15-780: Graduate AI

Homework Assignment #2

Out: February 12, 2015
Due: 5 PM February 25, 2015

Collaboration Policy: You may discuss the problems with others, but you must write all code and your writeup independently.

Turning In: Please email your assignment by the due date to `shayand@cs.cmu.edu` and `vdperera@cs.cmu.edu`. Make sure your solution to each problem is on a separate page. If your solutions are handwritten, then please take photos or scan them and make sure they are legible and clear. Please submit your code in separate files so we can easily run them.

1 Cryptoarithmethic

Solve the cryptarithmethic problem shown in Fig. 1 by hand, using the strategy of backtracking with forward checking and the MRV and least-constraining-value heuristic.

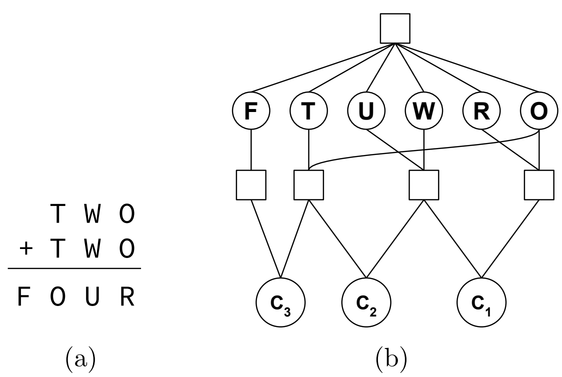

Figure 1: (a) The cryptarithmethic problem. (b) The constraint hypergraph.

In a cryptarithmethic problem each letter stands for a distinct digit; the aim is to find a substitution of digits for letters such that the resulting sum is arithmetically correct, with the added restriction that no leading zeros are allowed.

To help you out in Fig.1 you can see the constraint hypergraph for the cryptoarithmethic problem, showing the *Aldiff* constraint (square box on top) as well as the column addition constraints (four square box in the middle). The variables C_1,C_2 and C_3 represent the carry digits for the three column.

An *Aldiff* constraint is a global constraint which says that all of the variable involved in the constraint must have different value.

2 Scheduling Nightmare

As an overworked student, you must complete all of your homework assignments in each of your classes before they are due at the end of the semester (at time $t = T$). Each homework assignment has a specific duration $d \in \mathbb{N}$ representing how long it will take to complete. You must also complete them in order (so the first homework assignment in a class must be completed before the second in that same class); however, homework assignments in one class can be completed in any order relative to assignments in another class.

Since you are a student in computer science, many of your homework assignments require the use of specific machines for experimentation. You have a set of k computers; some homework assignments require one or more specific computers, and occupy them entirely for the entire duration of the homework assignment (i.e., if both homework A and homework B require computer 1, then you cannot work on A and B at the same time). Otherwise, you can work on multiple homework assignments at the same time, as long as they are not from the same class.

An Example Semester

Fig. 2 gives an example semester-long homework schedule for a student. The student has four classes, and each class has either three or four homework assignments. These homework assignments have (i) a duration in time periods, (ii) a relative ordering to other homework assignments in the same class, and (iii) an optional resource requirement. We assume the semester is 15 time periods long.

For example, the homework $H_{0,1}$ has duration 4 and requires resource 0. Similarly, homework $H_{1,0}$ has duration 1 and also requires resource 0. As discussed above, the shared resource constraint would prevent the student from overlapping these two homework assignments in her schedule.

Please use the example from Fig. 2 in these problems.

1. What are the variables and their domains?

2. Describe how you would write this problem as a CSP with arity $r > 2$, e.g., with at least some constraints having at most r variables.

3. Qualitatively, how would this change if you were limited to arity $r = 2$? Recall from class that any CSP can be written this way.

4. Solve your scheduling CSP—either the $r > 2$ or $r = 2$ version—by hand or with code using (i) a deterministic ordering of variables and values (e.g., alphabetical order) and

(ii) the most constrained variable and least constraining value heuristics discussed in class. How much of an effect did these heuristics have on the size of the search tree? Quantify your answer.

