Project: Planning Search - Written Analysis

Optimal Plans:

The problems given belong to the Air Cargo Domain and have the following action schema defined:

```
Action(Load(c, p, a),
       PRECOND: At(c, a) ∧ At(p, a) ∧ Cargo(c) ∧ Plane(p) ∧ Airport(a)
       EFFECT: ¬ At(c, a) ∧ In(c, p))
Action(Unload(c, p, a),
       PRECOND: In(c, p) ∧ At(p, a) ∧ Cargo(c) ∧ Plane(p) ∧ Airport(a)
       EFFECT: At(c, a) ∧ ¬ In(c, p))
Action(Fly(p, from, to),
       PRECOND: At(p, from) ∧ Plane(p) ∧ Airport(from) ∧ Airport(to)
       EFFECT: ¬ At(p, from) ∧ At(p, to))
```

An optimal plan -sequence of actions- for each of the given problems are as follows:

Problem	Initial State and Goal	Optimal Plan

1	`Init(At(C1, SFO) ∧ At(C2, JFK)`
	`∧ At(P1, SFO) ∧ At(P2, JFK)`
	`∧ Cargo(C1) ∧ Cargo(C2)`
	`∧ Plane(P1) ∧ Plane(P2)`
	`∧ Airport(JFK) ∧ Airport(SFO))`
	`Goal(At(C1, JFK) ∧ At(C2, SFO))`
		Load(C2, P2, JFK)
		Load(C1, P1, SFO)
		Fly(P2, JFK, SFO)
		Unload(C2, P2, SFO)
		Fly(P1, SFO, JFK)
		Unload(C1, P1, JFK)

2	`Init(At(C1, SFO) ∧ At(C2, JFK) ∧ At(C3, ATL)`
	`∧ At(P1, SFO) ∧ At(P2, JFK) ∧ At(P3, ATL)`
	`∧ Cargo(C1) ∧ Cargo(C2) ∧ Cargo(C3)`
	`∧ Plane(P1) ∧ Plane(P2) ∧ Plane(P3)`
	`∧ Airport(JFK) ∧ Airport(SFO) ∧`
	`Airport(ATL))`
	`Goal(At(C1, JFK) ∧ At(C2, SFO) ∧ At(C3, SFO))`
		Load(C2, P2, JFK)
		Load(C1, P1, SFO)
		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)

3	`Init(At(C1, SFO) ∧ At(C2, JFK) ∧ At(C3, ATL) ∧`
	`At(C4, ORD)`
	`∧ At(P1, SFO) ∧ At(P2, JFK)`
	`∧ Cargo(C1) ∧ Cargo(C2) ∧ Cargo(C3) ∧`
	`Cargo(C4)`
	`∧ Plane(P1) ∧ Plane(P2)`
	`∧ Airport(JFK) ∧ Airport(SFO) ∧`
	`Airport(ATL) ∧ Airport(ORD))`
	`Goal(At(C1, JFK) ∧ At(C3, JFK) ∧ At(C2, SFO) ∧`
	`At(C4, SFO))`
		Load(C2, P2, JFK)
		Load(C1, P1, SFO)
		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)

Performance Comparison:

The performance comparison for non-heuristic and heuristic planning searches was analyzed in term of **time efficiency** (execution time measured in seconds), **optimality** (Yes/No if the solution was the most optimized one which on this problem was equivalent to a solution with the shortest path length) and **memory efficiency** (measured in number of expansions).

Non-heuristic searches (uninformed algorithms):

Breadth first, depth first and uniform cost searches were compared and contrasted to solve planning problems 1,2 and 3. The results are shown in Fig 1.

Problem	Search	Expansions	Goal Tests	New Nodes	Time (seconds)	Plan lenght	Optimal
1	depth_first_graph_search	12	13	48	0.018	12	No
1	breadth_first_search	43	56	180	0.048	6	Yes
1	uniform_cost_search	55	57	224	0.053	6	Yes
2	depth_first_graph_search	582	583	5211	4.607	575	No
2	breadth_first_search	3343	4609	30509	17.801	9	Yes
2	uniform_cost_search	4835	4837	43877	15.471	9	Yes
3	depth_first_graph_search	627	628	5176	3.982	596	No
3	breadth_first_search	14663	18098	129631	16.580	12	Yes
3	uniform_cost_search	18234	18236	159707	66.547	12	Yes

Fig. 1: Result metrics for non-heuristic searches

The most time and memory efficient search for all problems was the depth first search (as it required the lowest number of node expansions until reaching to a solution), but it provided a non-optimal solution (its solution path was larger in term of number of actions needed until reaching the goal). On the contrary, breadth first search and uniform cost search were less time efficient and more memory consuming but their solutions were optimal.

