Implement a Planning Search

Metrics For *Non-heuristic* Searches

Searches used:
- Breadth-First Search
- Depth First Graph Search
- Depth Limited Search
- Uniform Cost Search

Node Expansions

Problem	Breadth First	Depth-first	Depth-limited	Uniform Cost
1	43	21	101	55
2	3343	624	**Stopped**	4852
3	14663	408	**Stopped**	18223

Goal Tests

Problem	Breadth First	Depth-first	Depth-limited	Uniform Cost
1	56	22	271	57
2	4609	625	**Stopped**	4854
3	18098	409	**Stopped**	18225

New Nodes

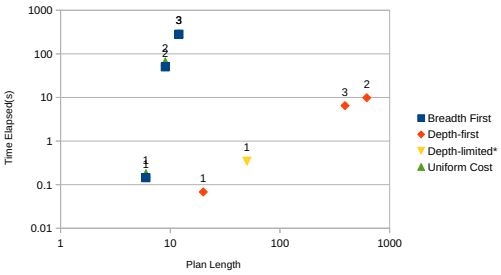
Problem	Breadth First	Depth-first	Depth-limited	Uniform Cost
1	180	84	414	224
2	30509	5602	**Stopped**	44030
3	129631	3364	**Stopped**	159618

Plan Length

Problem	Breadth First	Depth-first	Depth-limited	Uniform Cost
1	6	20	50	6
2	9	619	**Stopped**	9
3	12	392	**Stopped**	12

Time Elapsed (seconds)

Problem	Breadth First	Depth-first	Depth-limited	Uniform Cost
1	0.144	0.068	0.346	0.1789
2	50.208	9.871	**Stopped**	65.765
3	283.314	6.504	**Stopped**	288.776



Metrics for A* Searches With *Heuristics*

A* Searches conducted with the following heuristics:
- Fixed Cost of 1
- Ignore Preconditions
- Level Sum

Node Expansions

Problem	Fixed Cost(1)	Ignore Precond	Level Sum
1	55	41	32
2	4852	1450	168
3	18223	5040	935

Goal Tests

Problem	Fixed Cost(1)	Ignore Precond	Level Sum
1	57	43	34
2	4854	1452	170
3	18225	5042	937

New Nodes

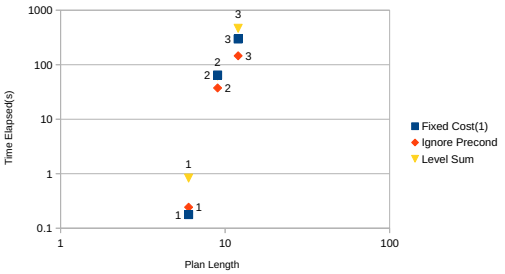
Problem	Fixed Cost(1)	Ignore Precond	Level Sum
1	224	170	138
2	44030	13303	1618
3	159618	44944	8670

Plan Length

Problem	Fixed Cost(1)	Ignore Precond	Level Sum
1	6	6	6
2	9	9	9
3	12	12	12

Time Elapsed (seconds)

Problem	Fixed Cost(1)	Ignore Precond	Level Sum
1	0.178	0.243	0.833
2	64.739	37.484	61.997
3	296.805	145.629	465.860



Analysis

Optimal Plans

Problem 1

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

Expansions   Goal Tests   New Nodes
    43          56         180

Plan length: 6  Time elapsed in seconds: 0.14189186997828074
Load(C1, P1, SFO)
Load(C2, P2, JFK)
Fly(P2, JFK, SFO)
Unload(C2, P2, SFO)
Fly(P1, SFO, JFK)
Unload(C1, P1, JFK)
```

Problem 2

```
Solving Air Cargo Problem 2 using astar_search with h_ignore_preconditions...

Expansions   Goal Tests   New Nodes
   1450        1452       13303

Plan length: 9  Time elapsed in seconds: 36.45908311600215
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

```
Solving Air Cargo Problem 3 using astar_search with h_ignore_preconditions...

Expansions   Goal Tests   New Nodes
   5040        5042       44944

Plan length: 12  Time elapsed in seconds: 143.9886807659932
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)
```

Uninformed/Non-heuristic Searches

Breadth-first search and uniform cost search appear to have similar performance metrics while depth-limited search is very slow and inefficient. For problems 2 and 3, using depth-limited search, I stopped the search after 10 minutes. Depth-limited search may be taking long to complete because the state-space is very large and the solution may be at deeper levels(nonoptimal).

Between breadth-first search and uniform cost search they both achieve the same plan length but the former achieves the goal more efficiently: it's faster while using fewer node expansions, new nodes, and goal tests. When all step costs are equal uniform cost and breadth-first search are essential equal. However, breadth-first stops as soon as it generates a goal while uniform cost search examines all nodes at goals depth, thus slightly less efficient.

Informed/Heuristic Searches

Planning problems are search problems with large state-spaces and without a good heuristic the searches are inneficient. The heuristic will essentially guide the search. With an admissible heuristic for the distance from a state s to the goal we can use A* search to find optimal solutions.

We relaxed the problem to obtain three heuristics:
- every action has cost of 1,
- drops all preconditions from actions, and
- sums the level costs of the goals.

Overall, all heuristics found the optimal plan length for each problem but they achieve it with varying efficiencies. To be expected, the fixed cost heuristic, the most relaxed heuristic with no domain-knowledge, expands the most nodes, creates the most new nodes, and processes the most goal tests. The fixed cost heuristic for A* is uninformed uniform cost search–not a real heuristic.

