Heuristic Analysis – Planning Search

Daniel Meechan

Udacity Artificial Intelligence Nanodegree

Term 1, Project 3 (November 2017)

For project 3, I implemented a search planning agent to solve deterministic air cargo transport logistics problems. The tables below compare multiple uninformed and informed search algorithms. The best solution for the problem (the one which found the optimal solution in the fastest time) is highlighted in bold.

Uninformed search (without heuristics)

Problem 1:

Search algorithm	Best solution?	Optimal	Expansions	Goal Tests	Plan length	Time
Breadth first search	No	Yes	43	56	**6**	0.031
Breadth first tree search		Yes	1458	1459	**6**	0.825
Depth first graph search		No	12	13	12	0.009
Depth limited search		No	101	271	50	0.085
Uniform cost search		Yes	55	57	**6**	0.040
Recursive best first search		Yes	4229	4230	**6**	2.870
Greedy best first graph search	**Yes**	**Yes (but not always)**	**7**	**9**	**6**	**0.004**

Due to the higher search time required for problems 2 and 3, I chose to test only 4 of the search algorithms those two problems.

Problem 2:

Search algorithm	Best solution?	Optimal	Expansions	Goal Tests	Plan length	Time
Breadth first search	**Yes**	**Yes**	**3343**	**4609**	**9**	**7.141**
Depth first graph search	No	No	582	583	575	2.887
Uniform cost search		Yes	4853	4855	**9**	11.222
Greedy best first graph search		No	998	1000	21	2.125

Problem 3:

Search algorithm	Best solution?	Optimal	Expansions	Goal Tests	Plan length	Time
Breadth first search	**Yes**	**Yes**	**12358**	**15077**	**12**	**32.629**
Depth first graph search	No	No	4315	4316	815	18.898
Uniform cost search		Yes	15601	15603	**12**	41.816
Greedy best first graph search		No	5297	5299	24	13.334

Informed search (using heuristics)

Problem 1:

Search algorithm	Best solution?	Optimal	Expansions	Goal Tests	Plan length	Time
A* with h1	No	Yes	55	57	**6**	0.033
A* with ignore preconditions	**Yes**	**Yes**	**41**	**43**	**6**	**0.031**
A* with level sum	No	Yes	11	13	**6**	0.932

Problem 2:

Search algorithm	Best solution?	Optimal	Expansions	Goal Tests	Plan length	Time
A* with h1	No	Yes	4853	4855	**9**	11.680
A* with ignore preconditions	**Yes**	**Yes**	**1450**	**1452**	**9**	**4.481**
A* with level sum	No	Yes	86	88	**9**	158.721

Problem 3:

Search algorithm	Best solution?	Optimal	Expansions	Goal Tests	Plan length	Time
A* with h1	No	Yes	18223	18225	**12**	50.298
A* with ignore preconditions	**Yes**	**Yes**	**5040**	**5042**	**12**	**18.665**
A* with level sum	No	Yes	315	317	**12**	898.438

Optimal Actions

Here is the optimal sequence of actions for each problem, as found by *greedy best first graph search* (problem 1) and *A* with ignore preconditions* (problems 2 and 3):

Problem 1:

Load(C1, P1, SFO)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)

Problem 2:

Load(C3, P3, ATL)
Fly(P3, ATL, SFO)
Unload(C3, P3, SFO)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)
Load(C1, P1, SFO)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)

Problem 3:
Load(C2, P2, JFK)
Fly(P2, JFK, ORD)
Load(C4, P2, ORD)
Fly(P2, ORD, SFO)
Unload(C4, P2, SFO)
Load(C1, P1, SFO)
Fly(P1, SFO, ATL)
Load(C3, P1, ATL)
Fly(P1, ATL, JFK)
Unload(C3, P1, JFK)
Unload(C2, P2, SFO)
Unload(C1, P1, JFK)

Results summary and analysis

The best heuristics for each problem were: *greedy best first graph search* for problem 1; *A* with ignore preconditions heuristic* for problems 2 and 3. *Breadth first*, *breadth first tree search*, *uniform cost search*, *recursive best first search*, and all three *A** searches found the optimal solution, but *greedy best first graph search* found it in the shortest time.

For problems 2 and 3, *breadth first search*, *uniform cost search*, *A**, and *A* with ignore preconditions* each found the shortest solutions. *A* with ignore preconditions* was the fastest in both problems, typically taking around half the time of the next-best search algorithm (*breadth first search*).

Interestingly enough, *depth first* and *greedy best first* both completed problems 2 and 3 in similar times to *A* with ignore preconditions*, but neither of them found the shortest solution because they are not optimal[1]. Although *greedy best first* performed more expansions than *depth first*, it always found a significantly shorter solution in a similar time.

These results indicate that for simple problems, heuristics may make little difference, and so uninformed search can be faster. For more complex problems, however, informed search appears to be always better: typically finding an optimal solution in far less time than optimal uninformed search. However, if an optimal solution is not needed and the search must be uninformed, then *greedy best first graph search* is a good option.

References

Source 1: figure 3.21 from chapter 3.4.7 of AI: A Modern Approach

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.