Heuristic Analysis

Optimal plan

The length of optimal plan for problem 1, 2 and 3 are 6, 9 and 12 respectively. And the solutions are

Problem1	Problem2	Problem3
$\boldsymbol{Load(C_1, P_1, SFO)}$	$Load(C_1, P_1, SFO)$	$Load(C_1, P_1, SFO)$
$\boldsymbol{Load(C_2, P_2, JFK)}$	$Load(C_2, P_2, JFK)$	$Load(C_2, P_2, JFK)$
$\boldsymbol{Fly(P_1, SFO, JFK)}$	$Load(C_3, P_3, ATL)$	$Fly(P_1, SFO, ATL)$
$\boldsymbol{Fly(P_2, JFK, SFO)}$	$Fly(P_1, SFO, JFK)$	$Load(C_3, P_1, ATL)$
$\boldsymbol{Unload(C_1, P_1, JFK)}$	$Fly(P_2, JFK, SFO)$	$Fly(P_2, JFK, ORD)$
$\boldsymbol{Unload(C_2, P_2, SFO)}$	$Fly(P_3, ATL, SFO)$	$Load(C_4, P_2, ORD)$
	$Unload(C_1, P_1, JFK)$	$Fly(P_1, ATL, JFK)$
	$Unload(C_2, P_2, SFO)$	$Fly(P_2, ORD, SFO)$
	$Unload(C_3, P_3, SFO)$	$Unload(C_1, P_1, JFK)$
		$Unload(C_2, P_2, SFO)$
		$Unload(C_3, P_1, JFK)$
		$Unload(C_4, P_2, SFO)$

Uninformed search

Problem 1

Search strategy	Optimal	Plan length	Node expansions	Number of goal tests	Time elapsed(s)
Breadth first	Y	6	43	56	0.032
Depth first	N	20	21	22	0.015
Uniform cost	Y	6	55	57	0.041
Greedy best first	Y	6	7	9	0.005

Problem 2

Search strategy	Optimal	Plan length	Node expansions	Number of goal tests	Time elapsed(s)
Breadth-first	Y	9	3346	4612	9.331
Depth-first	N	105	107	108	0.336
Uniform cost	Y	9	4853	4855	12.975
Greedy best first	N	21	998	1000	2.501

Problem 3

Search strategy	Optimal	Plan length	Node expansions	Number of goal tests	Time elapsed(s)
Breadth-first	Y	12	14120	17673	42.903
Depth-first	N	288	292	293	1.193
Uniform cost	Y	12	18223	18225	55.916
Greedy best first	N	22	5578	5580	17.776

In all the uninformed search cases, only breadth-first and uniform cost search can yield optimality. Between them, Breadth-first search performs better than uniform cost search on node expansions, number of goal tests and search time. The result is expected since when all step costs are equal, the cost of uniform search is $O(b^{1+d})$, which is larger than breadth-first search.

Criterion	Breadth-First	Uniform-Cost	Depth-First	Depth-Limited	Iterative Deepening	Bidirectional (if applicable)
Complete?	Yes[a]	Yes[a,b]	No	No	Yes[a]	Yes[a,d]
Time	$O(b^d)$	$O(b^{1+\lfloor C^*/\epsilon \rfloor})$	$O(b^m)$	$O(b^\ell)$	$O(b^d)$	$O(b^{d/2})$
Space	$O(b^d)$	$O(b^{1+\lfloor C^*/\epsilon \rfloor})$	$O(bm)$	$O(b\ell)$	$O(bd)$	$O(b^{d/2})$
Optimal?	Yes[c]	Yes	No	No	Yes[c]	Yes[c,d]

Figure 3.21 Evaluation of tree-search strategies. b is the branching factor; d is the depth of the shallowest solution; m is the maximum depth of the search tree; l is the depth limit. Superscript caveats are as follows: [a] complete if b is finite; [b] complete if step costs $\geq \epsilon$ for positive ϵ; [c] optimal if step costs are all identical; [d] if both directions use breadth-first search.

Informed search

Problem 1

Search strategy	Optimal	Plan length	Node expansions	Number of goal tests	Time elapsed(s)
A* with h1	Y	6	55	57	0.045
A* with ignore	Y	6	41	43	0.037
A* with level sum	Y	6	11	13	1.567

Problem 2

Search strategy	Optimal	Plan length	Node expansions	Number of goal tests	Time elapsed(s)
A* with h1	Y	9	4853	4855	12.240
A* with ignore	Y	9	1450	1452	4.340
A* with level sum	Y	9	86	88	204.522

Problem 3

Search strategy	Optimal	Plan length	Node expansions	Number of goal tests	Time elapsed(s)
A* with h1	Y	12	18223	18225	56.071
A* with ignore	Y	12	5040	5042	17.543
A* with level sum	Y	12	325	327	1089.815

For informed search, all three strategies yield optimality, the best strategy for search time is A* with ignore preconditions; for node expansions and number of goal tests, A* with level sum outperforms other strategies, this shows planning graph can better guide the A*. However, it does have a drawback, that is, it takes too much time, and the reason is the program needs $O(n(a+l)^2)$ time to construct planning graph. Without further optimizing the algorithm, the clear choice for our planning problems is A* with ignore precondition.