Alternatively, the ignore preconditions heuristic achieves the goal the fastest but uses more nodes, goal test, and node expansions than *level sum heuristic*.

As stated by Hoffman, the act of dropping some preconditions is guaranteed to lead to admissible heuristics but computing 'relaxed' heuristics can be as hard as solving the original problem. Further, eliminating all preconditions makes the problem feasible but we lose significant information.

Further, McDermott's UNPOP suffered from the same inefficiencies, as stated here:

> "[UNPOP] tends to do poorly on problems where a goal literal g that is true in the current situation is sure to be deleted by an action that must be taken, but not right away…[An example is the "rocket" domain.]The rocket can only be used once, a fact expressed by having the action of flying a rocket delete the precondition has-fuel(rocket), which is not added by any action. Unpop considers moving cargo to two different destinations by flying the same rocket, and once again will try all possible permutations of cargo and rockets

before finally flying the rocket and realizing that the plan prefix just canâ€™t be extended to a solution.” (Pg. 148)

We lose information by ignoring preconditions since we only care about the current state and not future interactions.

On the other hand, the *level sum heuristic* uses significantly fewer node expansions, goal tests, and new nodes than either of the other two searches, especially in problem 2 and 3.

The level sum heuristic assumes that the subgoals are independent; in other words, the cost of achieve all the goals is the sum of the cost of each goal. However, the heuristic leads to inadmissibility: it can overestimate the cost due to the redundant actions. But, Hoffman suggest that a planning graph eliminates redundant actions that could cause overestimates.

Sources

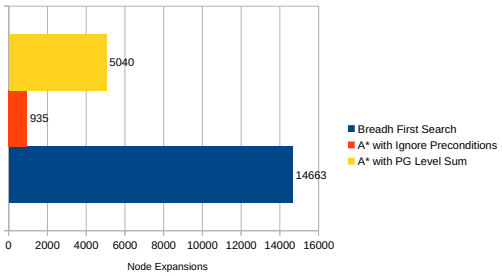
J. Hoffman, B. Nebel, The FF Planning System: Fast Plan Generation Through Heuristic Search, in: Journal of Artificial Intelligence Research 14 (2001) 253-302

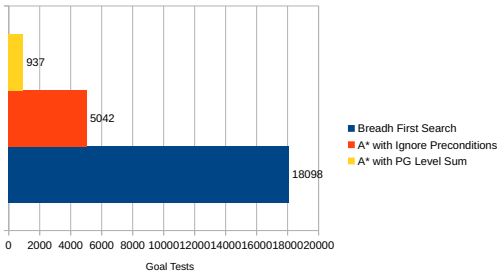
D. McDermott, A heuristic estimator for means-ends analysis in planning, in: Proc. 3rd International Conference on AI Planning Systems, AAAI Press, Menlo Park, CA, 1996, pp. 142â€“149

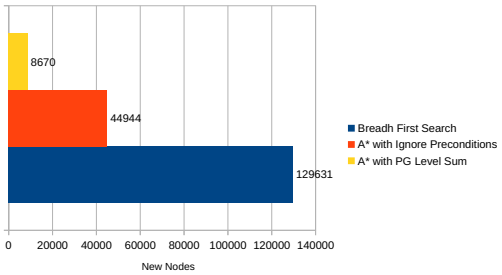
Best heuristic

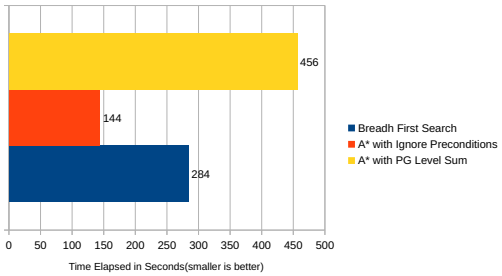
A* search using level sum appears to be by far the most efficient with the number of node expansions, goal tests and new nodes created. However it’s not the fastest especially with more complex problems such as problem 3. In fact, breadth first search is faster(~about 1.5 times faster) than level sum. However, breadth first search uses significantly more resources to achieve an optimal plan. Given this, we can look for a middle ground between speed and efficiency which leads us to *A* search with preconditions ignored*(heuristic). It’s quicker than A* with level sum but not as efficient. On the other hand, it’s more efficient and faster than breadth-first search. Therefore, the “best” heuristic for this problem seems to be A* with ignore preconditions heuristic.

Problem 3 comparisons










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

Expansions   Goal Tests   New Nodes
  14663       18098       129631

Plan length: 12  Time elapsed in seconds: 284.1923261680058
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)

Solving Air Cargo Problem 3 using astar_search with h_pg_levelsum...

Expansions   Goal Tests   New Nodes
   935         937         8670

Plan length: 12  Time elapsed in seconds: 456.05580870600534
Load(C1, P1, SFO)
Fly(P1, SFO, ATL)
Load(C3, P1, ATL)
Fly(P1, ATL, JFK)
Unload(C3, P1, JFK)
Load(C2, P2, JFK)
Fly(P2, JFK, ORD)
Load(C4, P2, ORD)
Fly(P2, ORD, SFO)
Unload(C2, P2, SFO)
Unload(C4, P2, SFO)
Unload(C1, P1, JFK)

Solving Air Cargo Problem 3 using astar_search with h_ignore_preconditions...

Expansions   Goal Tests   New Nodes
   5040        5042       44944

Plan length: 12  Time elapsed in seconds: 143.9886807659932
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)
```