

FIT5216: Modelling Discrete Optimization Problems

Inclass Task 14: Exam Study Timetable

1 Problem Statement

You have a number of exams to study for and only D days remaining before the exam period begins each with H hours (an even number) available for study, split into a morning session and afternoon session. The requirement is to create a study plan that starts as late as possible, so you can slack off as long as possible.

The constraints of the problem are:

- There are set of topics to study TOPIC.
- Each topic t requires $hours[t]$ continuous hours of study (although you can start one afternoon, and continue the next morning; or start)
- Some topics must be completed, before others are begun.
- Some topics have to be started first thing in the morning when you are freshest.
- Some days only the morning is available since you have sporting commitments in the afternoon.
- Each topic must be completed in a single day

Data for the problem is defined as follows:

```
enum TOPIC;
array[TOPIC] of int: hours;    % hours study per topic
int: m;                        % no of precedences
set of int: PREC = 1..m;
array[PREC] of TOPIC: before; % topic that must be finished before
array[PREC] of TOPIC: after;  % topic after starts
set of TOPIC: morning;       % which topics must start at morning
int: D;                       % number of days
int: H;                       % hours per day (even)
constraint assert(H mod 2 = 0, "Hours per day H must be even\n");
set of int: DAY = 1..D;
set of DAY: HALF;             % days where only morning is available
```

Build a MiniZinc model `examstudyp.mzn` The main decision are

```
set of int: HOUR = 0..H-1;
array[TOPIC] of var DAY: day;
array[TOPIC] of var HOUR: hour;
```

The aim is to maximize the earliest time for when any subject is first studied

For example a small dataset is

```

TOPIC = { BMATHS, AMATHS, ENGLISH, CHEM, PHYS };
hours = [2,5,4,3,6];
m = 2;
before = [BMATHS,CHEM];
after  = [AMATHS,PHYS];
morning = {AMATHS,CHEM};
D = 5;
H = 8;
HALF = {3,4};

```

for five topics over 5×8 hour days, where the afternoon of days 4 and 4 is unavailable.

One possible solution is `day = [1, 2, 4, 3, 5]`; and `hour = [5, 0,0,0,0]` where we start at hour 5 on day 1. The schedule is visualized as

	01234567
day 1	bb
day 2	aaaaa
day 3	ccc
day 4	eeee....
day 5	ppppp

The MiniZinc model should make use of packing constraints to solve the problem!

2 Instructions

Edit the provided `mzn` model files to solve the problems described above. Your implementations can be tested locally by using the *Run* icon in the MINIZINC IDE or by using,

```
minizinc ./modelname.mzn ./datafile.dzn
```

at the command line.