Final Project Guidelines

CS 660: Topics in Artificial Intelligence: Planning Algorithms

The purpose of the final project in this course is to explore a topic within planning research that is of interest to you and which applies principles studied in this course. The project is divided into three parts:

1. A roughly one page proposal to be approved by your instructor.
2. A deliverable consisting of code, data, a written report, or other products generated during the project.
3. A final presentation to your instructor and peers summarizing your results.

Proposal (20%)

The proposal should clearly state what topic you want to investigate for your final project. It should clearly define what you will submit for your deliverables. If your project proposes to improve on a planning algorithm in some way, you should also define the ideal results you will obtain; however, as long as your deliverables are as promised, you do not need to achieve these results in order to receive full credit for the project. For example, if you propose a new method to improve the speed of a planner, you must implement that new method correctly, but you do not necessarily need to beat every other planner to be successful. If your idea is correctly implemented but fails to achieve the expected results, you should identify why it failed. See some suggestions for possible projects at the end of this document for more examples.

The proposal should be about 1 page in length. In most cases, the proposal should cite relevant research papers related to the topic you want to investigate. You will be expected to have read and understood these papers as relevant to your project.

If your proposal is problematic in some way, your instructor may ask you to amend it to avoid significant penalties on later phases of the project.

Deliverable (40%)

This is a self-defined project, so your deliverables will depend on your proposal, but common deliverables include:

- An implementation of a new or improved planning algorithm.
- Results demonstrating the success of your project.
- A written report demonstrating that your deliverables work as expected.

Presentation (40%)

Each person will give a short (about 15 minute) presentation about their project at the end of the semester. The purpose of this presentation is to communicate to your instructor and your peers what you learned during the project. Ideally, your results will be interesting and useful to others.

Some Examples of Projects

These examples are meant to demonstrate the kinds of topics that would make acceptable final projects.

- **Improving the POP Heuristic:** The default POP heuristic is simply to count the number of open preconditions in a partial plan. Using a more sophisticated heuristic, such as h^+ or *FF*, should improve the speed of the POP planner. The deliverables will be (1) an implementation of POP that uses the HSP heuristic to estimate the cost of open preconditions, (2) a written report showing a full example of the improved planner solving the Sussman anomaly, showing a side-by-side comparison of the original POP heuristic and this improved heuristic, and (3) results of a comparison of this new planner vs. all other planners. Ideally, the improved POP algorithm will outperform POP and several other algorithms.
- **Identifying and Exploiting Landmarks:** Identifying non-trivial landmarks in a planning problem is, itself, a planning problem, but some techniques have been demonstrated to identify some landmarks efficiently. By identifying landmarks and using them as waypoints during search, we can improve state-space planners like HSP. The deliverables will be (1) a implementation of a landmark identification algorithm, (2) an implementation of HSP that first identifies landmarks and then uses them to solve problems faster, (3) a written report showing the landmarks discovered in one of the simple cargo problems, and (4) results of a comparison of this new planner vs. all other planners. Ideally, the improved HSP will outperform HSP and several other algorithms.
- **Planning with Durative Actions:** One of the many branches of planning research deals with actions that have an explicit time duration. For these problems, an optimal plan is no longer one with a minimal number of actions but one which takes a minimal amount of time. The deliverables for this project will be (1) a variation on the cargo domain which assigns a duration to each action and several example problems in this new cargo domain, (2) an implementation of HSP which accounts for action duration in order to generate plans that takes the least possible time, (3) results of comparing this algorithm to all other planners on the new problems in the new domain, showing that explicit reasoning about action duration can lead to shorter (in terms of time) solutions.