Graduate Artificial Intelligence 15-780

Homework #2: *Informed Search & Planning*

Out on February 13
Due on February 27

Problem 1: Planning [60 pts]

I. STRIPS

Using the basic grounded STRIPS, write a planning domain to solve a specific planning task of your choice.

a) The domain should have at least 4 types of operators.

b) Show two versions of the domain with different sets of operators.

c) Run a planner of your choice in these two different versions. Wesley Tansey from the University of Texas at Austin has posted an easy-to-use STRIPS planner (written in Python) here: https://github.com/tansey/strips.

Please submit a printout of both versions of your domain, printouts of two problems in your domain (i.e., the initial state and goal state that define each problem), and printouts of the output of your choice planner for both these problems each using both versions of the domain. Discuss the main choices you made in representing the state space and the operators, and discuss how the different versions of the domain affect the efficiency of solving problems. **20 pts**

II. Partial-order planning

Partial-order planning backtracks over which action to use to achieve an open condition and which method to use to resolve a threat (flaw). It does not, however, backtrack over the order in which to achieve open conditions, nor the order in which to resolve threats. For each of the questions below, please explain your answers precisely and thoroughly.

a) Why doesn't the fact that open conditions can be achieved in any order affect the completeness of partial-order planning? **5 pts**

b) Why doesn't the fact that threats can be resolved in any order affect the completeness of partial-order planning? **5 pts**

c) Give, and explain, an example of how the order in which open conditions are achieved affects the efficiency of partial-order planning. **5 pts**

d) Give, and explain, an example of how the order in which threats are resolved affects the efficiency of partial-order planning. **5 pts**

III. GraphPlan

a) 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. **10 pts**

b) If GraphPlan cannot find a solution during the backwards search, does this mean that the problem is not solvable? Carefully justify your answer. **5 pts**

c) It is possible for GraphPlan to terminate after finding an n time step plan of a total of k operators, while there actually exists an n time step plan with less than k operators in the planning graph? If your answer is positive, show a concrete example and explain the general case. If your answer is negative, explain why not? **5 pts**

Problem 2: Search & Integer Programming [40 pts]

In the United States, any individual wanting to broadcast using radio waves must first acquire an operating *license* for a certain frequency range in a certain geographic region. You were recently hired by the Federal Communications Commission (FCC), the regulatory body in charge of the spectrum, to conduct an auction to sell licenses to potential buyers.

The wireless spectrum is discretized into m disjoint segments of frequency range, $\{\ell_1, \ell_2, \ldots, \ell_m\}$. Each of the n potential buyers b express a *valuation* for combinations of these frequency ranges through `XOR` bids, an exclusive disjunction over conjunctions of spectrum segments. An example bid by bidder i is:

$$\langle (v_1^b, \ell_1 \wedge \ell_3) \oplus (v_2^b, \ell_6) \rangle$$

In this example, the bidder announces a valuation of v_1^b for ownership of both segments ℓ_1 and ℓ_3, and a valuation of v_2^b for ownership of segment ℓ_6. The $\oplus$ constraint between the spectrum conjunctions enforces the rule that only *one* of these conjunctins can be one by the bidder.

Solving the winner determination problem

The *winner determination problem* (WDP) is that of finding an allocation of items (in this case, spectrum licenses) to bidders that maximizes the auctioneer's revenue. The auctioneer's revenue is maximized by choosing an allocation that maximizes the sum (over all bidders) of the bidders' valuations for the subset of items they receive.

a) Formulate the WDP for the FCC spectrum auction as an integer linear program (IP). **5 pts**

Finding a solution to this integer linear program is (NP-)hard.However, you can quickly—i.e., in polynomial time—provide loose *upper* and *lower* bounds on the objective value of the integer program.

b) Formulate an *upper bound* for the problem, and prove that it is an upper bound. For this problem, please formulate an upper bound other than the LP relaxation. **5 pts**

c) Formulate a *lower bound* for the problem, and prove that it is a lower bound. **5 pts**

One way to solve the integer program defined above is by using a *branch-and-bound* tree search algorithm with A* node selection. This technique uses the divide-and-conquer paradigm to split the search problem into subproblems by assigning a value to each integer/binary variable at each node of the search tree.

d) Write a branch-and-bound solver that, given an instance of the FCC spectrum auction, solves the winner determination problem. This solver should use the upper and lower bounds you designed to prune the search tree at every node. **15 pts**

e) Compare the solution speed (runtime and search tree size) of your solver using (i) your upper bound and (ii) the LP relaxation of the subproblem at each node of the search tree. You can—and should—use a standard LP solver, many of which are available online. **10 pts**

Suggested LP solvers

Writing your own linear programming solver is a tricky task due to the ease in which numerical instability issues arise. We recommend you don't write your own solver for this homework; instead, check out one of the following:

Solver	Type	Download	Tutorial	Notes
CPLEX	Free (Academic)	Link	Link	Requires a (free) IBM Academic account
Gurobi	Free (Academic)	Link	Link	Requires a (free) Gurobi Academic account
CLP	Free	Link	Link	Very commonly used open source LP solver
SCIP (SoPlex)	Free	Link	Link	LP solver for the other big open source IP suite
lpsolve	Free	Link	Link	Slower than CLP/SoPlex, but the easiest to install

Note that many of these solvers can also directly solve integer linear programs. Please write your own branch-and-bound solver, as detailed above, but feel free to check your answers against any of these solvers (for correctness)!

Testing your solver

Your solver should accept auction data in the following form:

```
<num-segments>   <num-bidders>
<bidder-id>      <valuation>     <segment-id>   ...   <segment-id>
<bidder-id>      <valuation>     <segment-id>   ...   <segment-id>
     ⋮                ⋮                ⋮          ⋮       ⋮
<bidder-id>      <valuation>     <segment-id>   ...   <segment-id>
```

The first line represents m, the number of spectrum segments available for allocation and n, the number of bidders interested in the segments. The next lines specify one conjunction from a bidder's overall `XOR` bid. Note that everything is 1-indexed, not 0-indexed.

For example, a simple two-item auction with three bidders could like like this:

```
2   3
1   1.12   1
1   1.99   1   2
2   2.3    1
3   0.82   2
```

Here, bidder b_1 submitted bid $\langle(1.12, \ell_1) \oplus (1.99, \ell_1 \wedge \ell_2)\rangle$, while b_2 submitted bid $\langle(2.3, \ell_1)\rangle$ and b_3 submitted bid $\langle(0.82, \ell_2)\rangle$. The revenue maximizing allocation gives ℓ_1 to b_2 and ℓ_2 to b_3, for total revenue of 3.12.

Like with the last homework, we will generate some test instances and post them online. We will also release an instance generator with which you can more extensively test your code. These will be posted on the course website soon.

As output, your solver should write your Andrew ID and the revenue from a revenue maximizing allocation to standard out. That is, if you are Harry Q. Bovik with Andrew ID `hqbovik` and determine a revenue maximizing allocation with objective value of 12.355, your final output to standard out should be:

```
hqbovik   12.355
```