Curtin University – Department of Computing

Assignment Cover Sheet / Declaration of Originality

Complete this form if/as directed by your unit coordinator, lecturer or the assignment specification.

Last name:	Beardsmore	Student ID:	15504319
Other name(s):	Connor		
Unit name:	Artificial and Machine Intelligence	Unit ID:	COMP3006
Lecturer / unit coordinator:	Mihai Lazarescu	Tutor:	Stefan
Date of submission:	01/05/2017	Which assignment?	(Leave blank if the unit has only one assignment.)

I declare that:

- The above information is complete and accurate.
- The work I am submitting is *entirely my own*, except where clearly indicated otherwise and correctly referenced.
- I have taken (and will continue to take) all reasonable steps to ensure my work is *not accessible* to any other students who may gain unfair advantage from it.
- I have *not previously submitted* this work for any other unit, whether at Curtin University or elsewhere, or for prior attempts at this unit, except where clearly indicated otherwise.

I understand that:

- Plagiarism and collusion are dishonest, and unfair to all other students.
- Detection of plagiarism and collusion may be done manually or by using tools (such as Turnitin).
- If I plagiarise or collude, I risk failing the unit with a grade of ANN ("Result Annulled due to Academic Misconduct"), which will remain permanently on my academic record. I also risk termination from my course and other penalties.
- Even with correct referencing, my submission will only be marked according to what I have done myself, specifically for this assessment. I cannot re-use the work of others, or my own previously submitted work, in order to fulfil the assessment requirements.
- It is my responsibility to ensure that my submission is complete, correct and not corrupted.

Signature:  Date of signature: 01/05/2017

(By submitting this form, you indicate that you agree with all the above text.)

AMI300 Report

Informed Beam and SMA* Search Implementations

Connor Beardsmore - 15504319

Curtin University
Science and Engineering
Perth, Australia
May 2017

Informed Beam Search

Design Decisions

The informed beam search is a non-complete and non-optimal search technique based on an admissible heuristic measure. The cost of each node is determined as $f(n) = h(n)$, thus the decision of which nodes to expand is based solely on heuristic cost. The algorithm tracks up to k beams or paths at each step. Each further step expands all children nodes from these beams and expands the best k choices. Informed beam searches sacrifices optimality and completeness for increased memory efficiency and speed (Winston 1993).

The algorithm implemented utilizes a priority queue for both the *beam* and the *frontier* data structures, sorted via heuristic cost. The *beam* represents the current nodes in the k beams while the *frontier* stores all children of every node in the beams. At each level the frontier is trimmed to k length to allow the best k choices to be chosen. After this stage, the beam is replaced by the frontier and the process repeats until a solution is found, or no valid paths can be explored.

The algorithm allows for the discovery of alternate paths after an initial solution is discovered. Once the goal node is discovered in the frontier, the path is stored and the goal removed from the frontier to allow the beams to continue as if the goal was never discovered.

Problems and Bugs

Several issues were faced during the implementation and testing of the algorithm. However, these issues were effectively resolved and the algorithm currently has no known issues or bugs resulting in errors.

The beams in the search technique do not communicate, they progress independently. Initially, the beams were communicating via comparing successors as they were added to the frontier. This fault was simply resolved by not adding duplicate nodes to the frontier at any stage. Two beams can effectively converge at the same node whilst having different paths to reach that node. Initially, nodes were not being duplicated and paths were being overwritten and lost. To solve this problem, nodes were duplicated to save their specific individual paths to allow for beams to converge upon the same nodes.

Before improvements were applied, the list of partial paths stored when a solution is discovered was not being correctly stored. Similarly to other issues, this was resolved by creating a deep copy of the beam before it was modified to allow for partial paths to be displayed if required.

Simplified Memory Limited A* Search

Design Decisions

The simplified memory limited A* search (SMA*) is an extensible to pure memory bounded A* search, designed by Stuart Russell (Russell 1992). It provides a more memory efficient form of the regular A* search by placing a cap on the number of nodes in memory at anytime. Like A* search, the evaluation function for a given node is defined as $f(n) = g(n)+h(n)$, thus being the sum of accumulated path cost and heuristic cost. It will produce the optimal solution given an admissible and consistent heuristic (Russell and Norvig 2016).

Problems and Bugs

bookkeeping, what data structures used, issues with looping, duplicate nodes, continuing etc, how bad the regular pseudocode is (removing parent from memory etc)

References

Russell, Stuart. 1992. "Efficient Memory-bounded Search Methods". In *Proceedings of the 10th European Conference on Artificial Intelligence*. ECAI '92. Vienna, Austria: John Wiley & Sons, Inc.

Russell, Stuart, and Peter Norvig. 2016. *Artificial Intelligence: A Modern Approach*. 3rd ed. Pearson.

Winston, Patrick. 1993. *Artificial Intelligence*. 3rd ed. Addison-Wesley.