Combining Examinations to Accelerate Timetable Construction

Graham Kendall

Jiawei Li

School of Computer Science, University of Nottingham, UK

{gxk|jwl}@cs.nott.ac.uk

Abstract: In this paper we propose a novel approach that combines compatible examinations in order to accelerate both the initial timetable construction, as well as a later search. The conditions for combining exams are described, and we show that we are able to offer some guarantees as to the quality of solutions that remain in the reduced search space. The approach is applied to one of the standard benchmarks in this area; the St. Andrews83 instance. The results verify the effectiveness of this approach in simplifying examination timetabling problems, speeding up initial timetable construction and assisting any subsequent search.

Keywords: exam timetabling, heuristics, optimisation, combining exams

Introduction

Exam timetabling is a difficult combinatorial optimisation problem that is a key task in all educational institutions. The problem is concerned with distributing a collection of exams among a limited number of timeslots so that a set of constraints are satisfied. Various approaches have been developed with an objective to either obtain feasible timetables in short time or to achieve optimised timetables with low cost (Carter and Laporte (1996), Burke and Petrovic (2002), Qu et al (2006)). Since a full-tree search of an NP-complete problem is generally impractical, it is natural for researchers to consider methodologies such as heuristics, meta-heuristics and decomposition.

The idea of decomposition is to divide the original problem into smaller sub-problems so that each sub-problem can be handled by using relatively simple approaches (e.g. Carter (1983), Burke and Newall (1999), Qu and Burke (2007)). Cluster methods adopt similar ideas. In these approaches the exams are sorted and split into groups and then the groups are assigned to time periods within a greatly reduced search space (e.g. White and Chan (1979), Balakrishnan et al. (1992)). The main drawback of these methods is that the quality of the solutions that remain in the reduced search space may be poor.

The successful application of meta-heuristic methods, tabu search for example (Glover and Laguna (1993), Nonobe and Ibaraki (1998), White et al (2004), Gendreau and Potvin (2005)), in exam timetabling problems shows that 'good' solutions for a specific timetable have some common features. If the problems could be simplified according to these common features, 'good' solutions would remain in the reduced search space and then heuristic methods could not only improve the speed of convergence to a feasible solution but also the quality of the final timetable could come with some guarantees, with respect to the best solution that could be reached. In this paper, we present an approach that simplifies exam timetabling problems by combining multiple exams, so that they can be treated as a single exam. The conditions that dictate how examinations are combined are presented, under which (near) optimal timetables will still be achievable. Instead of dividing all exams into groups as cluster methods do, the proposed approach only combines those exams that satisfy predefined conditions.

For the problem of exam timetabling with m exams and n timeslots, the size of the search space is n^m. If two of the exams are combined, the size of the search space becomes n^{m-1}. With this smaller space of solutions, a previously used heuristic may either find feasible solutions in shorter time or reach a better solution in a given time.

In the following sections, we will first infer the conditions for exams to be combined, and then show the application of the proposed method on the St. Andrews83 benchmark instance.

The Approach

Consider a simple example of exam timetabling. Suppose that there are 10 students (*A, B, ..., J*), 6 exams (E1, …, E6) and 4 timeslots (s1, …, s4), as shown in Fig 1. The hard constraints stipulate that no student should sit for two exams at the same time. Our aim is to leave as much time as possible between each exam for every student, so that the students perceive the schedule as being fair from their perspective (Kendall and Mohd Hussin (2005) presents a typical mathematical formulation).

