

DM87 - Scheduling, Timetabling and Routing

Exam Project, Fall 2008

1 Practical Notes

Note 1 The project has to be carried out in groups of maximum two persons. It is not permitted to collaborate between groups.

Note 2 The project consists in modeling a real-life problem, designing a solution algorithm, implementing the solver, experimenting on some instances that are real and simulated data and documenting the work in a report.

The evaluation of the project is based on the written report that may be in English or in Danish. A program that implements the solver described in the report must also be delivered. The program will serve to check the correctness of the solutions discussed.

Note 3 Additional material to the project description is available on the web at: <http://www.imada.sdu.dk/Courses/DM87/index.html#exam>. Corrections or updates to the project description or to the web page above will be announced by email to the students through the email address they registered in the Blackboard system.

Note 5 *Submission*. Two printed copies of the written report must be handed in at the secretary office **before 12:00 of Monday, June 10, 2008**. Ask the secretary for a receipt showing that you have handed in the report in time. Contextually, an electronic version of the report and the program source code must be sent by email to the lecturer. A reply will be sent as receipt.

Reports and codes handed in after the deadline will generally not be accepted. System failures, illness, etc. will not automatically give extra time.

2 Problem Description

Course timetabling (CTT) consists in the weekly scheduling of the lectures of a set of courses within a given number of rooms and time periods, satisfying various constraints due to conflicts and other features.

For the curricula of Mathematics and Computer Science at the University of Southern Denmark the course timetabling is constructed in two stages. First, the mandatory courses are scheduled by the Faculty of Science and, then, the elective courses are scheduled by the Department of Mathematics and Computer Science. In both cases, the scheduling of the courses takes into account the *student enrollment data*.

This project focuses on the timetabling of the elective courses at the Department of Mathematics and Computer Science (IMADA). We assume to have available the student enrollment data to all the courses and the Faculty timetable for the mandatory courses. This latter determines unavailabilities of the students and teachers in certain periods.

In the following, we present the entities and data needed to model the IMADA timetable problem (IMADA-CTT).

Days, Periods, and Timeslots There are 5 teaching *days* in a week (from Monday to Friday). Each day is split in 5 *class periods* of 2 hours each, starting at 08:00 and finishing at 18:00. A *timeslot* is a pair composed of a day and a period. The total number of scheduling timeslots is 25.

Courses and Lectures Courses are divided into *elective* and *mandatory* courses. Each course consists of a fixed number of lectures to be scheduled in distinct timeslots, it is attended by a given number of students, and it is taught by a teacher.

The timeslots of lectures from mandatory courses are fixed and cannot be changed.

In order to spread or keep close lectures of the same course according to teacher's preferences, there is a minimal and a maximal *day distance* between lectures (*min_sep*, *max_sep*).

Student Enrollments Students are enrolled to courses and can attend different combinations of them.

Teachers Each course, elective or mandatory, has a teacher. There might be courses that are taught by the same teacher. A teacher can be unavailable for certain timeslots.

Rooms Each room has a capacity, expressed in terms of number of available seats. Some rooms may not be suitable for some courses (because they miss some equipment, for example, a beamer projector). In the specific case of IMADA-CTT, we have only one room available and suitable for all courses: this is the IMADA's seminar room. However, if needed, other rooms can be found in the Campus. We use the term *dummy room* to identify any such other room and we assume that a room from the set of dummy rooms is suitable for any course.

There might be some timeslots in which a room is unavailable.¹

These data are contained in a series of text files that compose a specific instance of the IMADA-CTT problem. See the Appendix for specifics on the organization and format of these input files.

A *timetable* is an assignment of a timeslot (day and period) and a room to all lectures of each course. A timetable should obey the following hard constraints:

1. Only one lecture is assigned in one room in a timeslot.
2. Lectures must only be assigned to rooms in timeslots when they are "available".
3. No teacher has to teach two lectures in the same timeslot or has to teach in a timeslot in which he declared to be "unavailable".
4. No student has to attend more than one lecture at the same time.
5. The room should be big enough and suitable for the lecture assigned.

A timetable is defined as:

¹For example, no lecture can be scheduled in the IMADA seminar room when there is the Computer Science Colloquium or the Mathematics Colloquium. Note that it is possible also to have no teacher lecturing at that time by setting them unavailable at the time of the colloquia.

- **valid** if *all* the lectures are scheduled, there are no occurrences of any hard constraint violations, but some of the lectures are assigned to rooms from the set of dummy rooms.
- **feasible** if *all* the lectures are scheduled, there are no occurrences of any hard constraint violations, and no dummy room is used.

