16350: Planning Techniques for Robotics, Homework 3

[redacted]
Carnegie Mellon University
[redacted]

1. Running Instructions

To run my code in Windows, use the following commands:
g++ planner.cpp -o planner.out

To execute:
planner.out example.txt
where 'example.txt' is the file containing the environment.
In linux you might need to use:
./planner.out example.txt
if the other command doesnt work.

Planner ids are assigned as follows.
$0 \rightarrow$ base planner
$1 \rightarrow$ inadmissible planner
$2 \rightarrow$ admissible planner

You can run the planner you want by changing the variable **heur** in the file 'planner.cpp'- line 1029.

2. Environment Information

The block triangles environment and the fire extinguisher environment have been described in the text files in the code folder. For the inadmissible planner, we use the relaxed heuristic which is described later in this report. However for it to work efficiently particularly for the fire extinguisher case, we defined the goal to include 6 states as shown in the text file.

3. Planner Description

3.1. Base planner

All heuristics are set to 0. Edge costs are set to 1. It is a basic Dijkstra search where the OPEN list is prioritized on the basis of the g values.

3.2. Inadmissible heuristic

It makes use of a heuristic which is basically the total no of conditions in the goal state minus the number of conditions present in the given state that are found in the goal state. Edge costs are set to 1. Whilst this heuristic speeds up the search, it is often times an overestimate since a since multiple conditions can be satisfied by a single action. An A star search is performed using this heuristic.

3.3. Admissible heuristic

The heuristic is calculated for each state using a forward Dijkstra search. It does not overestimate since it accounts for actions satisfying multiple conditions. In order to speed things up a bit, a weighted A star search is used which reduces the planning time significantly.

4. Results

4.1. Formulated Plan

For the **blocks environment**
MoveToTable(A,B)
Move(C,Table,A)
Move(B,Table,C)

For the **block triangles environment**
MoveToTable(T0,B0)
MoveToTable(B0,B1)
MoveToTable(T1,B3)
Move(B1,B4,B3)
Move(B0,Table,B1)
Move(T1,Table,B0)

For the **fire extinguisher**
MoveMobile(A,B)
QuadcopterLand(B)
MoveMobileAndQuadcopter(B,W)
FillTank(Q)
MoveMobileAndQuadcopter(W,F)
QuadcopterTakeOff(Q)
FirstDouse(Q)
QuadcopterLand(F)
MoveMobileAndQuadcopter(F,W)
FillTank(Q)
MoveMobileAndQuadcopter(W,F)
Charge(Q)
QuadcopterTakeOff(Q)
SecondDouse(Q)
QuadcopterLand(F)
MoveMobileAndQuadcopter(F,W)

FillTank(Q)
MoveMobileAndQuadcopter(W,F)
Charge(Q)
QuadcopterTakeOff(Q)
ThirdDouse(Q)

4.2. Evaluation of performance

There are 3 parameters of interest that are studied for all 3 environments with each of the 3 planners: time, states expanded and number of moves. All the 3 planners output plans with the same number of moves. In terms of steps taken for the given environments, all 3 planners are optimal. Th base planner probably is the easiest implementation. But since it has no heuristic, it always expands more states than the other 2 planners. In terms of time taken, the inadmissible heuristic planner performs the best for all 3 environments generating paths within a couple of seconds even for more complex environments. The admissible planner takes the most time compared to the other 2 planners (almost 22 minutes for the fire extinguisher case).

	Time (sec)	States Expanded	Moves	
Block	0.023	6	3	Base
	0.019	5	3	Inadmissible
	0.317	3	3	Admissible
Block triangle	4.771	220	6	Base
	0.93	64	6	Inadmissible
	35.61	6	6	Admissible
Fire extinguisher	16.848	866	21	Base
	5.892	736	21	Inadmissible
	1327.96	21	21	Admissible

Figure 1: A summary of results

5. Conclusion

If time is of essence, inadmissible heuristics are far superior compared to both the admissible planner and the base planner. However, they tend to expand more states than the admissible search. The admissible search might guarantee non overestimation of the heuristic and fewer expansion of states, however it has huge computational complexity and performs the worst in terms of time taken. In terms of completeness, inadmissible planner is superior though it is sub-optimal. Overall, using a heuristic makes more sense. But there should be a trade-off between time taken and whether the heuristic is admissible or not for a plan that is optimal