10th International Conference of the Practice and Theory of Automated Timetabling
PATAT 2014, 26-29 August 2014, York, United Kingdom

Exam timetabling at Université de Technologie de Compiègne: a memetic approach

Taha Arbaoui · Jean-Paul Boufflet · Kewei Hu · Aziz Moukrim

1 Introduction and problem description

The exam timetabling problem at Université de Technologie de Compiègne (UTC) has some of the usual hard and soft constraints introduced in the second International Timetabling Competition ITC2007. For the sake of simplicity, we use the same terminology for these constraints in the sequel. The other constraints, however, fall into the scope of some of the potential extension of the ITC2007 problem (McCollum et al. 2012).

A timetable is considered as feasible by the practitioner if all exams are assigned to a room and a period while respecting the hard constraints. The quality of the solution is measured using soft constraints. Despite the allowance of scheduling exams in overlapping periods, examination rooms which are spread on different sites cannot be used twice at two overlapping periods. Moreover, rooms have a list of allowed periods.

Contrary to the ITC2007 problem, splitting exams between rooms is permitted. Thus, there are two types of exams: *splittable exams* and *non-splittable exams*. Each exam disposes of a list of allowed rooms and periods. As a result, for an exam to be assigned to room r and period p, they should both be allowed for the exam and room r must be available in the same period as well.

The hard constraints are the following:

- A student cannot sit two exams at the same period or at two overlapping periods.
- An exam must be assigned to a unique period.
- A non-splittable exam must be assigned to a unique room.
- The duration of the exam must be less than or equal to the duration of the period in which it is assigned.

Taha Arbaoui (✉) · Jean-Paul Boufflet · Kewei Hu · Aziz Moukrim
Université de Technologie de Compiègne
Laboratoire Heudiasyc, UMR 7253 CNRS 60205 Compiègne, France
E-mail: taha.arbaoui@hds.utc.fr
jean-paul.boufflet@hds.utc.fr
kewei.hu@etu.utc.fr
aziz.moukrim@hds.utc.fr

Instance	n^E	n^S	n^P	n^R	Conflict density
S2011	141	2322	32	8	0.30
F2011	119	2388	24	9	0.32
S2012	142	2412	36	9	0.31
F2012	117	2296	26	8	0.30
S2011noTC	122	1988	32	8	0.34
F2011noTC	102	2089	24	8	0.33
S2012noTC	123	2057	36	8	0.35
F2012noTC	100	2098	26	8	0.32

Table 1 Characteristics of UTC instances

- The capacity of any room should not be exceeded at any period.
- The sum of the parts of a splittable exam should be equal to the total number of enrolled students.
- A room can be used only at one period of a set of overlapping periods.
- Each exam should be assigned only once.

A solution is considered to be feasible when all the hard constraints are satisfied. On the other hand, when a soft constraint is not satisfied, a penalty is applied. The soft constraints used to measure the quality of the solution differ from one institution to another. A quick look at the ITC2007 benchmark shows that the Front Load penalty is not as important as the penalty for the *Two In a Row*, *Two In a Day* and *Period Spread*. However, this is not the case in our university. Due to the limited time given to professors after the exams to mark them, the practitioner informed us the most important penalty to minimize is the Front Load penalty (minimize the number of big exams planned at the end of the examination session). The following definitions describe briefly the soft constraints used by the practitioner:

Two In a Row: Examinations of a student allocated back to back in the same day should be avoided.

Two In a Day: Examinations of student scheduled in the same day but not back to back should be avoided.

Front Load: Large-size exams should be assigned before a certain period.

Note that ITC2007 differs in both the hard and the soft constraints. Our problem does not consider all the soft constraints used in the ITC2007 problem (*e.g.* Room and Period penalty). On the other part, ITC2007 lacks some of the hard constraints considered in our problem. For example, the overlapping periods and splitting the exams do not exist in ITC2007.

Table 1 presents the characteristics of four instances relative to four semesters in UTC. Column *instance* reports the labels where “S” stands for the spring semester and “F” stands for the fall semester. In UTC, the curriculum is split into two parts. The first part, called “Tronc Commun” (TC), is done in the first two years. The second part is a three-year specialization, at the end of which the student holds an engineering degree. Exams associated to the courses in TC usually constitutes a set of large exams. To measure the impact of these large exams, we studied another type of instances that does not contain the “TC” exams.

Columns n^E, n^S, n^P, n^R and *Conflict density* show for each instance the number of exams, the number of students, the number of periods, the number of rooms and the density of conflict graph, respectively.

2 The memetic algorithm

Evolutionary algorithms, such as Genetic Algorithms (Corne et al. 1994) and Memetic algorithms (Burke et al. 1996), have proved to be effective approaches to tackle exam timetabling problems (Qu et al. 2009). A memetic algorithm (MA) can be viewed as a genetic algorithm in which the key feature is the coupling of the global optimization and local optimization. The global optimization is managed by the crossover procedure while the local optimization is performed using the local search as mutation operators. We proposed a memetic algorithm to solve our problem.