Note that only elective courses and their interaction with the mandatory courses has to be taken into account in the IMADA timetabling. Therefore, violations of any hard constraint that are due only to mandatory courses do not count as violations. This is because nothing could be done to repair such violations in the timetabling at IMADA.

In addition to the hard constraints given above, we are also interested in satisfying a number of soft constraints.

1. Lectures should not be scheduled in the first and last timeslot of a day.
2. Lectures of the same course should be spread or kept close according to teacher's preferences. To this end lectures should be separated by a day distance within the minimal and the maximal value specified.²
3. Students should not be required to attend more than three lectures in a day.

The task is to find a timetable that is feasible and that minimizes the violations of soft constraints.

3 Project Content

The aim of the project is to design and implement an automatic solver for the IMADA-CTT problem. In a first stage, the use of exact methods has to be considered either for solving the overall problem or the feasibility problem. If exact methods result infeasible, because it is impossible to model some of the constraints or because they fail to return a solution within a reasonable amount of time, say 1 or 2 hours, then heuristic methods can be used. Solutions are evaluated as described next.

Timetable Evaluation

All solutions must be valid. It is permissible for some lectures to be scheduled in the dummy rooms (these lectures could be scheduled later by using some of the rooms left free at the Campus). If a solution is valid the number of lectures scheduled in the dummy rooms are used to compute the DISTANCE TO FEASIBILITY measure. This is calculated by identifying the number of students that are required to attend each of the lectures scheduled in the dummy rooms and adding these values together. Thus if for example, a solution has two lectures scheduled in the dummy rooms and the number of students attending these is 12 and 8, then the DISTANCE TO FEASIBILITY is simply $(12+8)=20$. A feasible timetable, by definition, has a DISTANCE TO FEASIBILITY of zero.

The cost due to the soft constraint violations, named SOFT COST, is then calculated as follows:

²For example, courses that want two lectures close each other might have $\text{min_sep} = 0$ and $\text{max_sep} = 1$ while courses that want lectures more uniformly spread over the week might have $\text{min_sep} = 2$ and $\text{max_sep} = 4$.

Instance	Lectures	Students	Teachers	Rooms
E04	14	56	11	1
F05	22	67	12	1
E05	16	78	11	1
F06	20	66	10	1
E06	20	58	10	1
comp-2007-2-7	109	500	180	20
comp-2007-2-8	100	500	180	20
comp-2007-2-15	87	500	180	10
comp-2007-2-16	93	500	180	10

Table 1: Statistics about the instances. The first 5 are the IMADA instances and they have only one room available (IMADA’s seminar room). The last 4 are the LARGE instances and they have 10 to 20 rooms available. If needed in both classes, dummy rooms can be added.

- count the number of occurrences of a student having a lecture in the first or the last timeslot of the day. (Mandatory courses *do not* count.)
- count the number of occurrences of a student having to attend two lectures of an elective course that are separated less than their minimal day distance or more than their maximal day distance.³
- count the number of occurrences of a student having more than 3 lectures in a day. (Mandatory courses count as well.)

The SOFT COST of the timetable is the total of these four values.

The quality of a valid timetable is therefore reflected by a pair of values: (DISTANCE TO FEASIBILITY, SOFT COST). In order to compare two timetables and decide which is the best the following procedure should be applied.

First, examine the DISTANCE TO FEASIBILITY of the two timetables. The timetable with the lowest value is deemed the best. If the two timetables are equal in this respect, then the SOFT COST values are to be examined. The timetable with the lowest SOFT COST is deemed the best.

Instances

There are two sets of instances. One set, IMADA, are historical data of the real-life problem at IMADA. Another set, LARGE, are data simulated for a larger problem. The instances are composed of a set of files and grouped in directories. They are downloadable from the project web page. See Appendix for organization and format of these files. Table 1 summarizes some statistics about these instances.

To simplify the problem there are no restrictions to the suitability of the rooms. All rooms are therefore suitable for all courses, the hard constraint 5 is always satisfied and it can be ignored.

³The day distance between two lectures is always computed through the working days and not through the weekend.

On all the instances, it is not guaranteed that a feasible timetable with SOFT COST equal to zero exists. However, on the IMADA instances, it is expected that feasible timetables exist and that it should not be too hard finding them. In the past it has been possible to accomplish this operation by hand often in just a quarter of an hour. On the LARGE instances, even a feasible solution might not be guaranteed to exist because these instances have been generated at random and never yet solved.