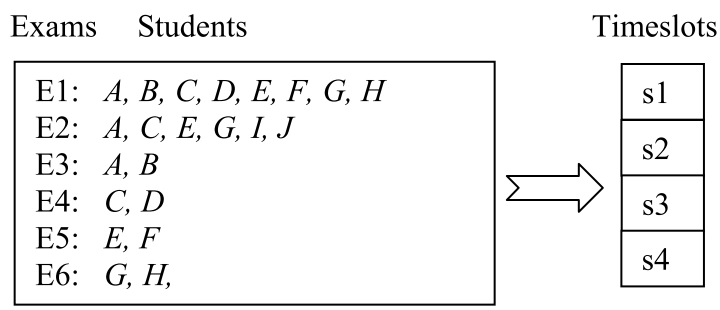

Figure 1 A simple example of exam timetabling

Although the optimal solution can be found for this problem, by using a full-tree search, there is a more efficient way. Exams E3, E4, E5, and E6 can be combined and treated as one exam, as they have no students in common. In this case, the problem is simplified to three exams. Thus, we are faced with the problem of defining under what conditions two or more exams can be combined together?

Of course, combining several exams together should not violate the hard constraints. That is, there must not be any common students in the exams we are combining.

Secondly, combining several exams together should minimize the soft constraints as far as possible. If the task is to find the optimal solution, combining these exams should ensure that the optimal solution(s) remain in the reduced search space. If, on the other hand, the task is to quickly find feasible solutions, combining exams should ensure that more feasible solutions remain in the reduced search space.

	E1	E2	E3	E4	E5	E6
E1	—	4	2	2	2	2
E2	4	—	1	1	1	1
E3	2	1	—	0	0	0
E4	2	1	0	—	0	0
E5	2	1	0	0	—	0
E6	2	1	0	0	0	—

Figure 2 Clash matrix (Clashes between exams are expressed by positive numbers).

The clash matrix of the example exam timetable (Fig. 2) reveals some common features these combined exams should have. The entries marked zero (in the dashed box) show that there is no clash between the two respective exams. The entries in the dotted box show those exams that clash with other exams 'in a similar way' - each of these exams has two clashes with E_1 and one clash with E_2. As will be discussed later in the paper, these equivalent numbers of clashes among these exams makes sure that combining them minimizes the penalty of a soft constraint so that good solutions can be retained in the search space. Therefore, the conditions for combining two or more exams can be expressed as:

1. There are no clashes between them.
2. They are equally clashed with other exams.

The optimal solution remains after combining two or more exams when Conditions 1 and 2 (above) are satisfied. We give this proof below.

Proof: We use reduction to absurdity. Suppose that two exams, E_1 and E_2, are scheduled into different timeslots, s_i and s_j, for the optimal solution of an exam timetabling problem. Let P_1 and P_2 denote the cost of scheduling E_1 to s_i and E_2 to s_j respectively. According to Condition 1 and 2, we can obtain a new feasible solution by moving E_1 from s_i to s_j. Comparing the new solution with the best solution, there should be

$$P_1 + P_2 \leq 2P_2 \tag{1}$$

Again, another feasible solution is obtained by moving E_2 from s_j to s_i, and we have

$$P_1 + P_2 \leq 2P_1 \tag{2}$$

It is obvious that (1) and (2) cannot hold at the same time unless $P_1 = P_2$, which means that the cost of scheduling E_1 and E_2 into the same timeslot is equivalent to the cost of the best solution. Therefore, scheduling E_1 and E_2 into same timeslot must keep the best solution in the search space. □

The cost of the timetable obtained is minimized when combining exams that satisfy Conditions 1 and 2. This means that we can search for the best solution in a reduced space. However, Condition 2 is too strict to apply since it may be difficult to find two exams that satisfy this condition in real world exam timetabling problems. This condition needs to be relaxed and then developed into a suitable algorithm.

Consider the exam timetabling problem with m exams (E_1 , …, E_m) and n timeslots (s_1, …, s_n). Let c_{ij} denote whether or not there is clash between exams E_i and E_j,

$$c_{ij} = \begin{cases} 1 & \text{with clashes} \\ 0 & \text{without clash} \end{cases} \tag{3}$$

where $i, j \in \{1,\ldots,m\}$ and $i \neq j$.

