Report
Build a Forward Planning Agent

Mohammed Elshambakey

10/06/2018

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

Table 1: Nodes expanded vs. number of actions for different algorithms and heuristics

	Air Cargo Problem 1	Air Cargo Problem 2	Air Cargo Problem 3	Air Cargo Problem 4	Problem
Algorithm	**20**	**72**	**88**	**104**	**Actions**
breadth first search	43	3343	14663	99736	**Expansions**
depth first graph search	21	624	408	25174	
uniform cost search	60	5154	18510	113339	
greedy best first graph search with h unmet goals	7	17	25	29	
greedy best first graph search with h pg levelsum	6	9	14	17	
greedy best first graph search with h pg maxlevel	6	27	21	56	
greedy best first graph search with h pg setlevel	6	9	35	107	
astar search with h unmet goals	50	2467	7388	34330	
astar search with h pg levelsum	28	357	369	1208	
astar search with h pg maxlevel	43	2887	9580	62077	
astar search with h pg setlevel	33	1037	3423	22606	

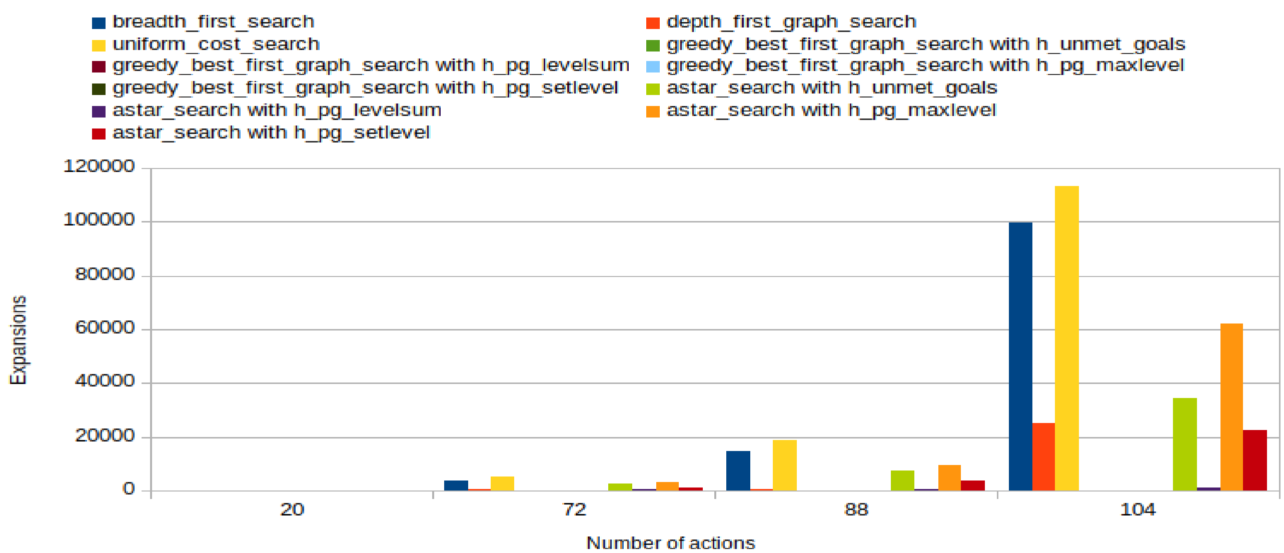

Figure 1: Nodes expanded vs. number of actions for different algorithms and heuristics

It is shown from Table 1, and Fig. 1, that number of expanded nodes increases with increasing domain size. Uniform Cost Search had the maximum number of expanded nodes. Greedy Best First Graph Search with different heuristics, especially LEVELSUM heuristic function, had the minimum number of expanded nodes.

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

Table 2: Search time vs. number of actions for different algorithms and heuristics

Problem	Air Cargo Problem 1	Air Cargo Problem 2	Air Cargo Problem 3	Air Cargo Problem 4	
Actions	20	72	88	104	Algorithm
Time (Sec)	0.233186958000033	9.32208232999994	16.7975480700001	94.7388729539998	breadth first search
	0.122530521000044	52.990589304	307.966252391	3156.024047635	depth first graph search
	0.30208527000002	120.21959184	693.805046097	7389.849309276	uniform cost search
	0.015042947000097	0.570422437000047	0.770595014000037	3.88743919199987	greedy best first graph search with h unmet goals
	0.015943917999948	0.270107745000018	0.669564173000026	5.00919776700016	greedy best first graph search with h pg levelsum
	0.00516289799998	0.450499178999962	0.146837311000013	965.863128274	greedy best first graph search with h pg maxlevel
	0.218474410999988	0.369129243999964	0.863370798999995	1.40851894200023	greedy best first graph search with h pg setlevel
	0.103391274000046	0.634742369000037	0.835418521999941	2.59617345700008	astar search with h unmet goals
	0.551640850000013	1.42796521399998	7.44061273200009	34.665185457	astar search with h pg levelsum
	0.00138369299998	0.008520957999963	0.013813171000038	0.023433104000105	astar search with h pg maxlevel
	0.013565766000056	0.545184608999989	1.10464959000001	7.64540633199977	astar search with h pg setlevel