In case of solutions not feasible, it might be worth recalling that it is not the number of lectures scheduled in dummy rooms that is important for the measure DISTANCE TO FEASIBILITY, rather it is the *number of students* scheduled in these lectures. This is done in order to satisfy as many people as possible within the timetable.

4 Remarks

Remark 1 In the report there must be at least one numerical table resuming the results in terms of quality of timetables and computational effort. For heuristic methods a time limit of about 5 minutes must be imposed. The time limit can be determined independently from the machine using the benchmark program available at the project web site. If the solver is stochastic, at least the best result and the average result for each instance should be reported (a boxplot might also be helpful to visualize the data more comprehensively). For the best solution it should be indicated the random seed that was used to attain that solution. This will be used to check the results reported. Both the IMADA instances and the LARGE instances should be included in the table.

Remark 2 The evaluation of a solution can be verified by means of the program checker made available from the web page of the project. This program will be used by the lecturer to verify the results presented in the report. The source code of this program is available. Please report to the lecturer, if discrepancies with what stated in this document are found,

Remark 3 The quality of the timetables achieved will be a primary factor in the evaluation of the project. In order to discourage cooperation between groups, the final grades will reflect as much as possible the ranking of the solutions presented. Results on the IMADA instances are more important than those on the LARGE instances.

Remark 4 In case the solution is found by exact methods, the source code of the model must be delivered and it must be stated which program is used to solve it (e.g., CPLEX, SCIP, GECODE, etc.). For heuristic algorithms, the programs must output the solution in a reasonable amount of time. Solutions must be written in a file in text format as described in the Appendix. The programs will be run by the lecturer. It is student's responsibility to make sure that they can run on IMADA machines with the following command line:

```
ctt instance_dir solution_file time_limit
```

Remark 5 The total length of the report should not be less than 10 pages and not be more than 15 pages, appendix included (lengths apply to font size of 11pt and 3cm margins). Although these bounds are not strict, their violation is highly discouraged. Do not include source code in the report.

Acknowledgments This document has been prepared with the help of Rolf Fagerberg who has been responsible for the IMADA timetabling in the last years.

Appendix A Instance Format

There are 5 instances for the IMADA class and 4 instances for the LARGE. Each instance is composed by 7 files. In the following we describe the content and the format of these files.

`electives.dat` list of elective courses and number of lectures to schedule in a week.

Example:

```
DM71  2
DM73  2
DM74  2
...
```

`mandatory.dat` list of lectures for mandatory courses with the scheduled day and start time. Example:

```
DM01   T       8
DM01   W       14
DM01   W       16
DM14   M       10
DM14   W       14
...
```

M = Monday; T = Tuesday; W = Wednesday; H = Thursday; F = Friday

`teaches.dat` list of teachers and courses they teach. Example:

```
NJN    MM504
AS      MM08
KM      DM504
...
```

`teacher_unavailable.dat` list of teachers with day and time when they are unavailable. (The file might be empty if no teacher unavailability exists.) Example:

```
MC     T       8
MC     F       8
MC     F       16
...
```

`takes.dat` list of students and their course enrollments. Example:

```
STD-0001  AI02
STD-0002  AI02
STD-0003  BM119
STD-0004  DM/DMP51
...
```

rooms.dat list of rooms available. Example:

```
Seminarrum
Audimax
...
```

room_unavailable.dat list of room unavailabilities. Example:

```
Seminarrum T 14
Seminarrum H 14
...
```

The following data must be assumed implicitly:

Room assignment for mandatory courses: we assume all mandatory courses to be scheduled in different rooms from those available, that is, in dummy rooms.

Minimal day distance: 2.

Maximal day distance: 4.

Appendix B Solution Format

In order to check the validity of the results reported, the program submitted must output the solution in a textual file when finishing.

In each row of the file, separated by a tab character, there is the name of the course followed by the identifier of the timeslot and the identifier of the room in which it is scheduled. The identifier of the timeslot is a number $t = 0, 1, 2, \dots, 24$, where 0 represents the first class period on Monday and 24 the last class period on Friday, while the room is a number $r = 0, 1, \dots, n_rooms - 1, n_rooms, n_rooms + 1, \dots$ where values in $0, 1, \dots, n_rooms - 1$ represent the allowed rooms and values $n_rooms, n_rooms + 1, \dots$ represent dummy rooms.

```
DM71 2 0
DM73 3 1
DM74 10 0
...
```