5. The problem above is solved relatively easily with simple heuristics and backtracking search. This is not always the case. Come up with a short "worst-case" instantiation of the general scheduling problem. What are some properties of these search problems that will thwart the most constrained variable heuristic? The least constraining value heuristic?

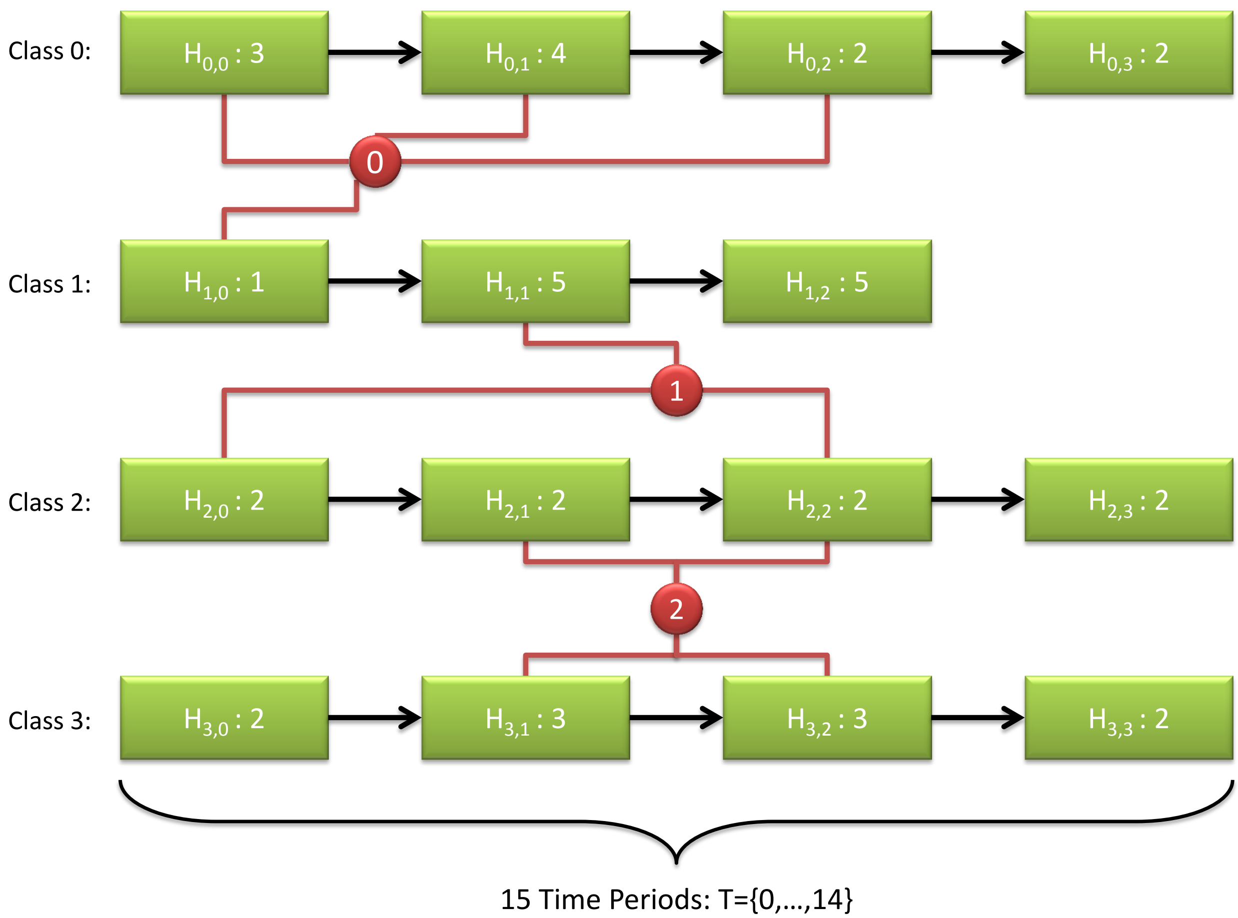

Figure 2: A semester's worth of homeworks for four classes. Each homework has a relative ordering (e.g., $H_{0,0}$ must be completed before $H_{0,1}$), an integral duration, and (possibly) a resource constraint. The semester is assumed to be 15 time periods in length.

