I. Optimal Plans and Search Results for Problems

PROBLEM - 1						
Search	Expansions	Goal Tests	New Nodes	Plan length	Time elapsed in seconds	Optimal Plan
breadth_first_search	43	56	180	6	0.044	YES
breadth_first_tree_search	1458	1459	5960	6	1.339	YES
depth_first_graph_search	21	22	84	20	0.021	NO
depth_limited_search	101	271	414	50	0.139	NO
uniform_cost_search	55	57	224	6	0.053	YES
recursive_best_first_search h_1	4229	4230	17023	6	3.99	YES
greedy_best_first_graph_search h_1	7	9	28	6	0.008	YES
astar_search h_1	55	57	224	6	0.053	YES
astar_search h_ignore_preconditions	41	43	170	6	0.054	YES
astar_search h_pg_levelsum	11	13	50	6	1.266	YES
OPTIMAL PLAN FOR PROBLEM - 1						
Load(C1, P1, SFO) Load(C2, P2, JFK) Fly(P1, SFO, JFK) Fly(P2, JFK, SFO) Unload(C1, P1, JFK) Unload(C2, P2, SFO)						

- Except depth-first graph and depth-limited searches, search algorithms find optimal plan.
- For this basic problem, Greedy best-first graph search with h1 algorithm finds optimal solution in the shortest search time.
- A-star search with h_1 and Uniform-cost search algorithm has the same result in this problem.
- Heuristic function of A-star search returns same value 1, so path cost is important for node selection.

PROBLEM - 2						
Search	Expansions	Goal Tests	New Nodes	Plan length	Time elapsed in seconds	Optimal Plan
breadth_first_search	3343	4609	30509	9	19.357	YES
breadth_first_tree_search	NO RESULT IN 20 MINUTES					
depth_first_graph_search	624	625	5602	619	4.567	NO
depth_limited_search	222719	2053741	2054119	50	1374.941	NO
uniform_cost_search	4835	4837	43877	9	17.316	YES
recursive_best_first_search h_1	NO RESULT IN 10 MINUTES					
greedy_best_first_graph_search h_1	958	960	8622	21	3.392	NO
astar_search h_1	4835	4837	43877	9	17.71	YES
astar_search h_ignore_preconditions	1450	1452	13303	9	6.1	YES
astar_search h_pg_levelsum	86	88	841	9	199.203	YES
OPTIMAL PLAN FOR PROBLEM - 2						
Load(C1, P1, SFO) Load(C2, P2, JFK) Load(C3, P3, ATL) Fly(P2, JFK, SFO) Unload(C2, P2, SFO) Fly(P1, SFO, JFK) Unload(C1, P1, JFK) Fly(P3, ATL, SFO) Unload(C3, P3, SFO)						

- In this problem, best performance is seen at A-star search with h_ignore_preconditions.
- A-star search with h_pg_levelsum find optimal plan with the least expansions, goal tests and new nodes distinctively. However, it requires more computation so this increases elapsed time.
- Again, result of A-star search with h_1 and Uniform-cost search algorithm is same.
- Breadth-first tree search and recursive best-first search with h1 did not complete search in more than 20 minutes.
- Depth-limited search took enormous time and could not find optimal plan in this problem.

PROBLEM - 3						
Search	Expansions	Goal Tests	New Nodes	Plan length	Time elapsed in seconds	Optimal Plan
breadth_first_search	14663	18098	129631	12	136.848	YES
breadth_first_tree_search	NO RESULT IN 20 MINUTES					
depth_first_graph_search	408	409	3364	392	2.411	NO
depth_limited_search	NO RESULT IN 20 MINUTES					
uniform_cost_search	18225	18227	159630	12	75.033	YES
recursive_best_first_search h_1	NO RESULT IN 20 MINUTES					
greedy_best_first_graph_search h_1	5578	5580	49147	13	24.183	NO
astar_search h_1	18225	18227	159630	12	76.637	YES
astar_search h_ignore_preconditions	5040	5042	44944	12	23.305	YES
astar_search h_pg_levelsum	309	311	2845	12	1366.66	YES
OPTIMAL PLAN FOR PROBLEM - 3						
Load(C1, P1, SFO) Load(C2, P2, JFK) Fly(P2, JFK, ORD) Load(C4, P2, ORD) Fly(P1, SFO, ATL) Load(C3, P1, ATL) Fly(P1, ATL, JFK) Unload(C1, P1, JFK) Unload(C3, P1, JFK) Fly(P2, ORD, SFO) Unload(C2, P2, SFO) Unload(C4, P2, SFO)						

- A-star search with ignoring preconditions far and away the best performer in this problems in terms of time.
- Uniform cost is better than breadth-first in this problem in terms of execution time.
- Again, result of A-star search with h_1 and Uniform-cost search algorithm is same.
- Breadth-first tree search, depth-limited search and recursive best-first search with h1 did not complete search in more than 20 minutes.

II. Comparison of Non-heuristic Searches

BFS (breadth_first_search) and UCS (uniform_cost_search) find optimal plans for all problems. When complexity is problem increasing, elapsed time is also increasing exponentially. They guarantee to reach optimal plan, BFS stops when it found optimal solution and UCS continues to last depth level.

BFTS (breadth_first_tree_search) find optimal plan just for Problem-1 due to the least complexity. In other problems, together with DLS (depth_limited_search), due to the cycles of repetitive states.

DFGS (depth_first_graph_search) visit the least nodes in all three problems, but it is far away from optimal plan. DFGS stops at first when it reached to solution, whenever optimal or not.

Among these BFS and UCS are best performers, UCS is keen to get better than BFS in more complicated problems.

III. Comparison of Heuristic Searches

All heuristic searches found optimal plan solutions. Also expanded nodes are less than non-heuristic searches.

A-star search with ignore preconditions is much faster although it performs more expansions, goal tests and new nodes than A-star search with level sum.

A-star search with level sum is more precise but it takes more time to compute.

IV. Result

For small problems, breadth_first_search will be the appropriate choice. For larger problems, A-start search with relaxed preconditions will be useful for larger problems. If there is limitation for expansion, A-star search with refined heuristic is able to find optimal solution.

In addition, Norvig and Russell's textbook also stated that "It seems likely that further progress on larger problems cannot rely only on factored and propositional representations, and will require some kind of synthesis of first-order and hierarchical representations with the efficient heuristics currently in use."