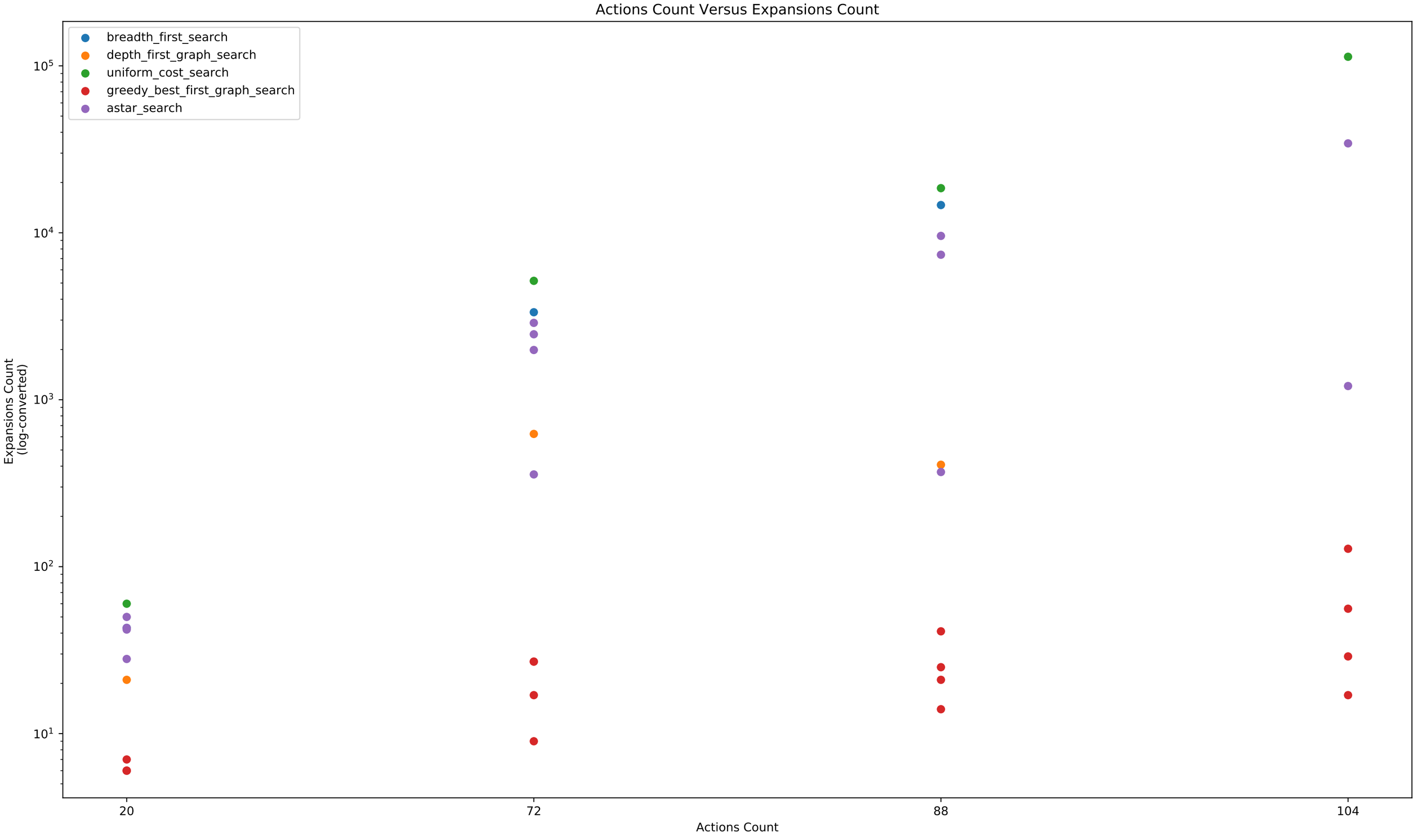
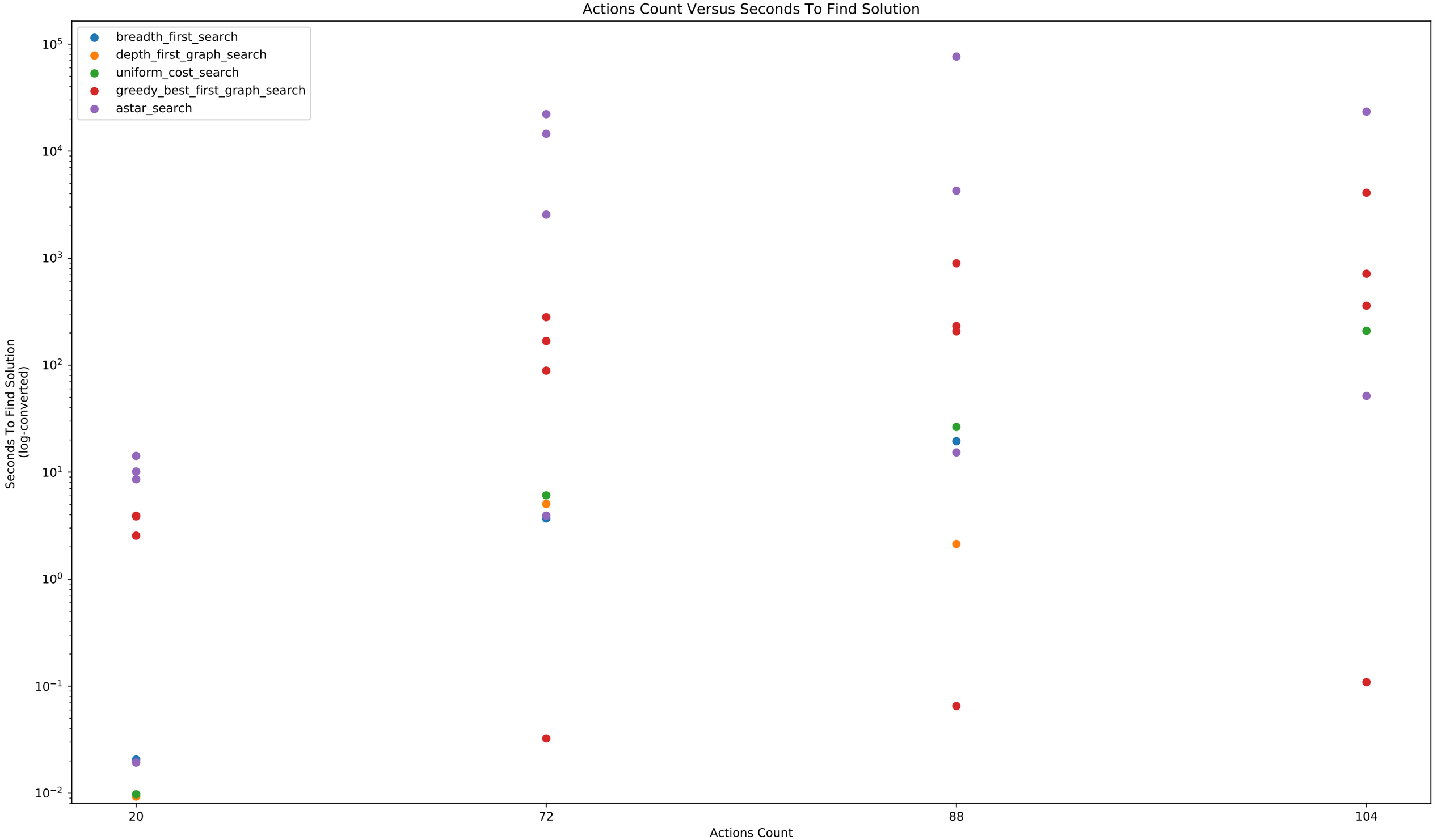
