Artificial Intelligence Nanodegree
Build a Forward Planning Agent

Nalbert Gabriel Melo Leal[1]

[1]Department of Informatics and Applied Math (DIMAp),
Federal University of Rio Grande do Norte (UFRN),
Natal, RN, Brazil

nalbertgml@gmail.com

Abstract. *This document explains the analysis of the second udacity artificial intelligence nanodegree project. In this project was to develop the code of an agent that perform progression search to solve an air cargo problem using a combination of logic an classical search. Tables with metrics shown bellow help the reader to understand the conclusions.*

1. Report

After run the algorithms to all four problems it appears that the greedy best fist graph search unmet goals is by far in all problems the fastest in execution time. The DFS was taking to much time to solve the fourth problem, so instead the BFS was chosen to solve the problem four. The BFS took 95 seconds to solve the problem four, but it's certainly faster than the DFS that with 10 minutes didn't solved the problem.

So if the requirement is a fast agent to solve the air cargo problem the greedy best fist graph search unmet goals it's obvious the best choice. But if the agent must deliver the best solution than the A* search unmet goals is the best choice, because to the ploblems 3 and 4 (apparently more complex problems since it demands more time to the algorithms find a solution) the A* search unmet goals find a solution with less plan length.

Air cargo problem 1:

The table bellow show the metrics of all possible algorithms solving the problem 1

	Actions	Expansions	Goal Tests	New Nodes	Plan length	Time elapsed
breadth_first_search	20	43	56	178	6	0,006654964
depth_first_graph_search	20	21	22	84	20	0,0037494
uniform_cost_search	20	60	62	240	6	0,007480125 9
greedy_best_first_graph_search h_unmet_goals	20	7	9	29	6	0,001742404 9
greedy_best_first_graph_search h_pg_levelsum	20	6	8	28	6	0,448663095
greedy_best_first_graph_search h_pg_maxlevel	20	6	8	24	6	0,33615749
greedy_best_first_graph_search h_pg_setlevel	20	6	8	28	6	1,235391928
astar_search h_unmet_goals	20	50	52	206	6	0,009204217 9
astar_search h_pg_levelsum	20	28	30	122	6	1,095628532
astar_search h_pg_maxlevel	20	43	45	180	6	1,177270447
astar_search h_pg_setlevel	20	33	35	138	6	3,085952417

Figure 1. Air cargo problem 1

Air cargo problem 2:

The table bellow show the metrics of all possible algorithms solving the problem 2

	Actions	Expansions	Goal Tests	New Nodes	Plan length	Time elapsed
breadth_first_search	72	3343	4609	30503	9	2,008850647
depth_first_graph_search	72	624	625	5602	619	2,974004779
uniform_cost_search	72	5154	5156	46618	9	3,103015087
greedy_best_first_graph_search h_unmet_goals	72	17	19	170	9	0,018862413
greedy_best_first_graph_search h_pg_levelsum	72	9	11	86	9	10,2556034
greedy_best_first_graph_search h_pg_maxlevel	72	27	29	249	9	19,47962646
greedy_best_first_graph_search h_pg_setlevel	72	9	11	84	9	25,73791762
astar_search h_unmet_goals	72	2467	2469	22522	9	2,029838502
astar_search h_pg_levelsum	72	357	359	3426	9	254,392112
astar_search h_pg_maxlevel	72	2887	2889	26594	9	1491,354372
astar_search h_pg_setlevel	72	1037	1039	9605	9	2011,071961

Figure 2. Air cargo problem 2

Air cargo problem 3:

The table bellow show the metrics of the algorithms DFS, greedy best fist graph search unmet goals, greedy best fist graph search level sum, A* search unmet goals and A* search level sum to the problem 3.

	Actions	Expansions	Goal Tests	New Nodes	Plan length	Time elapsed
depth_first_graph_search	88	408	409	3364	392	1,161440023
greedy_best_first_graph_search h_unmet_goals	88	25	27	230	15	0,035885112
greedy_best_first_graph_search h_pg_levelsum	88	14	16	126	14	22,60318312
astar_search h_unmet_goals	88	7388	7390	65711	12	8,187521174
astar_search h_pg_levelsum	88	369	371	3403	12	406,7438995

Figure 3. Air cargo problem 3

Air cargo problem 4:

The table bellow show the metrics of the algorithms BFS (the DFS was chosen because the DFS was taken too long to finish), greedy best fist graph search unmet goals, greedy best fist graph search level sum, A* search unmet goals and A* search level sum to the problem 4.

	Actions	Expansions	Goal Tests	New Nodes	Plan length	Time elapsed
breadth_first_search	104	99736	114953	944130	14	95,60525513
greedy_best_first_graph_search h_unmet_goals	104	29	31	280	18	0,06006783
greedy_best_first_graph_search h_pg_levelsum	104	17	19	165	17	41,02808099
astar_search h_unmet_goals	104	34330	34332	328509	14	54,75283274
astar_search h_pg_levelsum	104	1208	1210	12210	15	2297,232126

Figure 4. Air cargo problem 4

2. Questions and Answers

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?

To this case the greedy best fist graph search unmet goals is the best choice since deliver a solution in a very small 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)

To this case the greedy best fist graph search level sum appears to be the best choice because it have the least **expansions** and least **new nodes** metric.

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

To this case the A* search unmet goals is the best choice since deliver the best solution in all four problems.