Automated Planning

HTN Planning Assignment
Assigned: 14 September
Due: Wednesday, 28 September, 8am

Prof. Felipe Meneguzzi*

September 21, 2015

1 Gold Miners Domain

You must work on this project **individually**. You are free to discuss high-level design issues with the people in your class, but every aspect of your actual formalisation must be entirely your own work. Furthermore, there can be no textual similarities in the reports generated by each student. Plagiarism, no matter the degree, will result in forfeiture of the entire grade of this assignment.

For this assignment you will be using the JSHOP2 open source implementation of HTN planning available at [1]. You can obtain a copy of this software from its SourceForge project site, or you can download a more conveniently packaged version of the software from our Moodle area. Your deliverables must be handed in via Moodle using the correspondingly named upload rooms. At the end of this assignment, you will upload **one zip file** containing the domain and problem specification files in the JSHOP2 format, specifically:

- one file named `miners.jshop` containing the domain encoding;
- three files containing the problems you formalised, named `pb`σ`.jshop`, where σ is the sequential number of your problem, e.g. `pb1.jshop`, `pb2.jshop`;
- the same number of files (with similar names) showing the traces created by JSHOP2 for the problems you created (e.g. `pb1.jshop` and `pb1.txt`); and
- one report following the guidelines detailed below.

2 Overview

In this assignment you will formalise HTN methods for the *Gold Miners* domain, originally a challenge for teams of multiple agents working on a dynamic environment, described as follows.

> Recently, rumours about the discovery of gold scattered around deep Carpathian woods made their way into the public. Consequently hordes of gold miners are pouring into the area in the hope to collect as much of gold nuggets as possible. A small team of gold miners find itself exploring the area, avoiding trees and bushes and planning to collect the gold nuggets spread around the woods.

For this project, we will model the domain knowledge encoded as methods in an HTN domain specification. The miners have three operations available to them. First, agents can *move* about the environment to an adjacent quadrant, if this quadrant is not already occupied. Second, agents can *pick up* gold nuggets from the ground and carry a maximum of one gold nugget. Third, agents can *drop* gold nuggets anywhere on the map, but their overall objective is to drop them at a particular collection point. You must use the operators (action templates) as defined below in the JSHOP2 language.

*Credit to the Multiagent Contest organisers Mehdi Dastani, Jüergen Dix and Peter Novák for the scenario `http://multiagentcontest.org/2006`.

[1] `https://sourceforge.net/projects/shop/files/JSHOP2/`

Listing 1: JShop Operators

```
(:operator (!move ?agent ?from ?to)
        ((at ?agent ?from) (adjacent ?from ?to) (not (blocked ?to))) ; Precondition
        ((at ?agent ?from)) ; Delete List
        ((at ?agent ?to)) ; Add List
)

(:operator (!pick ?agent ?gold ?where)
        ((at ?agent ?where) (on ?gold ?where))
        ((on ?gold ?where))
        ((has ?agent ?gold))
)

(:operator (!drop ?agent ?gold ?where)
        ((at ?agent ?where))
        ((has ?agent ?gold))
        ((on ?gold ?where))
)
```

Your assignment is to develop a domain file using the specification above, and encode domain knowledge that will help the JSHOP2 planner to finish the task of collecting all gold nuggets from the environment. The following hints may be useful, but you are welcome to use your creativity as long as you adhere to the specification mentioned above:

- You do not need extra actions for the domain, but you may need to create "bookkeeping" actions to keep track of the locations you have tried to decompose methods to;
- You will need a recursive method to make the agent move all the way to a particular location;
- There are domains that do search over grid space in the examples included in JSHOP2, you may want to look at the `rovers` domain; and
- You may need methods to pick up and drop gold nuggets that use the `!move` method above.

3 Problem Instances

Below are images of the problem instances that you need to model in JSHOP2, once you are done making your domain file. The legend that accompanies each image should be fairly self-explanatory; remember, you must model connections between grid locations for the agents to move from one to the other, and you can only move between grid locations that are connected. In the instances shown in Figures 1 and 2, any locations that share an edge can be considered connected. Locations that share only corners and no edges are not connected.

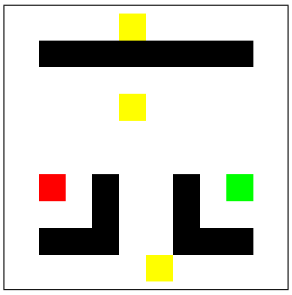

Figure 1: Problem 1 (10x10)

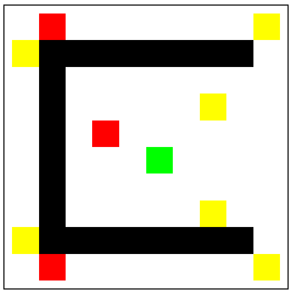

Figure 2: Problem 2 (10x10)

Notice that the above pictures give you an idea of the initial state of the world (which you must encode in your JSHOP2 problem file). They also tell you what the goals are - the deposit (in green), the agents (in red), the various gold nuggets (yellow) that must be collected, and the walls (black). If you look at the syntax of example JSHOP2 problem files, you will see that these are the two parts of a problem file - (1) the initial state, and (2) the initial task network.

4 Grading

In order to properly evaluate your work and thought process, you will write a 2-page report in the AAAI conference format explaining your encoding and experiments. These guidelines are to be followed **exactly**. **Reports that are less than two pages of actual content, or not in format will receive 0 marks for the report criterion.** This report will be included in the deliverables of **Part B** of the assignment. The formatting instructions are available at the AAAI 2014 website[2]. The report must have the following sections:

- An introduction with your understanding of the problem domain, outlining the remainder of the paper;
- Two domain formalisation sections explaining your approach to formalising the problems from Section 3
- One experimentation section where the performance of JSHOP2 is measured using your domain knowledge for each of the domains, on multiple problems.
- One conclusion section, where you will summarise your experience in encoding planning domains and discuss the performance of JSHOP2, and any limitations encountered in solving the problems you encoded.

Grading will take consider elements of your encoding, experimentation and reporting of the work done. The criteria, as well as their weight in the final grade is as follows:

- Domain Encoding (30%) — correctness of the domain knowledge encoding, in relation to the domain specification from Section 2;
- Problem specifications (20%) — correctness of the problem specifications used for the experiments, particularly the initial state specification, as missing predicates here will jeopardise JSHOP2's ability to solve your problem;
- Overall report readability (20%) — how accessible and coherent your explanation of your encoding is;
- Experiments (30%) — how coherent the proposed experiments are in measuring the performance of JSHOP2, notice that this criteria is complementary to the one regarding problem specification.

[2] `http://www.aaai.org/Conferences/AAAI/2014/aaai14call.php`

5 Miscellaneous Advice

Here are some lessons we learned in creating our own solution and writing papers/reports:

- As a first step, you should at least look at the sample domain and problem files included in the JShop2 package.
- The best way to figure out how to model a domain and associated problems is to look at these examples. If you feel the need for documentation, there is a complete manual included in the package from Sourceforge as well as the one from moodle.
- Tables and graphs are a useful tool to show runtime performance of software[3];
- In order to evaluate the performance of a planning encoding, you need to specify problems with most of the parameters locked in, and measure runtime as one parameter increases (e.g. number of locations, number of containers, etc);
- Pasting your entire domain specification (or problems) into the paper does not count as content (now you cannot say you were not warned);
- Overly large figures used to simply fill space in the report are also not a good idea;
- Reviewers have a more pleasant reading experience when papers are generated using LaTeX, it is very easy to spot the difference.

[3] GnuPlot is an excellent (and free) graph making software, available at `http://www.gnuplot.info/`