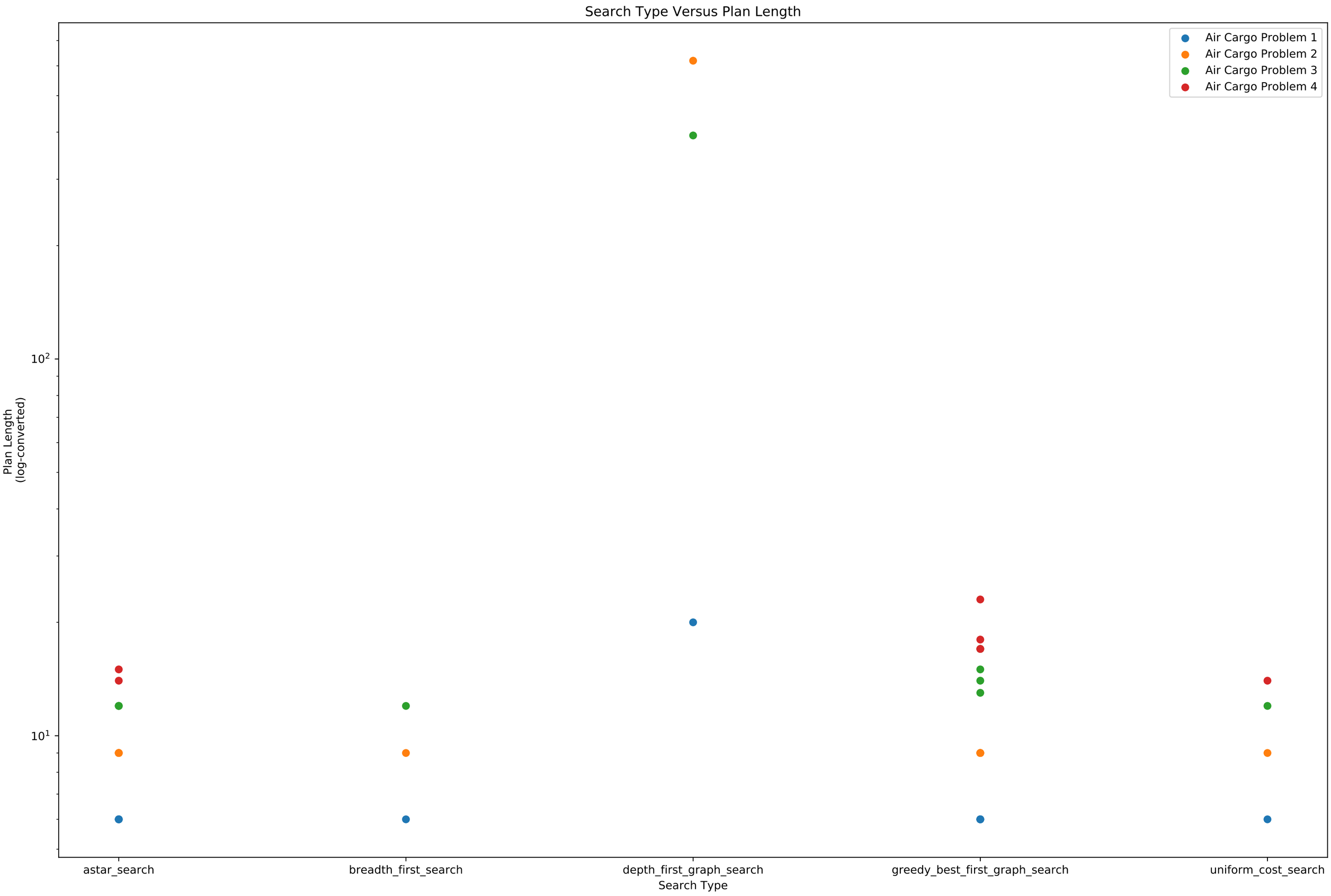


