Lab4 - Planning

Naman Goyal (2015CSB1021)

April 10, 2017

1 Planners for Block World

Objective To compare the performance of various planners

- Forward (progression) planner using breadth first search
- Forward (progression) planner using A* search
- Goal Stack planner

1.1 Statistics

1.txt (4 blocks)

Performance Metric	**BFS**	**Admissible A* Search**	**Inadmissible A* Search**	**Goal Stack**
Search Time (in sec)	0.018	0.003	0.003	0.003
Path Length	10	10	10	18
Number of Nodes expanded	109	21	19	–
Size of Goal Stack	–	–	–	9

3.txt (5 blocks)

Performance Metric	**BFS**	**Admissible A* Search**	**Inadmissible A* Search**	**Goal Stack**
Search Time (in sec)	0.024	0.004	0.004	0.003
Path Length	14	14	16	18
Number of Nodes expanded	499	45	42	–
Size of Goal Stack	–	–	–	15

5.txt (12 blocks)

Performance Metric	**BFS**	**Admissible A* Search**	**Inadmissible A* Search**	**Goal Stack**
Search Time (in sec)	–	4.012	0.145	0.004
Path Length	–	26	26	30
Number of Nodes expanded	–	22040	936	–
Size of Goal Stack	–	–	–	19

BFS was not used for 12 blocks case since BFS expands all nodes and hence becomes memory intensive. Also BFS takes a huge time since it parses level by level to reach goal state.

1.2 Analysis

1. **Observation**

   A* Search reduces the search time and number of nodes expanded significantly over BFS

   Explanation

   A* Search tries expanding those nodes which are more closer to goal state using heuristic value.

2. **Observation**

   Inadmissible heuristics gives plan quickly but may return suboptimal plan over the admissible one.

   Explanation

   Inadmissible heuristics overestimates the value of reaching the goal state and hence may return suboptimal answer.

3. **Observation**

   Goal Stack planning is faster than other two and requires much less memory but returns suboptimal plan.

   Explanation

   Goal Stack Planning is a Goal Directed Search hence finds solution quickly; but Sussman's anomaly proves that Goal Stack Planning is suboptimal.

1.3 Heuristics

Heuristics returns an estimated value of arriving at goal state.

It returns the estimated cost of solving a relaxed version of original sub-problem. The constraint that robotic arm can hold only one block is discarded and the robotic hand can now hold any number of blocks called **pool**. A block is added to pool only once using a cost of 1 step and is then removed from the pool again.

Cost of removing a block for admissible case is 1. While the cost of removing a block for 'inadmissible' case is 1 (if block is ontable) and 3 (if block is on another block). Each of the unsatisfied literals in goal state is considered and the cost estimates of satisfying them are added.

Consider the situation

Current State (on 1 2) (on 2 3) (ontable 3) (clear 1) (empty)
Goal State (on 1 2) (ontable 2) (ontable 3) (clear 3) (clear 1) (empty)

To satisfy (Clear 3) all the blocks above 3 i.e. 1 and 2 are put into the pool. Since none of them was already in pool the cost is 2 steps.

Then since the goal state has arm empty; 1 and 2 are removed from the pool.

For admissible case, the blocks 1 2 are removed from pool using a step cost 1 each i.e. total 2 steps.

For inadmissible case, the block 1 (on 2) is removed using a cost of 3 and block 2 (ontable) is removed using a cost of 1 from pooli.e. total 4 steps.

Hence Heuristic Value is 4 (for admissible) and 6 (for inadmissible) while actual cost is 6

Consider the situation

Current State (on 1 3) (on 3 4) (ontable 4) (ontable 2) (clear 1) (clear 2) (empty)
Goal State (ontable 1) (ontable 2) (ontable 3) (ontable 4) (clear 1) (clear 2) (clear 3) (clear 4) (empty)

To satisfy (Clear 3) all the blocks above 3 i.e. 1 is put into the pool. Since 1 was not already in pool the cost is 1 step.

Then to Satisfy (Clear 4) all the blocks above 4 i.e. 1 and 3 are put into the pool. Since 1 was already in pool the cost is 0 step while 3 was not in pool so cost is 1 step. Net cost of putting in pool is $1 + 0 = 1$ step

For admissible case, the blocks 1 3 are removed from pool using a step cost 1 each i.e. total 2 steps.

For admissible case, the blocks 1 3 are removed from pool using a step cost 1 each (as both on table) i.e. total 2 steps.

Hence Heuristic Value is 4 (for admissible) and 4 (for inadmissible) while actual cost is 4.