Those results are expected for those searching algorithms based on the following characteristics:

- Breadth first search is guaranteed to return the shallowest path in the search tree, which on this problem is also equivalent to optimality, as all actions (edges) have the same cost and therefore it returned the shortest path [1]. The shortest path had 6, 9 and 12 actions for problems 1,2 and 3 respectively.
- Uniform cost search is guaranteed to return a path which is optimal in terms of cost [2]. On this problem, as discussed above, the cost search is equal to the number of nodes in the path and therefore, the number of nodes of the optimal solution is equal to the number of nodes of the shortest possible path.
- Depth first search always expands the deepest node [3] and returns the leftmost possible path in the search tree, therefore it does not guarantee to return the shortest path, it just returns the first solution if finds and hence its high speed. As we have observed on this problem, the obtained plan length increased with the problem complexity reaching up to 596 actions for problem 3 which roughly approximated to the number of node expansions.

Heuristic searches (informed algorithms):

A* search using the ignore preconditions and level sum heuristics were compared and contrasted to solve planning problems 1,2 and 3. The results are shown in Fig 2.

Problem	Search	Expansions	Goal Tests	New Nodes	Time (seconds)	Plan lenght	Optimal
1	astar_search with h_ignore_preconditions	41	43	170	0.052	6	Yes
1	astar_search with h_pg_levelsum	11	13	50	0.768	6	Yes
2	astar_search with h_ignore_preconditions	1450	1452	13303	5.368	9	Yes
2	astar_search with h_pg_levelsum	86	88	841	70.608	9	Yes
3	astar_search with h_ignore_preconditions	5040	5042	44944	20.539	12	Yes
3	astar_search with h_pg_levelsum	316	318	2914	330.226	12	Yes

Fig. 2: Result metrics for A* planning searches using ignore preconditions and level sum heuristics.

Both implemented heuristics were admissible -they never overestimated the distance from a state to the goal node- and hence they provided an optimal solution where the resulting path had the smallest possible number of actions to reach the goal (6,9 and 12 for problems 1,2 and 3 respectively).

Ignore preconditions heuristic was the most time efficient informed search for all 3 problems.
Level sum heuristic was the most memory efficient algorithm requiring the lowest number of node expansions. In spite of it, it was more time consuming, which could be explained by the computational time required to build at each state a planning graph representation of the problem state space. This planning graph representation estimates the sum of all actions that must be carried out from the current state in order to satisfy each individual goal condition. The complexity of graph generation is polynomial in number of literals [4]. On contrast, the ignore precondition implementation only has to compute at each state the count of unsatisfied goals if we accept the subgoal independence assumption.[5]

Comparing both uninformed vs informed algorithms, the fastest one to achieve a solution is the depth first search, but if an optimal solution is required, then the fastest algorithm is A* search with ignore preconditions heuristic.

The metrics obtained showed that the use of heuristics -estimating how far the goal state is- improve the efficiency both in term of memory usage and time cost of search problems when searching for an optimal plan. This efficiency gain is the consequence of having to expand less of the search tree and visit less nodes [6]. On the performance analysis we have shown, that informed planning algorithms using A* search with ignore preconditions heuristic returned the optimal path faster than the uniform cost search. In addition, savings in time were more pronounced with increasing problem complexity as we appreciate in problems from 1 to 3 (P1,P2,P3) comparing uniform cost search vs A* search with ignore preconditions (P1: 0.053 vs 0.052, P2: 15.4 vs 5.3 and P3: 66.5 vs 20.5).

[1] Artificial Intelligence: A Modern Approach by Stuart Russell, Peter Norvig. Section 3.4.1 Breadth-first search
[2] Artificial Intelligence: A Modern Approach by Stuart Russell, Peter Norvig. Section 3.4.3 Depth-first search
[3] Artificial Intelligence: A Modern Approach by Stuart Russell, Peter Norvig. Section 3.4.2 Uniform-cost search
[4] Artificial Intelligence: A Modern Approach by Stuart Russell, Peter Norvig. Section 10.3 Planning Graphs
[5] Artificial Intelligence: A Modern Approach by Stuart Russell, Peter Norvig. Section 10.2.3 Heuristics for planning
[6] Hart, P. E.; Nilsson, N. J.; Raphael, B. (1968). "A Formal Basis for the Heuristic Determination of Minimum Cost Paths". IEEE Transactions on Systems Science and Cybernetics SSC4. 4 (2): 100–107. doi:10.1109/TSSC.1968.300136.