Artificial Intelligence Nanodegree

Build a Forward Planning Agent

Candidate: Milos Randic

Problem	1					
Algorithm	Actions	Expansions	Duration	Plan Length	Expansions vs Actions	Durations vs Actions
breadth_first_search	20	43	0,05699	6	2,15000	0,00285
depth_first_graph_search	20	21	0,02608	20	1,05000	0,00130
uniform_cost_search	20	60	0,05325	6	3,00000	0,00266
greedy_best_first_graph_search h_unmet_goals	20	7	0,00605	6	0,35000	0,00030
greedy_best_first_graph_search h_pg_levelsum	20	6	1,68365	6	0,30000	0,08418
greedy_best_first_graph_search h_pg_maxlevel	20	6	0,51008	6	0,30000	0,02550
greedy_best_first_graph_search h_pg_setlevel	20	6	2,53579	6	0,30000	0,12679
astar_search h_unmet_goals	20	50	0,04952	6	2,50000	0,00248
astar_search h_pg_levelsum	20	28	0,88009	6	1,40000	0,04400
astar_search h_pg_maxlevel	20	43	0,71670	6	2,15000	0,03584
astar_search h_pg_setlevel	20	33	1,88787	6	1,65000	0,09439

Problem	2					
Algorithm	Actions	Expansions	Duration	Plan Length	Expansions vs Actions	Durations vs Actions
breadth_first_search	72	3343	0,93396	9	46,43056	0,01297
depth_first_graph_search	72	624	3,19903	619	8,66667	0,04443
uniform_cost_search	72	5154	1,53799	9	71,58333	0,02136
greedy_best_first_graph_search h_unmet_goals	72	17	0,02830	9	0,23611	0,00039
greedy_best_first_graph_search h_pg_levelsum	72	9	5,74163	9	0,12500	0,07974
greedy_best_first_graph_search h_pg_maxlevel	72	27	5,52037	9	0,37500	0,07667
greedy_best_first_graph_search h_pg_setlevel	72	9	6,16439	9	0,12500	0,08562
astar_search h_unmet_goals	72	2467	1,74837	9	34,26389	0,02428
astar_search h_pg_levelsum	72	357	63,81781	9	4,95833	0,88636
astar_search h_pg_maxlevel	72	2887	319,01634	9	40,09722	4,43078
astar_search h_pg_setlevel	72	1037	419,55836	9	14,40278	5,82720

Problem	3					
Algorithm	Actions	Expansions	Duration	Plan Length	Expansions vs Actions	Durations vs Actions
breadth_first_search	88	14663	1,83189	12	166,62500	0,02082
depth_first_graph_search	88	408	1,20738	392	4,63636	0,01372
uniform_cost_search	88	18510	2,07631	12	210,34091	0,02359
greedy_best_first_graph_search h_unmet_goals	88	25	0,01851	15	0,28409	0,00021
greedy_best_first_graph_search h_pg_levelsum	88	14	3,14828	14	0,15909	0,03578
greedy_best_first_graph_search h_pg_maxlevel	88	21	2,75077	13	0,23864	0,03126
greedy_best_first_graph_search h_pg_setlevel	88	35	12,20955	17	0,39773	0,13874
astar_search h_unmet_goals	88	7388	1,67958	12	83,95455	0,01909
astar_search h_pg_levelsum	88	369	94,78334	12	4,19318	1,07708
astar_search h_pg_maxlevel	88	9580	804,54120	12	108,86364	9,14251
astar_search h_pg_setlevel	88	3423	931,23567	12	38,89773	10,58222

Problem	4					
Algorithm	Actions	Expansions	Duration	Plan Length	Expansions vs Actions	Durations vs Actions
breadth_first_search	104	99736	6,87324	14	959,00000	0,06609
depth_first_graph_search					#DIV/0!	#DIV/0!
uniform_cost_search	104	113339	8,32641	14	1.089,79808	0,08006
greedy_best_first_graph_search h_unmet_goals	104	29	0,03959	18	0,27885	0,00038
greedy_best_first_graph_search h_pg_levelsum	104	17	4,65061	17	0,16346	0,04472
greedy_best_first_graph_search h_pg_maxlevel	104	56	8,72063	17	0,53846	0,08385
greedy_best_first_graph_search h_pg_setlevel	104	107	51,61538	23	1,02885	0,49630
astar_search h_unmet_goals	104	34330	5,10252	14	330,09615	0,04906
astar_search h_pg_levelsum	104	1208	199,16870	15	11,61538	1,91508
astar_search h_pg_maxlevel						
astar_search h_pg_setlevel						

Analyze the search complexity as a function of domain size, search algorithm, and heuristic.

From table above it can be easily noticed that number of node expansions vs number of actions dramaticaly increases as number of actions in domain gets higher, for algorithms *breadth_first_search, uniform_cost_search, astar_search h_unmet_goals, astar_search h_pg_maxlevel* and *astar_search h_pg_setlevel*.

For the rest of the algorithms, number of expansion nodes seems to be rasonably lower.

Analyze search time as a function of domain size, search algorithm, and heuristic.

As the problem size increases, it can be concluded that A* algotithms (*h_pg_setlevel, h_pg_maxlevel and h_pg_levelsum*) have the highest duration over problem size which makes them not that optimal in real-time execution, tough the plan length is the shortest among those algorithms. In addition, the lowest duration over problem size is achieved among *uniform_cost_search, breadth_first_search, astar_search h_unmet_goals, depth_first_graph_search* and *greedy_best_first_graph_search h_unmet_goals*.

Analyze the optimality of solution as a function of domain size, search algorithm, and heuristic.

The highest plan length, for all problems is generated by *depth_first_graph_search* algorithm. This solution migth not be desireable to use in practice due really high number of actions that need to be executed in order to achieve the goal state.

Contrarily, for P1 and P2 it is clear that plan length is constant for the rest of the algorithms.

Problems P3 and P4 have an increase of plan length when using *greedy_best_first_graph_search* algorithms.

Questions

1. **Which algorithm or algorithms would be 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?**

P1 seems to have a very restricted domain (20 actions), so, from the table above, *greedy_best_first_graph_search h_unmet_goals* algorithm gives the lowest duration over number of actions, which makes it a good candidate for real-time execution with a plan length of 6. Also, *astar_search h_unmet_goals* and *uniform_cost_search* would be good candidates for operating in real-time.

2. **Which algorithm or algorithms would be most appropriate for planning in very large domains (e.g., planning delivery routes for all UPS drivers in the U.S. on a given day)**

P4 has the large domain (104 actions) and *greedy_best_first_graph_search h_unmet_goals* would be a good option in terms of duration of execution, plan length and number of expansions. As an alternative, *greedy_best_first_graph_search h_pg_levelsum* can be executed with reasonable number of expansions and time duration.

3. **Which algorithm or algorithms would be most appropriate for planning problems where it is important to find only optimal plans?**

In terms of plan length, optimal plan should be the one with shortest lenght, for P1 and P2 those are all except *depth_first_graph_search,* for P3 those are A* search algorithms and all uninformed search algorithms except *depth_first_graph_search* and for P4 *astar_search h_unmet_goals* algorithm and all uninformed search algorithms except *depth_first_graph_search* algorithm are considered as optimal.