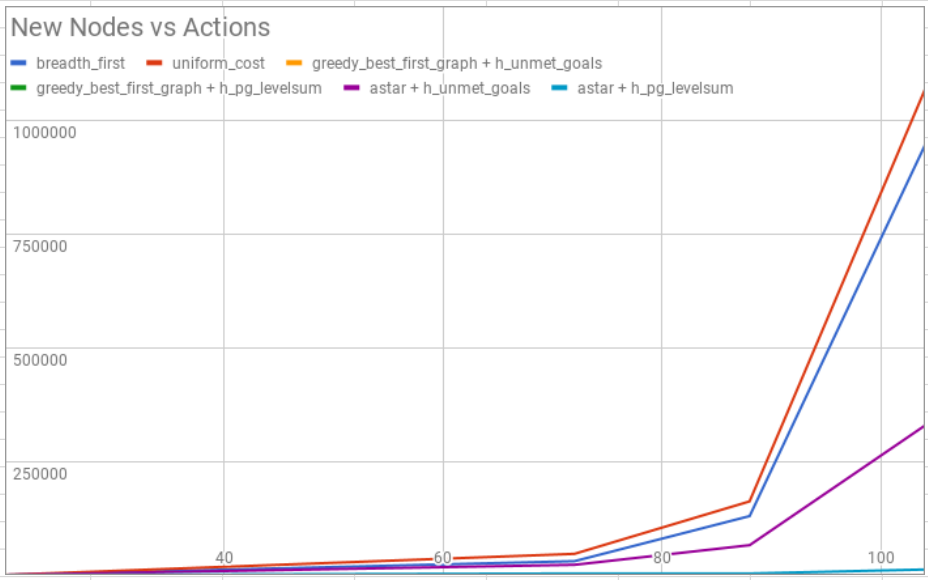
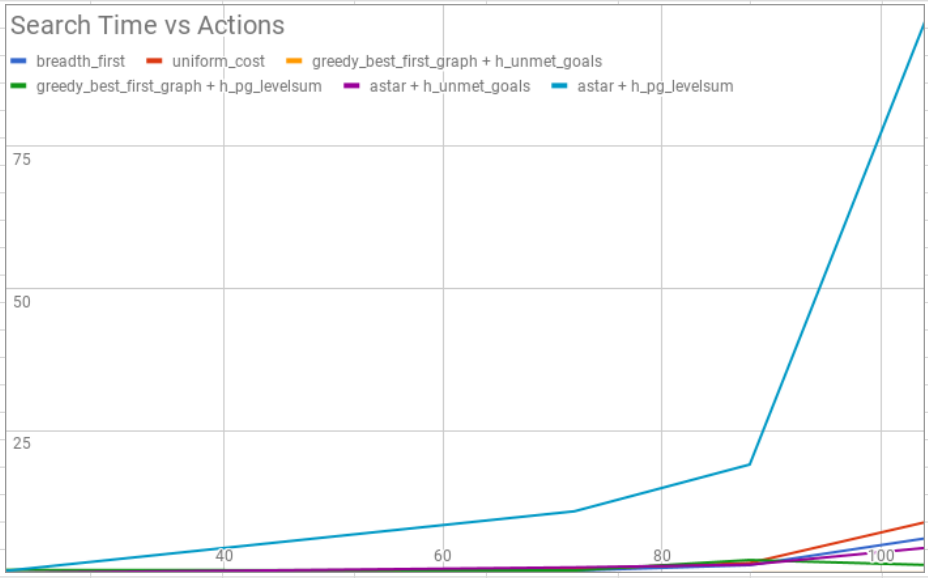
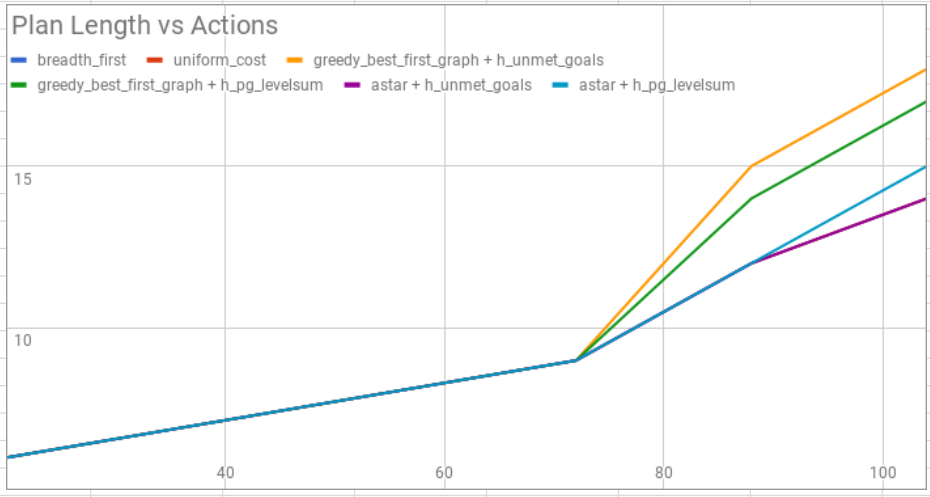


The number of new nodes may represent the search complexity of the algorithm, a large number of nodes may result in an increase in space complexity or searching time. For the algorithms of breath first, uniform cost, the number of nodes grow fast when problem domain become larger. On the other hand, all greedy algorithms and A* with levelsum heuristics, the number of nodes grow very slowly. For the A* algorithm with unmet goals heuristics, it has a moderate speed of growth.



Searching time represents the time complexity of the algorithm. The greedy series run really fast even when the domain become larger. The searching time of A* algorithm with levelsum heuristics increase very fast when problem domain grows. The other algorithms go between them.



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?
I think we can go with uniform cost search or a* with unmet goals heuristic algorithm, considering the number of actions is limited, search complexity will not increase so dramatically, and search time is the most critical factor in such a 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)
The a* algorithm with unmet goals heuristic would be appropriate for such a domain, because the search complexity/search time grow in a relatively lower speed when the actions increase, in the meanwhile the algorithm guarantee to find the optimal plan.
However the a* algorithm with levelsum heuristic would be even better, if we can allow a less optimal plan, for the the search complexity/search time grow really slowly even when the problem domain become very large.

Which algorithm or algorithms would be most appropriate for planning problems where it is important to find only optimal plans?
We can pick the a* algorithm with unmet goals heuristic, it has a better balance on search complexity and speed, while guarantee to find the optimal plan. If the problem domain is not that large, we can also choose the uniform cost search algorithm.