A modeling project: Schedule your department's courses with Pseudo Boolean Solvers (or 0-1 ILP solvers)

In this project you are requested to formulate the problem of scheduling the courses offered by your department. You are welcome to adjust the constraints to those imposed by your university / department. You are welcome to invent your own input, but ideally you should feed the system with real data.

As input your program receives the following data:

1. A list of the form :
   Course ID, weekly lecture hours, weekly recitation hours, weekly lab hours, lecturer ID, TA ID (possibly more than one TA)

2. Sets of courses that cannot be scheduled in the same slot. With these constraints we can make sure that the courses on the recommended list of courses for some population (say, 3rd-semester software engineering students) do not clash.

3. Partial schedule. With these constraints we can handle a situation in which the schedule of some courses is determined a priory by other departments etc.

4. Constraints imposed by the lecturers / TAs that specify when they *cannot* teach. These constraints are given in the form:
   ID, day, hour

5. Scheduling parameters:
   A. # hours per day (default = 10),
   B. Minimum block size (default = 2). The meaning of this parameter is that courses that require two or more hours a week will be scheduled in 2-hours blocks.

The objective: find a schedule with a minimum number of 'holes'.

The program should output the constraints in the input language of your favorite Pseudo-Boolean solver (PBS) or your favorite 0-1 ILP solver.
It should invoke the solver, read its output and then translate it back into a readable format. Specifically it should display a list of the form
course, day, hour
and a similar list for each of the sets defined in item 2 above.