Exams are labelled by integers and a *chromosome* is a permutation of these integers. A decoding procedure is needed to transform this *indirect encoding* into a solution. We used the *First Fit Decoding* (FFD) procedure, inspired from the Bin Packing first fit heuristic (Johnson 1974): exams in the permutation are taken in turn and assigned to the first period and room that respect the hard constraints. The decoding procedure does not always lead to a feasible solution. When it is the case, a *Repairing Method* (RM) is performed until a feasible solution is reached. The chromosome is then updated to the new permutation. The cost of a chromosome is assessed while decoding.

The population is initialized by generating fifty chromosomes at random. Five chromosomes are improved using a rapid destruction-construction local search. The idea is to remove randomly a number of exams, and to reinsert them using the following *Best Insertion* (BI) strategy: allocate an exam to the period and the room that minimize the penalty of the soft constraints.

We applied the Linear Order Crossover (LOX) (Dahal et al. 2007) on parents selected using the following Binary Tournament strategy: two couples of chromosomes are selected at random. The best of the first couple is the first parent and the best of the second couple is the second parent. LOX then randomly selects one of the parents and defines two indices on its permutation. Exams between the two indices are then given to the child and the rest of child is completed from the remaining parent.

The child is mutated with a certain probability. We used three different local searches as mutation operators. The operators are chosen at random. If an operator fails to improve the solution, one of the remaining operators is then applied. The mutation is stopped when none of the operators improves the chromosome.

The mutation operators used are Hill-Climbing (HC), Exam Swap (SWAP) and Light Destruction/Construction (LDC). A conflict graph in which nodes are exams and edges represent incompatibilities between exams is used for Light Destruction/Construction. We briefly describe the mutation operators as follows:

- HC: a random order of exams is considered. Next, exams are removed one after another in the defined order, and then inserted back in the solution using BI.
- SWAP: two periods are randomly selected. The exams assigned to these periods are swapped. The new solution is accepted *iff* all the hard constraints are respected and the quality of the solution is improved.
- LDC: a random exam is removed from the solution. Some of its adjacents in the conflict graph are also removed. These exams are then shuffled and reinserted using BI. If some exams are left unscheduled, the last best permutation of the current chromosome is restored and LDC is reapplied.

Instance	MA	Previous approach	noTC
P2011	5462	10926	3896
A2011	3400	7501	2020
P2012	3450	12008	2586
A2012	3241	9266	1671

Table 2 Results obtained by MA. "noTC" presents the results of MA when the exams of "Tronc Commun" are removed.

The population is updated by replacing an existing chromosome having the same quality of the solution or by replacing the worst chromosome of the population if the new chromosome is better.

3 Preliminary results

Table 2 presents the preliminary results obtained using the memetic algorithm. Results show that MA improved the quality of solutions compared to the previous approach used by the practitioner. The cost of the solutions has been more than halved.

In order to help the practitioner determine the exams involved in most of the solution penalty, we removed the exams of the core curriculum (TC) to measure their impact. The results show that "TC" exams highly contribute to the cost of the solutions on the different instances. This allows us to investigate a two-stage approach aiming at giving more priority to "TC" exams in the planning. In fact, they share students with many other exams which leads to a bigger conflict in the examination session. Taking these exams in priority will effectively lead to high-quality solutions.

References

Edmund K. Burke, James P Newall, and Rupert F Weare. A memetic algorithm for university exam timetabling. In *Lecture notes in computer science: Practice and Theory of Automated Timetabling I: selected papers from the 1st international conference*, volume 1153, pages 241–250. Springer, 1996.

Dave Corne, Peter Ross, and Hsiao-Lan Fang. Evolutionary timetabling: Practice, prospects and work in progress. In *UK planning and Scheduling SIG Workshop*, 1994.

Keshav P. Dahal, Kay Chen Tan, and Peter I. Cowling, editors. *Evolutionary Scheduling*, volume 49 of *Studies in Computational Intelligence*. Springer, 2007.

David S. Johnson. Approximation algorithms for combinatorial problems. *Journal of Computer and System Sciences*, 9(3):256 – 278, 1974.

Barry McCollum, Paul McMullan, Andrew Parkes, Edmund K. Burke, and Rong Qu. A New Model for Automated Examination Timetabling. *Annals of Operations Research*, 194: 291–315, 2012.

Rong Qu, Edmund Burke, Barry McCollum, Liam T. G. Merlot, and S. Y. Lee. A Survey of Search Methodologies and Automated System Development for Examination Timetabling. *Journal of Scheduling*, 12(1):55–89, 2009.