Heuristic Analysis

Experiment and document metrics for non-heuristic planning solution searches

Metrics

Problem #	Algorithm	Expansions	Goal Tests	New Nodes	Elapsed time	Plan length	Optimality
P1	breadth_first_search	43	56	180	0.03899741001	6	yes
P1	breadth_first_tree_search	1458	1459	5960	1.146969888	6	yes
P1	depth_first_graph_search	12	13	48	0.009340999997	12	no
P1	depth_limited_search	101	271	414	0.11319335	50	no
P1	uniform_cost_search	55	57	224	0.04488533101	6	yes
P1	recursive_best_first_search with h_1	4229	4230	17029	3.379986227	6	yes
P1	greedy_best_first_graph_search with h_1	7	9	28	0.006120803009	6	yes
P1	astar_search with h_1	55	57	224	0.09059759599	6	yes
P1	astar_search h_ignore_preconditions	41	43	179	0.07623481099	6	yes
P1	astar_search h_pg_levelsum	11	13	50	1.672863064	6	yes
P2	breadth_first_search	3343	4609	30509	15.95759631	9	yes
P2	breadth_first_tree_search	More than 30 min					
P2	depth_first_graph_search	1669	1670	14863	19.09312299	1444	no
P2	depth_limited_search	222719	2053741	2054119	1800.648565	50	no
P2	uniform_cost_search	4853	4855	44041	28.54468096	9	yes
P2	recursive_best_first_search	More than 30 min					

	with h_1						
P2	greedy_best_first_graph_search with h_1	895	897	8009	3.00111182	21	no
P2	astar_search with h_1	4853	4855	4401	30.13854012	9	yes
P2	astar_search h_ignore_preconditions	1450	1452	13303	10.61081526	9	yes
P2	astar_search h_pg_levelsum	86	88	841	401.1721352	9	yes
P3	breadth_first_search	14663	18098	129631	155.7868547	12	yes
P3	breadth_first_tree_search	More than 30 min					
P3	depth_first_graph_search	592	593	4927	6.906903099	571	no
P3	depth_limited_search	More than 30 min					
P3	uniform_cost_search	18223	18225	159618	129.5421904	12	yes
P3	recursive_best_first_search with h_1	More than 30 min					
P3	greedy_best_first_graph_search with h_1	5578	5580	49150	44.00343115	22	no
P3	astar_search with h_1	18223	18225	159618	132.9183056	12	yes
P3	astar_search h_ignore_preconditions	5040	5042	44944	40.61527371	12	yes
P3	astar_search h_pg_levelsum	325	327	3002	1666.267878	12	yes

Results - P1 Breadth First Search

```
Solving Air Cargo Problem 1 using breadth_first_search...

Expansions   Goal Tests   New Nodes
    43          56          180

Plan length: 6  Time elapsed in seconds: 0.03454302300087875
Load(C2, P2, JFK)
Load(C1, P1, SFO)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)
Fly(P1, SFO, JFK)
```

```
Unload(C1, P1, JFK)
```

Results - P2 Breadth First Search

```
Solving Air Cargo Problem 2 using breadth_first_search...

Expansions   Goal Tests   New Nodes
   3343        4609       30509

Plan length: 9  Time elapsed in seconds: 14.996932713002025
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)
```

Results - P3 Breadth First Search

```
Solving Air Cargo Problem 3 using breadth_first_search...

Expansions   Goal Tests   New Nodes
  14663       18098       129631

Plan length: 12  Time elapsed in seconds: 111.63809110399961
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)
```

Results - P1 Depth First Search

```
Solving Air Cargo Problem 1 using depth_first_graph_search...

Expansions   Goal Tests   New Nodes
```

```
   12          13          48

Plan length: 12  Time elapsed in seconds: 0.009610704997612629
Fly(P1, SFO, JFK)
Fly(P2, JFK, SFO)
Load(C1, P2, SFO)
Fly(P2, SFO, JFK)
Fly(P1, JFK, SFO)
Unload(C1, P2, JFK)
Fly(P2, JFK, SFO)
Fly(P1, SFO, JFK)
Load(C2, P1, JFK)
Fly(P2, SFO, JFK)
Fly(P1, JFK, SFO)
Unload(C2, P1, SFO)
```