All search algorithms except depth-first and greedy-best-first search increase the number of nodes expanded exponentially as the number of actions taken to find a solution increases. Depth-first and greedy-best-first search algorithms seem to increase more linearly than the rest of the search algorithms in regards to the number of expansions increasing the number of actions taken to find a solution increases.



Generally and for all search algorithms, as the number of actions taken before finding a solution increases, the time to find the optimal solution increases exponentially. A-star and breadth-first searches seem most reliable in finding the optimal solution the fastest for two reasons: 1) there exists at least one data point for each of A-star and breadth-first searches that is close to or below the average time taken to find a solution for each set of data points plotted for each number of actions taken to find a solution. 2) A-star and breadth-first searches always find the optimal solution when a solution is found. Greedy-best-first, uniform-cost, and depth-first searches are shown to find some solution the fastest but may not have found the optimal solution.


Search Type Versus Plan Length
Air Cargo Problem 1
Air Cargo Problem 2
Air Cargo Problem 3
Air Cargo Problem 4
Plan Length
(log-converted)
10^2
10^1
astar_search
breadth_first_search
depth_first_graph_search
greedy_best_first_graph_search
uniform_cost_search
Search Type

Algorithm Analysis

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?

When very few actions are taken to find a solution (e.g. 20 actions), uniform-cost search finds the optimal solution the fastest when compared with the other search algorithms graphed in the previous pages and most reliably due to uniform-cost search finding the optimal solution when a solution is found.

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)

When planning in very large domains (e.g. domains that require more than 72 actions to find a solution), greedy-best-first search is the most appropriate to use due to it finding a solution the fastest when compared with all of the other search algorithms graphed in the above pages.

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

A-star search is the most appropriate search algorithm when finding only optimal plans is important to use due to it finding the optimal solution in or below average time when compared with all of the other search algorithms graphed in the above pages.