

Homework 9

OpenNERO—Extending Planner

Due: Thursday, Nov. 7th, at beginning of class (11:10AM).
(Submit to Blackboard by due date/time.)

[Everyone, for steps 1-3] Extending State-Space Search Planner using OpenNERO

You must work with the team arrangement you “declared” in Part 1 of HW 5, or alone.

1. Read the OpenNERO discussion of Symbolic Planning, here:
<https://code.google.com/p/opennero/wiki/SymbolicPlanning>

(nothing to turn in)

2. Extensions to the above planning approaches are discussed; read about them here:
<https://code.google.com/p/opennero/wiki/PlanningExercise>

(nothing to turn in)

3. **Extending State-Space Search Planner:** For this assignment, you will implement *one* of the suggested extensions – namely, the second one which extends the State-Space Search method, referred to as “Generating Optimal Plans with State-Space Search”. This will change the search implementation from depth-first search to iterative deepening, which you implemented in an earlier homework. You should modify the relevant code that is provided to implement this extension. You should apply your approach to the Towers of Hanoi problem with 4 disks.

(Here, you will turn in your code, a screenshot showing the plan that is generated by your code for the 4-disk problem (printed in the text window), and a screenshot of the BlocksTower visualization environment at the end of running your extended planner with 4 disks. See specific instructions at end of this document.)

[Grad Students only, for steps 4 and 5]:

4. **Comparing Extension with Original:** The point of the above extension is to make the plans more optimal. The original planner uses depth-first search, whereas your extension uses iterative deepening. Illustrate the differences between the two approaches by finding an example (in the Towers of Hanoi, 4-disk problem) of where the original planner generates more steps than needed to solve the task. State the plan generated by the original planner, as well as the comparable plan generated by the extended planner, pointing out the inefficiencies in the original plan.

5. **Comparing Your Extended State-Space Planner with Original Problem Reduction**

Planner: Compare and contrast the effectiveness of your extended state-space planner (from step 4) with the original Problem Reduction Planner, as you increase the number of disks from 4 to 5, to 6, to 7, to 8 (or higher, if you like). Specifically, examine the number of actions in the resulting plans, and the time (or number of planning steps) required to discover a plan. Other relevant observations should also be included, such as the optimality, or non-optimality, of the discovered solutions, and your conclusions about the comparative advantages and disadvantages of the two approaches. Write up your findings in a simple report, which states what you discovered, along with the data that supports your findings.

[Everyone] What to submit to BlackBoard (only one submission per team; either team member may submit on behalf of both):

In a single (compressed if needed) tarball or zip file:

- (For bullet 3 above, for Everyone):
 - Your **extended planner code** (in Python), commented appropriately
 - **Instructions** for how to run your code in OpenNERO (in case you change any aspects of OpenNERO other than your agent code)
 - A **screenshot** showing the plan that is generated by the code (which is normally printed out in the text window) for Towers of Hanoi task with 4 disks.
 - A **screenshot** of the BlocksTower environment at the end of running your planner with 4 disks.
- (Grad Students only, for bullets 4 and 5 above): a single pdf file of the report that states your findings.
- (For teams): A **Workload Report** (agreed upon by both team members) that states what each team member did on this project, along with a percentage breakdown (totaling 100%) of how much work each member contributed to the solution. The contribution should be measured over all aspects of the assignment, including conceptual discussions, etc., and should not just be based on how many lines of code each team member wrote.

How grading will be done for teams:

The grading for this assignment, when working in teams, will be as follows. Both members of the team will receive the same grade as a starting point, based on the submission. Then, the Workload Report information either results in the scores staying the same, or it can result in one student's score moving down by some amount (maximum of 10 points).