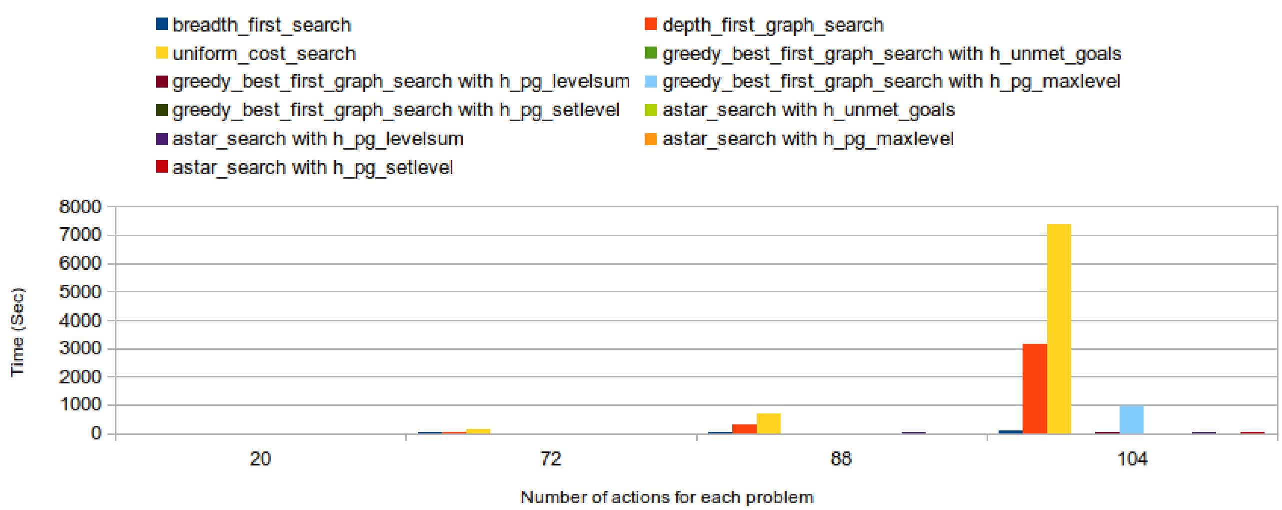

Fig 2: Search time vs. number of actions for different algorithms and heuristics

It is shown in Table 2, and Fig. 2, that search time increases with increasing domain size. Problem 1, with 20 actions, consumed the shortest search time among all algorithms, while problem 4, with 104 actions, consumed the longest search time. For each problem, A* with MAXLEVEL heuristic consumed the shortest search time, while Uniform Cost Search consumed the longest search time in problems 2, 3, and 4. A* with LEVELSUM heuristic consumed the longest search time in problem 1. In general, Depth-First and Breadth-First consume longer time than other algorithms. Heuristics effect on search time of Greedy Best First and A* differ according to problem size (i.e., the ascending order of search times between Greedy Best First and A*, combined with different heuristics, change with the problem size).

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

Table 3: Plan length vs. number of actions for different algorithms and heuristics

	Air Cargo Problem 1	Air Cargo Problem 2	Air Cargo Problem 3	Air Cargo Problem 4	Problem
Algorithm	**20**	**72**	**88**	**104**	**Actions**
breadth first search	6	9	12	14	**Plan length**
depth first graph search	20	619	392	24132	
uniform cost search	6	9	12	14	
greedy best first graph search with h unmet goals	6	9	15	18	
greedy best first graph search with h pg levelsum	6	9	14	17	
greedy best first graph search with h pg maxlevel	6	9	13	17	
greedy best first graph search with h pg setlevel	6	9	17	23	
astar search with h unmet goals	6	9	12	14	
astar search with h pg levelsum	6	9	12	15	
astar search with h pg maxlevel	6	9	12	14	
astar search with h pg setlevel	6	9	12	14	

It is shown from Table 3 that plan length increases with increasing domain size. It is also shown that Depth-First produced the longest plan for all problems. Algorithms other than Depth-First produced equal plan lengths for problems 1 and 2. With increasing domain size (i.e., problems 3 and 4), Breadth-First, Uniform-cost, and A* produced the shortest plans.

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?

A* with MAXLEVEL heuristic consumed the shortest time among all algorithms for all problems. Additionally, A* with MAXLEVEL is one of the algorithms that produced the shortest plan. Thus, A* with MAXLEVEL would be appropriate for planning in real-time for a very restricted domain

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)?

A* with MAXLEVEL heuristic consumed the shortest time among all algorithms for all problems despite A*, with MAXLEVEL, made many expansions and generated many new nodes. Additionally, A* with MAXLEVEL is one of the algorithms that produced the shortest plans for all problems. Thus, A* with MAXLEVEL would be appropriate for planning for very large domains.

Greedy Best First search can also be appropriate for large domains as Greedy Best First Search (with different heuristics) produced the minimum expansions and new nodes. However, Greedy Best First Search (with different heuristics) consumed longer time than A* with MAXLEVEL heuristic.

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

Uniform Cost, and Breadth First (as step costs are all identical). A* is implemented as graph-based, so it needs consistent (not just admissible) heuristics to be optimal.