Compatibility is defined to measure to what degree two exams are suitable to be combined.

$$C_{ij} = \begin{cases} \dfrac{1}{m-2}\displaystyle\sum_{k=0}^{m} s_{ij}(k) & \text{if } c_{ij} = 0 \\ 0 & \text{if } c_{ij} = 1 \end{cases} \tag{4}$$

where $i, j, k \in \{1,\ldots,m\}$, $i \neq j$, and $s_{ij}(k) = \begin{cases} 0 & \text{if } c_{ik} \neq c_{jk}, \text{or } k = i, \text{or } k = j \\ 1 & \text{if } c_{ik} = c_{jk} \end{cases}$.

Values of C_{ij} are ranged in $[0,1]$. $C_{ij} = 1$ denotes perfect compatibility between two exams. For example, the values of compatibility between E3, E4, E5, and E6 in example of Fig.1 are all 1, and they should be scheduled into the same timeslot. Small values of C_{ij} denote low compatibility and are therefore unsuitable for scheduling these exams into same timeslot.

In applying the measure of compatibility, we can predefine a value C_0 and combine those exams that satisfy $C_{ij} > C_0$.

Because the optimal solution does not necessarily remain in the reduced search space after combining two exams with $C_{ij} < 1$, there may be a trade-off between reducing computational times and the search for optimal solutions. Therefore, the value of C_0 may have a significant effect of any later search.

Example of Methodology

We have investigated applying this approach to the University of Toronto benchmarks, and found it is especially suitable for the St. Andrews83 (sta83-I) instance. This data has 139 exams, 611 students, 5751 enrolments, and 13 timeslots. The usual hard constraint is defined (i.e. exams with the same students should not be assigned to the same timeslot). A soft constraint is to minimize an evaluation function which denotes the cost of timetables that are generated. The average cost per student is used to compare different timetables. The evaluation function which calculates the cost of a timetable is presented in formula (5) below:

$$Cost = (\sum_{s=1}^{5} w_s N_s)/S \tag{5}$$

where w_s is the weight that represents the importance of scheduling exams with common students s timeslots apart, where $w_1 = 16$, $w_2 = 8$, $w_3 = 4$, $w_4 = 2$, $w_5 = 1$, and $w_s = 0$ for any $s > 5$. N_s is the number of common students involved in the violation of the soft constraint. S is the number of students in the problem.

Table 1 Compatibility matrix between 9 exams for sta83-I benchmark

	E1	E2	E3	E4	E5	E6	E7	E8	E9
E1	----	0.94	0.93	0.91	0.94	0.93	0.94	0.94	0.94
E2	0.94	----	0.93	0.91	0.94	0.93	0.94	0.94	0.94
E3	0.93	0.93	----	0.92	0.93	0.93	0.93	0.93	0.93
E4	0.91	0.91	0.92	----	0.91	0.91	0.91	0.91	0.91
E5	0.94	0.94	0.93	0.91	----	0.93	0.94	0.94	0.94
E6	0.93	0.93	0.93	0.91	0.93	----	0.93	0.93	0.93
E7	0.94	0.94	0.93	0.91	0.94	0.93	----	0.94	0.94
E8	0.94	0.94	0.93	0.91	0.94	0.93	0.94	----	0.94
E9	0.94	0.94	0.93	0.91	0.94	0.93	0.94	0.94	----

The measure of compatibility is expressed in the form of matrix so that all exams could be compared with each other. For example, the compatibility matrix between 9 exams is shown in Table 1. Because of zero clashes and high compatibility among them, these exams are suitable to be combined and scheduled into the same timeslot.

With S_0 = 0.91, a total of 10 groups of 84 exams was combined to form 10 new larger exams, and the number of exams decreased from 139 to 65. This greatly simplified the original timetabling problem. Heuristic ordering and local search methods have been applied to both the simplified problem and the original version. The results show speed ups in both convergence to feasible solutions and searching for optimized timetables.

