CompSci 767: Intelligent Software Agents

Assignment 2 Version 1.0

Worth: 25%

Due: Friday 30 May 2014

Goal of Assignment:
To give a taste of what research is like.

Context:
To decide whether one heuristic is better than another for a particular problem, you need to estimate how these heuristics will affect the problem solver's time to solve the problem. One component in this calculation will be predicting how many nodes will be generated/expanded during the search for a solution. Recently, there has been a lot of research into predicting this for the IDA* search algorithm. Your assignment will be to become acquainted with that research and propose an extension to the current state of the art. Some background on this task follows below.

When generating a search space, the same state may be generated numerous times. While this is often unavoidable, expanding that state more than once is a waste of time. There are three different approaches to avoiding the multiple expansions of states. Each of these approaches has a different impact on the number of nodes generated/expanded during the search for a problem's solution.

There are various formulas for predicting how many nodes will be generated/expanded by IDA* heuristic searches when not checking for duplicated states. However, no one has published a formula that accurately predicts the number of nodes generated/expanded for A* (A* does global duplicate state checking). The same formulas, that are so accurate for IDA* without duplicate state checking, are quite inaccurate for A*. *Global duplicate state checking* is the removal of a node's child that has a state that is already in the closed list (i.e., has been expanded). Global duplicate state checking is one extreme of duplicate state checking. The other extreme is grandfather pruning. *Grandfather pruning* is the removal of a node's child that has a state that matches the state of the node's parent.

In between global duplicate state checking and grandfather pruning is loop elimination. *Loop elimination* is the removal of a node's child that has a state that already appears on the path from the root node to this node. Note that loop elimination subsumes grandfather pruning and is subsumed by global duplicate state checking. While there are formulas that accurately predict the number of nodes generated/expanded by IDA* heuristic searches using grandfather pruning, there are no such formulas for loop elimination.

While finding a formula that accurately predicts the number of nodes generated/expanded by IDA* with loop elimination would be a contribution in and of itself, it also serves as an intermediate step to the larger goal of finding such a formula for A*. We will simplify this one step further, by using iterative deepening (ID) search. In ID search, we still need to model the impact of loop elimination but we can elimination the complications introduced by using a heuristic.

Task:

Your task is to write a *research proposal* for a project to find and verify a formula for predicting the number of nodes generated/expanded by ID search with loop elimination.

You will need to read, at a minimum, the papers in the reading list and try to understand theorem 1's formula in the Korf, Reid, Edelkamp 2001 paper. Check out how accurate their formula is for IDA* with and without loop elimination. Assuming that it is more accurate for IDA* without loop elimination, you need to:

- Establish a baseline for how accurate the basic formula is without loop elimination (in doing this, you will need to correctly model the state types).
- Establish how inaccurate the formula is when using loop elimination.
- Hypothesize an extension to one of the standard formulas for ID that provides a better prediction of the number of nodes generated/expanded by ID.
- Write a research proposal.

The domain you will be using is the 8-puzzle.

Resources:

The following will be provided:

- Initial reading list.
- Prolog implementation of ID.
- Prolog implementation of 8-puzzle domain.
- Various helper code files.
- Collection of 8-puzzle problems categorized by optimal solution length between ten and twenty-five steps.

Outline of Report:

- Introduction
- Problem Description
- Literature Survey
- Description of your experimental results to determine baseline accuracy of the standard formula for ID without loop elimination. Specifically, what is the mean error and its standard deviation for the problems in 10.txt without loop elimination?

- Description of your experimental results to determine the level of inaccuracy of the standard formula when predicting performance of ID with loop elimination. Specifically, what is the mean error and its standard deviation for the problems in 10.txt with loop elimination?

- Description of your proposed modification to the formula and the rationale behind the modification
- Description of experiments you would run to verify whether your extension improved the formula's accuracy when predicting ID's performance with loop elimination.
- Summary

Reading List:
"The branching factor of regular search spaces" by Edelkamp, S. and Korf, R.E. in 1998 Proceedings of the National Conference on Artificial Intelligence.

"Complexity analysis of admissible heuristic search" by Korf, R.E. and Reid, M. in 1998 Proceedings of the National Conference on Artificial Intelligence.

"Time complexity of iterative-deepening-A∗" by Korf, R.E. and Reid, M. and Edelkamp, S. in Artificial Intelligence Journal, Volume 129, number 1, 2001.

"Fast and Accurate Predictions of IDA*'s Performance" by Lelis, L.H.S. and Zilles, S. and Holte, R.C. in 2012 Proceedings of the National Conference on Artificial Intelligence.

I recommend that you read these papers in the order given above. These papers are not light reading, I suggest that you get together with other students in this class and discuss them. This will not only improve your understanding of the papers, but may make them more enjoyable. These four papers should be just the beginning for your literature survey. Use them to find other relevant papers.

What You Turn In
You need to send me (barley@cs.auckland.ac.nz) the following:
- The Report
- A zip archive of your code
- A text file listing that shows the problems used for the experimental runs