Udacity – Artificial Intelligence Nanodegree Program

Project: Forward Planning Agent
Student: Carlos Hernandez
Submission Date: August 19th, 2018

For this project, a progression search agent was developed to solve planning problems. The project required the implementation of support functions and a progression search experiment to evaluate the performance for different algorithms

Section 1. Functions Implemented

Action Layer
- _inconsistent_effects
- _interference
- _competing_needs

Literal Layer
- _inconsistent_support
- _negation

Heuristics
- h_levelsum
- h_maxlevel
- h_setlevel

Using the functions outlined above, a set of experiments was performed for 4 airplane cargo problems. The experimental setup consisted of different search algorithms and heuristics.

Section 2. Progression Search Experiment

The experiment consists of 3 uninformed and 2 informed search algorithms. For the informed search, there are 4 heuristics available.

Problem 1 and 2 were solved using the 11 different search-heuristic combinations possible. Results are below:

Problem 1 - 20 Actions					
Algorithm	Expansions	Goal Tests	New Nodes	Time (s)	Plan Length
Breadth-First Search	43	56	178	0.006	6
Depth-First Graph Search	21	22	84	0.003	20
Uniform Cost Search	60	62	240	0.010	6
Greedy Best First Graph Search					
Unmet Goals	7	9	29	0.002	6
Planning Graph - Level Sum	6	8	28	0.132	6
Planning Graph – Max Level	6	8	24	0.093	6
Planning Graph – Set Level	6	8	28	0.377	6
A*					
Unmet Goals	50	52	206	0.009	6
Planning Graph - Level Sum	28	30	122	0.324	6
Planning Graph – Max Level	43	45	180	0.345	6
Planning Graph – Set Level	33	35	138	0.887	6

Problem 2 - 72 Actions					
Algorithm	Expansions	Goal Tests	New Nodes	Time (s)	Plan Length
Breadth-First Search	3,343	4,609	30,503	1.787	9
Depth-First Graph Search	624	625	5,602	2.683	619
Uniform Cost Search	5,154	5,156	46,618	3.026	9
Greedy Best First Graph Search					
Unmet Goals	17	19	170	0.017	9
Planning Graph - Level Sum	9	11	86	2.702	9
Planning Graph – Max Level	27	29	249	4.228	9
Planning Graph – Set Level	9	11	84	9.194	9
A*					
Unmet Goals	2,467	2,469	22,522	2.074	9
Planning Graph - Level Sum	357	359	3,426	72.722	9
Planning Graph – Max Level	2,887	2,889	26,594	427.298	9
Planning Graph – Set Level	1,037	1,039	9,605	829.104	9

Problem 3 and 4 were solved using 1 uninformed search and 2 heuristics for each informed search. Results are below:

Problem 3 - 88 Actions					
Algorithm	Expansions	Goal Tests	New Nodes	Time (s)	Plan Length
Uniform Cost Search	18,510	18,512	161,936	13.201	12
Greedy Best First Graph Search					
Unmet Goals	25	27	230	0.032	15
Planning Graph - Level Sum	14	16	126	6.251	14
A*					
Planning Graph - Level Sum	369	371	3,403	133.527	12
Planning Graph – Max Level	9,580	9,582	86,312	2403.088	12

Problem 4 - 104 Actions					
Problem 4	Expansions	Goal Tests	New Nodes	Time (s)	Plan Length
Breadth-First Search	99,736	114,953	944,130	85.289	14
Greedy Best First Graph Search					
Planning Graph - Level Sum	17	19	165	11.497	17
Planning Graph – Max Level	56	58	580	19.544	17
A*					
Unmet Goals	34,330	34,332	328,509	51.465	14
Planning Graph - Level Sum	1,208	1,210	12,210	761.521	15