Results - P2 Depth First Search

```
Solving Air Cargo Problem 2 using depth_first_graph_search...

Expansions   Goal Tests   New Nodes
   1669        1670       14863

Plan length: 1444  Time elapsed in seconds: 16.057206666999264
Fly(P3, ATL, SFO)
Fly(P1, SFO, ATL)
Fly(P3, SFO, JFK)
Fly(P1, ATL, JFK)
Fly(P2, JFK, ATL)
Fly(P3, JFK, ATL)
Fly(P2, ATL, SFO)
Fly(P3, ATL, SFO)
Load(C1, P3, SFO)
Fly(P3, SFO, ATL)
…

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

Results - P3 Depth First Search

```
Solving Air Cargo Problem 3 using depth_first_graph_search...

Expansions   Goal Tests   New Nodes
   592         593         4927

Plan length: 571  Time elapsed in seconds: 3.373958999000024
Fly(P1, SFO, ORD)
Fly(P2, JFK, ORD)
Fly(P1, ORD, ATL)
Fly(P2, ORD, ATL)
Fly(P1, ATL, JFK)
Fly(P2, ATL, SFO)
Load(C1, P2, SFO)
Fly(P2, SFO, ORD)
…
Fly(P2, ATL, JFK)
Unload(C3, P2, JFK)
Fly(P2, JFK, ORD)
Fly(P1, JFK, ORD)
Fly(P2, ORD, ATL)
Fly(P1, ORD, ATL)
Fly(P2, ATL, SFO)
Fly(P1, ATL, SFO)
Unload(C4, P1, SFO)
Fly(P2, SFO, ORD)
Fly(P1, SFO, ORD)
Fly(P2, ORD, ATL)
Fly(P1, ORD, ATL)
Fly(P2, ATL, JFK)
Fly(P1, ATL, JFK)
Unload(C1, P1, JFK)
```

Results - P1 Greedy Best first with H1 heuristic

```
Solving Air Cargo Problem 1 using greedy_best_first_graph_search with h_1...

Expansions   Goal Tests   New Nodes
    7           9           28
```

```
Plan length: 6  Time elapsed in seconds: 0.0058852840011240914
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P1, SFO, JFK)
Fly(P2, JFK, SFO)
Unload(C1, P1, JFK)
Unload(C2, P2, SFO)
```

Results - P2 Greedy Best first with H1 heuristic

```
Solving Air Cargo Problem 2 using greedy_best_first_graph_search with h_1...

Expansions   Goal Tests   New Nodes
   895         897         8009

Plan length: 21  Time elapsed in seconds: 2.430014035999193
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Load(C3, P3, ATL)
Fly(P1, SFO, ATL)
Fly(P2, JFK, ATL)
Fly(P3, ATL, JFK)
Fly(P2, ATL, SFO)
Unload(C2, P2, SFO)
Fly(P2, SFO, ATL)
Fly(P3, JFK, SFO)
Load(C2, P3, SFO)
Fly(P3, SFO, JFK)
Fly(P1, ATL, JFK)
Unload(C1, P1, JFK)
Load(C1, P3, JFK)
Fly(P1, JFK, ATL)
Fly(P3, JFK, SFO)
Unload(C3, P3, SFO)
Unload(C2, P3, SFO)
Fly(P3, SFO, JFK)
Unload(C1, P3, JFK)
```

Results - P3 Greedy Best first with H1 heuristic

```
Solving Air Cargo Problem 3 using greedy_best_first_graph_search with h_1...

