. Performing similar evaluation for the re-numbered time slots generated using Method 1 and 2 we obtain 6.13 and 11.60 respectively, as reported in Table 4:

Table 4: Cost Functions For Nott Dataset Based on Nott94b Evaluation.

			Penalty (Violations of the first

Method	Adjacent Periods	Overnights	constraint Using Carter et al, 1996)
Initial Ordering	3565	1536	10.34
Method 1	1729	1298	**6.13**
Method 2	3260	2465	11.60

The results reported in Table 4 are consistent with the nature of the Method 1 and 2. Since the cost function penalizes only the adjacent examinations the optimization implemented by Method 1 provides an optimal choice (resulting in the overall cost of 6.13). The more balanced approach to the optimization of the spread afforded by Method 2 does not produce any advantage here since it does not reflect the narrow focus of the cost function. Consequently the cost associated with Method 2 is not only worse than the one obtained with Method 1 but also is worse than the one for the original ordering.

In order to enable a cross-comparison of the results generated by granular modeling with those produced using alternative approaches, we have archived complete solutions reported here at URL http://www.intelligentmodelling.org.uk/UNMC/timetabling.html.

CONCLUSION

Information granulation is a powerful methodological approach to emphasizing the relevant information embedded implicitly in the raw data. Granular models of the problem that are built on such aggregated (granular) data can be much simpler in terms of computational complexity and indeed can lead to simple analytical or combinatorial optimization.

We have shown that by capturing the persistent feature of potential conflicts (as opposed to actual) in the conflict chain information granule one can construct a much simpler model of the exam-scheduling task. Such a granular model is then amenable to optimization by minimization of the trace of minor diagonals of the exam spread matrix.

Numerical experiments conducted on the Nottingham Dataset confirmed the insights about the potential of granular modeling of the exam scheduling problem and have delivered very competitive exam schedules.

REFERENCES

Asmuni, H, E.K. Burke, J.M. Garibaldi, and Barry McCollum. 2004. “Fuzzy Multiple Heuristic Orderings for Examination Timetabling”. E. K. Burke and M. Trick, editors, Practice and Theory of Automated Timetabling V (PATAT 2004, Pittsburg USA, August 2004, Selected Revised Papers), Lecture Notes in Computer Science 3616, Berlin, 2005. Springer, 334 – 353.

Bargiela A., Pedrycz W., 2002. “Granular Computing: An Introduction”. Kluwer Academic Publishers

Bargiela A., Pedrycz W., 2008, “Toward a theory of Granular Computing for human-centred information processing”. IEEE Trans. on Fuzzy Systems, vol. 16, 2, 320-330

Burke E. K., Elliman D. G., and Weare R. F. 1994. “A Genetic Algorithm for University Timetabling”. AISB Workshop on Evolutionary Computing, University of Leeds, UK.

Burke E.K., Newall J. and Weare, R. 1996. “A Memetic Algorithm for University Exam Timetabling”. In: Burke E.K. and Ross P. (eds.) Selected Papers from the 1st International Conference on the Practice and Theory of Automated Timetabling, Lecture Notes in Computer Science 1153, pp. 241-250.

Burke E.K. and S. Petrovic. 2002. “Recent Research Directions in Automated Timetabling’. European Journal of Operational Research, 140(2), 266-280.

Burke E.K., Bykov Y., Newall J. and Petrovic S. 2004. “A Time-Predefined Local Search Approach to Exam Timetabling Problems.” IIE Transactions on Operations Engineering, 36(6) 509-528.

Carter M., Laporte G. and Lee S. 1996. “Examination Timetabling: Algorithmic Strategies and Applications”. Journal of Operations Research Society, 47 373-383.

Djannaty F. and A.R. Mirzaei. 2008. “Enhancing Max-Min Ant System for Examination Timetabling Problem”. International Journal of Soft Computing 3 (3): 230-238.

Lewis R, “A Survey of Metaheuristic-based Techniques for University Timetabling Problems”. 2008. *OR Spectrum,* vol 30 (1) pp 167-190.

Qu, R., EK Burke, B McCollum, LTG Merlot, SY Lee. 2006, “A Survey of Search Methodologies and Automated Approaches for Examination Timetabling”. Computer Science Technical Report No NOTTCS-TR-2006-4, University of Nottingham, UK.

Taufiq Abdul Gani, Ahamad Tajudin Khader and Rahmat Budiarto, 2004, "Optimizing Examination Timetabling using a Hybrid Evolution Strategies". IN: Proceedings of the Second International Conference on Autonomous Robots and Agents (ICARA 2004), 13-15 December 2004, Palmerston North, New Zealand, pp. 345-349.