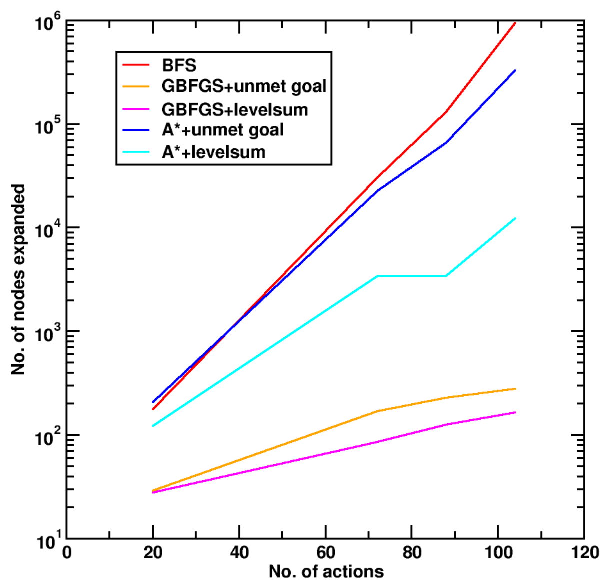
1. Number of nodes expanded versus number of actions

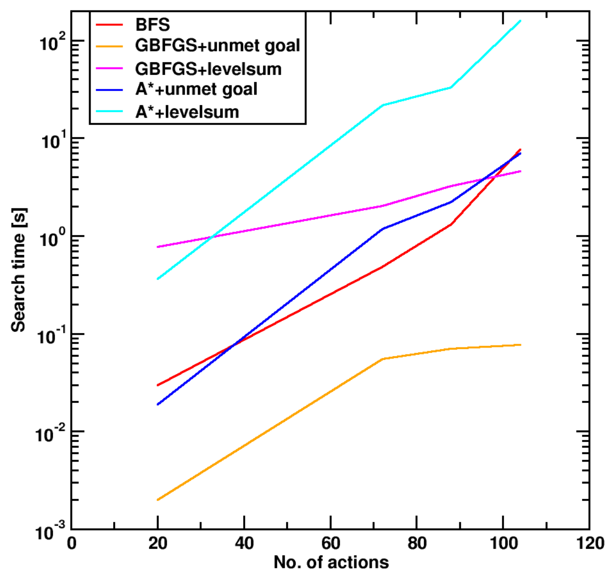
	Problem1	Problem2	Problem3	Problem4
	No of actions			
	20	72	88	104
Algorithm	No of nodes expanded			
BFS	178	30503	129625	944130
DFS	84	5602		
UCS	240	46618		
GBFGS+unmetG	29	170	230	280
GBFGS+levelsum	28	86	126	165
GBFGS+maxlevel	24	249		
GBFGS+setlevel	28	84		
A*+unmetG	206	22522	65711	328509
A*+levelsum	122	3426	3403	12210
A*+maxlevel	180	26594		
A*+setlevel	138	9605		



Overall, the number of nodes need to be expanded grow exponentially with respect to the number of actions no matter what algorithm it is. However, we can see from the figure that greedy best first graph search grows much slower than uninformed search and A* search. In addition, regarding four different heuristics, level-sum and set-level can bring down the number of nodes expanded significantly. Among three uninformed searches, the depth first search expands much less nodes than the breadth first search and the uniform cost search.

2. Search time versus number of actions

	Problem1	Problem2	Problem3	Problem4
	No of actions			
	20	72	88	104
Algorithm	Search time [s]			
BFS	0.030	0.486	1.306	7.705
DFS	0.009	0.746		
UCS	0.024	1.057		
GBFGS+unmetG	0.002	0.055	0.070	0.077
GBFGS+levelsum	0.775	2.031	3.232	4.569
GBFGS+maxlevel	0.235	2.862		
GBFGS+setlevel	1.013	4.354		
A*+unmetG	0.019	1.185	2.214	7.038
A*+levelsum	0.362	21.739	33.322	159.220
A*+maxlevel	0.255	123.693		
A*+setlevel	0.708	161.108		



Overall, the search time grows exponentially with respect to the number of actions. Greedy best first graph search with unmet goal heuristic provides the best search time, while A* search with set-level heuristic yields the worst search time. In general, the time spent on search for three categories of algorithm is greedy best first graph search > uninformed search > A* search. Among four heuristics, unmet goal gives the best search time while set-level is the worst. The greedy best first graph search class of algorithms have the slowest grow rate with respect to system size, indicating their potential advantages to very large systems.

3. Length of plan

	Problem1	Problem2	Problem3	Problem4
Algorithm	Length of plan			
BFS	6	9	12	14
DFS	20	619		
UCS	6	9		
GBFGS+unmetG	6	9	15	18
GBFGS+levelsum	6	9	14	17
GBFGS+maxlevel	6	9		
GBFGS+setlevel	6	9		
A*+unmetG	6	9	12	14
A*+levelsum	6	9	12	15
A*+maxlevel	6	9		
A*+setlevel	6	9		

Among the three uninformed searches, the depth first search cannot yield the optimal plan while the other two can. Greedy best first graph search is not guaranteed to give the optimal plan. A* search can provide near optimal plan.

4. 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?

When the action domain is small, uninformed search such as breadth first search and uniform cost search would be appropriate, since they are simple and fast.

(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)

If the optimal solution is not required, greedy best first graph search family of algorithm would be most appropriate. With the unmet goal heuristic, it has the fastest search speed.

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

Uninformed search including breadth first search and uniform cost search. In addition, A* search with appropriate heuristic such as unmet goals.