Expansions   Goal Tests   New Nodes
   5578        5580       49150
```

```
Plan length: 22  Time elapsed in seconds: 18.52128859700315
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P1, SFO, ORD)
Load(C4, P1, ORD)
Fly(P2, JFK, ATL)
Load(C3, P2, ATL)
Fly(P2, ATL, ORD)
Fly(P1, ORD, ATL)
Unload(C4, P1, ATL)
Fly(P1, ATL, ORD)
Fly(P2, ORD, ATL)
Load(C4, P2, ATL)
Fly(P2, ATL, ORD)
Unload(C3, P2, ORD)
Load(C3, P1, ORD)
Fly(P1, ORD, JFK)
Unload(C3, P1, JFK)
Unload(C1, P1, JFK)
Fly(P1, JFK, ORD)
Fly(P2, ORD, SFO)
Unload(C4, P2, SFO)
Unload(C2, P2, SFO)
```

Optimal Plans

Problem #	Algorithm	Expansions	Goal Tests	New Nodes	Elapsed time	Plan length	Optimality
P1	greedy_best_first_graph_search with h_1	7	9	28	0.006120803009	6	yes
P2	astar_search h_ignore_preconditions	1450	1452	13303	10.61081526	9	yes
P3	astar_search h_ignore_preconditions	5040	5042	44944	40.61527371	12	yes

Comparisons and contrasts

Problem	Algorithm	Notes
P1	Breadth first	3rd fastest for P1, 2nd for an optimal one and 6 times more expansions than greedy
P1	Depth first	2nd non fastest, non optimal, plan twice as long
P1	Greedy best first	Top performer, smaller number of expansions
P2	Breadth first	2nd optimal performer,
P2	Depth first	Worst plan length, as in 160 times worst than the optimal plan length
P2	Greedy best first	Top performer, non optimal, 3 times worst plan length, not so bad expansion, half than the top optimal performer
P3	Breadth first	Didn't finish
P3	Depth first	Didn't finish
P3	Greedy best first	Non optimal

Problem	Algorithm	Notes
P1	A* ignore preconditions	Optimal, 6 times as much expansions as the top performer, mid performer
P1	A* level-sum	Optimal, second place in expansions, almost the worst performer
P2	A* ignore preconditions	Optimal, top optimal performer, second place in expansions just below A* level-sum
P2	A* level-sum	Optimal, the least number of expansions of all
P3	A* ignore preconditions	Optimal, top optimal performer, 15 times more expansions than level-sum
P3	A* level-sum	Optimal, worst optimal performer but the one with

		the least number of expansions

Conclusions

What was the best heuristic used in these problems?

For P1 it was greedy then breadth and last uniform search. But a* algs were right next just .02 ms below the uniform cost search.

For P2 it was first a* ignoring preconditions then breadth and last uniform search.

For P3 it was again first a* ignoring preconditions and then uniform cost and last a* with h_1.

So uninformed search strategies were present and all problems, specifically the uniform cost search.

Was it better than non-heuristic search planning methods for all problems?

No, a* ignoring preconditions was the most common top performer.

Definitions

Optimal solution

Means the planner must provide the shortest possible plan

Satisfying solution

Good enough, means no optimal solutions are accepted

Hand-coded solution

Means domain specific heuristics are allowed

Automated solution

Means domain specific heuristics are not allowed

Measuring problem-solving performance

Completeness

Is the algorithm guaranteed to find a solution when there is one?

Optimality

Does the strategy find the optimal solution?

Time Complexity

How long does it take to find a solution?

Space Complexity

How much memory is needed to perform the search?

Complexity

In theoretical computer science the typical measure is the size of the state space graph, |V| + |E|, this is appropriate when the graph is an explicit data structure that is input to the search program.

In AI the graph is often represented by the initial state, actions, and transition model and is frequently infinite.

Complexity in AI

It is expressed in terms of three quantities:

- Branching factor, maximum number of successors of any node
- Depth of shallowest goal node, number of steps along the path from the root
- Maximum length of an path in the state space

Algorithms qualities

breadth_first_search	Checks all neighbours first. Complete and optimal as long as it returns within a reasonable amount of time.
breadth_first_tree_search	Checks all neighbours first. Complete and optimal as long as it returns within a reasonable amount of time.
depth_first_graph_search	Picks a neighbour and goes deep before checking the other neighbours. Depth-ish algorithms are not complete or optimal.
depth_limited_search	Same as depth first, but the deep is limited, and looks at the neighbours as soon as the depth is reached
uniform_cost_search	Puts the neighbours in a priority queue and first explores neighbours with a specific cost. Complete and optimal as long as it returns within a reasonable amount of time.
recursive_best_first_search with h_1	Like a recursive depth first, but remembers the f-cost of an alternative path, if deepening reaches a higher f-cost, it then unwinds and goes to the alternative path and update the f-cost of alternative paths on each node.
greedy_best_first_graph_search with h_1	As best first, tries to expand the node that is closest to the goal.
astar_search with h_1	Another best first, the heuristic is a combination of cost to reach node plus cost to reach the goal from that node.
astar_search h_ignore_preconditions	Same as before, but ignores the preconditions of a state
astar_search h_pg_levelsum	Same as before, the heuristic adds up the costs of each individual goal in a level