Feasible Solutions Achieved by Heuristic Orderings

Different heuristic ordering methods are widely used in obtaining feasible solutions that can be further optimized by other meta-heuristics. The idea of these methods is that by estimating how difficult each exam will be to deal with, we can arrange the exams in a sequence from difficult to easy, and then the exams are scheduled according to their order in the sequence.

Some common used heuristic orderings include: Largest Degree (LD), Largest Colour degree (LC), and Saturation Degree (SatD). The details of these methods can be found in (e.g. Carter and Laporte (1996), Burke and Newall (2004)). Besides these, two other heuristic orderings, Largest Enrolment (LE) and Random ordering, are also used. These heuristic orderings were applied to both the new timetabling problem with a reduced number of exams and the original version. Backtracking is employed whenever there is a conflict.

Table 2 Results when using different initial orderings

Methods	Original problem		Simplified problem	
	Cost	Time (s)	Cost	Time (s)
LD initial order	184.03	0.30	159.43	0.02
LC initial order	173.05	0.40	159.22	0.01
SatD initial order	170.35	0.20	161.20	0.01
LE initial order	172.26	0.20	159.43	0.02
Random initial order	176.22	1.40	165.68	0.30
Average	175.18	0.50	160.99	0.07

The algorithm was coded in Visual C++ and experiments were run on a PC with dual 2.66GHz Intel CPU and 3.25GB RAM. The results in table 2 show that by reducing the number of exams it takes less time for the heuristic orderings to obtain better solutions. We do not attempt to compare our results in terms of the computational time with others in the literature because comparisons across different platforms are impractical.

Optimized Solution Achieved by Local Search

In order to check the influence of combining exams on meta-heuristics, a local search algorithm was developed to improve feasible solutions that were obtained by using graph ordering. The local search we use adopts a simple strategy that removes several exams from the feasible solution and reschedules them. If a better solution is found, the local search will restart based on the new timetable. Otherwise, several other exams will be tried. This process will continue until no further improvement can be made. The local search adopts a two-stage structure that reschedules two exams in the first stage and four exams in the second stage. It permits only those new timetables with lower cost into the second stage in order to decrease the computational time.

Table 3 Results when using local search

Methods	Original problem		Simplified problem	
	Time (s)	Best solution	Time (s)	Best solution
First stage	57,764	160.50	7,085	158.90
Second stage	85,020	159.10	11,257	157.08
Total	142,784	319.60	18,342	315.98

It took approximately five hours for the local search to reach a timetable with cost 157.08 (see Table 4) that was very close to the best solution so far (a timetable with cost 157.00 achieved by Cote et al (2005)). Without the initial combining of exams, the same local search did not reach any timetable with a cost below 159.10, and the computational time is also much longer. This result shows that the proposed approach can also be effective in accelerating meta-heuristics.

Table 4 A timetable for St. Andrews 83 benchmark with cost 157.08

Timeslots	Exams
1	5,6,7,8,30,31,32,33,34,35,36,37,39,40,41,42,43,44,45,46,47,120
2	38,59,60,61,62,63,64,65,66,67,68,69,70,71,131
3	1,2,28,135,139
4	50,51,52,53,54,55,56,57,72,95,96,100
5	4,85,134
6	98,99,101,102,132,138
7	48,77,78,79,80,81,82,83,84,130,133
8	18,19,20,21,22,23,24,25,26,29,49,97
9	136,137
10	58,86,87,88,89,90,91,92,93,94,103,104,112,113,114,115,116,117,118,119
11	27,105,106
12	3,17,109,110,111,121,122,123,124,125,126,127,128,129
13	9,10,11,12,13,14,15,16,73,74,75,76,107,108