3 Planning

1. GraphPlan can backtrack during its second phase while searching backwards from the goals through the planning graph. Explain, in words and with a simple example, why GraphPlan needs to backtrack in its backwards search.

2. If GraphPlan cannot find a solution during the backwards search, does this mean that the problem is not solvable? If so, explain why. If not, explain what GraphPlan can do to find a solution in such cases.

3. It is possible for GraphPlan to terminate after finding an n time step plan of k operators, while there actually exists an n time step plan with less than k operators in the planning graph. Show a concrete example of this and explain why, in general, this can occur.

4. FF performs forward heuristic search using as an heuristic the number of levels to the goal of the relaxed GraphPlan graph (no deletes). Is this heuristic admissible? Explain why, and if not give a counterexample, i.e., show an example for which FF does not find the optimal solution.

5. **Extra credit:** We have seen in class that linear planning using a stack of goals and means-ends-analysis was incomplete (remember the one-way-rocket domain). Can you find an example for which nonlinear planning (i.e., using a set of goals, instead of a stack of goals) with means-ends-analysis is incomplete?

4 Q-Learning

A robot moves deterministically in a world of 12 states laid out as shown in the Fig. 3. The robot can take four actions at each state, namely N, S, E and W. An action against a wall leaves the robot in the same state (note: not all walls are shown). Otherwise, the outcome of an action is deterministic (e.g. if a robot takes action N and it does not hit a wall, it will always end up in the state above it). The robot converged to the Q-table shown in Fig. 4,

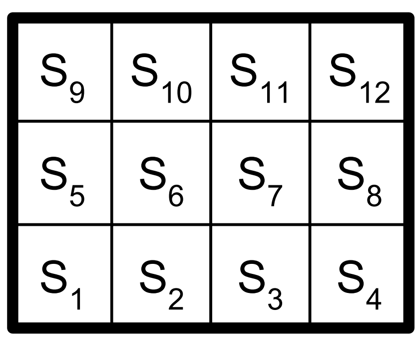

Figure 3: The robot world

where we show only the maximum values for each state, as the other values (represented with a dash) do not matter in terms of the final policy.

1. The robot uses this learned Q-table to move in the world. Write the sequence of actions that the robot takes and states reached in three different episodes (of six actions each) starting in three different start states, namely S1, S6 and S7.

   Note: If there is more than one best action available, choose one randomly.

	N	S	E	W
S1	60	-	60	-
S2	-	-	66	-
S3	-	-	73	-
S4	81	-	-	-
S5	66	-	-	-
S6	-	-	59	59
S7	-	66	-	-
S8	90	-	-	-
S9	-	-	73	-
S10	-	-	81	-
S11	-	-	90	-
S12	100	-	100	-

Figure 4: Q-table

State	Action	State	Action	State	Action	State	Action	State	Action	State	Action	State
S1												
S6												
S7												

You are told that the discount factor for Q-learning is $\gamma = 0.9$. Based on this knowledge and the learned Q-table, answer the following questions:

2. A friend of yours, Chris, tells you that there is a single state s, such that $r(s, a) > 0$, for all actions a. What is that state and what is the reward?

3. Chris also tells you that the robot world has interior walls, besides the outside walls shown in the picture above. Where are those walls?

5 8-Queens

1. For the chess board in Fig. 5, the red queens are attacking each other. Show two levels of the CSP local search using the min-conflict heuristic, justifying your choice of the queens to move.

2. Discuss the relationship between the MRV and min-conflict heuristics.

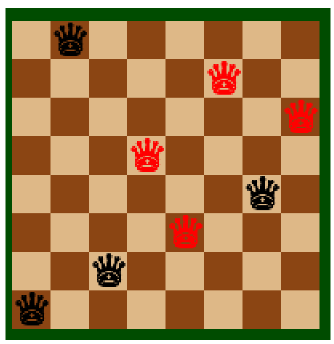

Figure 5: An 8-queen problem.