Section 3. Conclusions

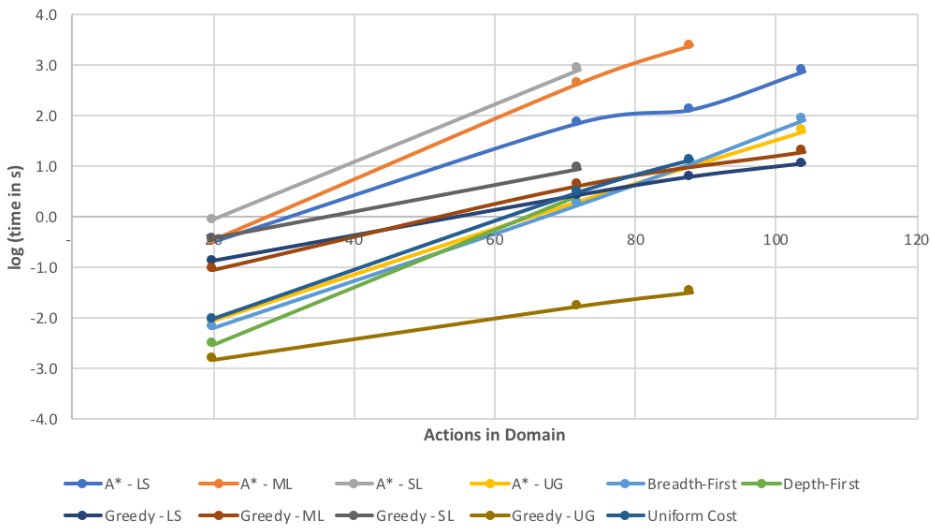

Figure A. Graph outlining Actions in Domain vs. the natural logarithm of time

1. **Algorithms most appropriate for planning in a very restricted domain (i.e. one that has only a few actions) and needs to operate in real time?**

As observed in Figure A, Greedy Best First Graph Search using the Unmet Goals heuristic (Greedy – UG) outperforms all other algorithms with a time under 1 sec. for all tested problems. If we consider a very restricted domain, uninformed search algorithms appear to generally outperform informed search one. Similarly, the Unmet Goals heuristic performs better than other heuristics for informed search.

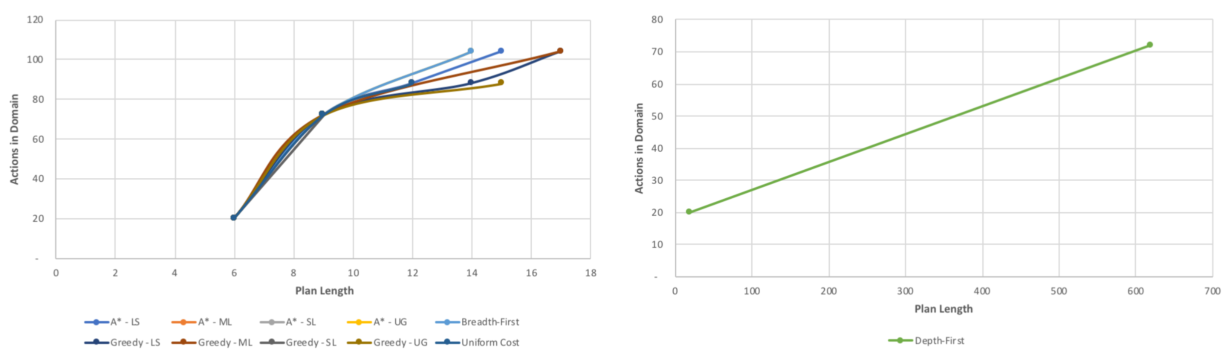

Figure B. Graphs outlining Actions in Domain vs. Plan Length

2. **Algorithms most appropriate for planning in very large domains (i.e. planning routes for all UPS drives in the U.S. on a given day)?**

As observed in Figure B, Greedy searches provide the best trade-off between time and plan length. Breadth-first search and Uniform Cost search are also good alternatives.

3. **Algorithms most appropriate for planning problems where it is important to find only optimal plans?**

Breadth-First Search and Uniform Cost Search are most appropriate for planning problems requiring an optimal plan as they are guaranteed to find an optimal solution. Similarly, A* might be a good alternative if the heuristic implemented always remains optimistic, that is always underestimates the true length of the plan.