Conclusions and Future Work

The success of meta-heuristic approaches has provided evidence of common features among those high quality solutions for a specific exam timetabling problem. The problem will be greatly simplified if these common features can be realised so as to limit the search space. The study in this paper demonstrates that simplifying the exam timetabling problem is possible. A criterion of compatibility is defined to measure how well several exams can be combined. By combining exams with relatively high compatibility, the quality of solutions remaining in the reduced search space could be guaranteed. The main drawback of the measure of compatibility is that it may not be suitable for every exam timetabling problem. There is the possibility that all compatibility measures are high (or low) because of very low (or high) density of clashes between exams. It would therefore be difficult to choose which exams to combine. However, compatibility might not be the only measure we can use to express the common features of high quality solutions. Our future work will focus on searching for general and practicable conditions that can be used to provide a general approach to simplifying exam timetabling problems. Reducing the number of exams in an exam problem can also help later heuristic and meta-heuristic methods and, potentially, this approach can also be applied in other scheduling problems (personnel timetabling, space allocation for example).

References

Balakrishnan N, Lucena A and Wong R (1992) Scheduling Examinations to Reduce Second-Order Conflicts, *Computers & Operations Research*, 19: 353-361

Burke E and Newall J (2004) Solving Examination Timetabling Problems through Adaptation of Heuristic Orderings Annals of Operations Research, 129: 107-134

Burke E and Newall J (1999) A Multi-Stage Evolutionary Algorithm for the Timetable Problem, *IEEE Transactions on Evolutionary Computation*, 3(1) 63-74

Burke E and Petrovic S (2002) Recent Research directions in automated timetabling, *European Journal of Operational Research*, 140(2): 266-280

Carter M (1983) A decomposition algorithm for practical timetabling problems. Technical Paper 83-06, Department of Industrial Engineering, University of Toronto.

Carter M and Laporte G (1996) Recent developments in practical examination timetabling. In: Burke E and Ross P (eds.) The practice and theory of automated timetabling: Selected papers from 1st International Conference on the Practice and Theory of Automated Timetabling, vol.1153, pp 3-21

Cote P, Wong T and Sabouri R. (2005) Application of a hybrid multi-objective evolutionary algorithm to the uncapacitated exam proximity problem, In: E.K. Burke and M. Trick (eds). (2005) Selected Papers from the 5th International Conference on the Practice and Theory of Automated Timetabling. Springer Lecture Notes in Computer Science, vol. 3616. 151-168

Gendreau M and Potvin J (2005) Tabu Search. In: E.K. Burke and G. Kendall (eds.) (2005). Introductory Tutorials in Optimisation, Decision Support and Search Methodology. Springer. Chapter 6, 165-186.

Glover F and Laguna M (1993) Tabu search. In: C.R. Reeves (ed.). Modern Heuristic Techniques for Combinatorial Problems. Oxford: Scientific Publications.

Kendall G and Mohd Hussin N (2005) A Tabu Search Hyper-heuristic Approach to the Examination Timetabling Problem at the MARA University of Technology, Lecture Notes in Computer Science 3616, pp 270-293

Nonobe K and Ibaraki T (1998) A Tabu search approach to the constraint satisfaction problem as a general problem solver, *European Journal of Operational Research*, 106: 599-623

Qu R and Burke E (2007) Adaptive Decomposition and Construction for Examination Timetabling Problems. In Multidisciplinary International Scheduling: Theory and Applications (MISTA 2007), (Baptiste P., Kendall G., Munier-Kordon A., and Sourd F., Eds.), 418-425

Qu R, Burke E, McCollum B, Merlot L and Lee S (2006) A Survey of Search Methodologies and Automated Approaches for Examination Timetabling, Computer Science Technical Report No. NOTTCS-TR-2006-4

White G and Chan P (1979) Towards the Construction of Optimal Examination Timetables, *INFOR*, 17 219-229

White G, Xie B, and Zonjic S (2004) Using tabu search with longerterm memory and relaxation to create examination timetables. *European Journal of Operational Research*, 153(16): 80-91.