# Heuristic Analysis

## Optimal Plans

I worked out following planning solutions

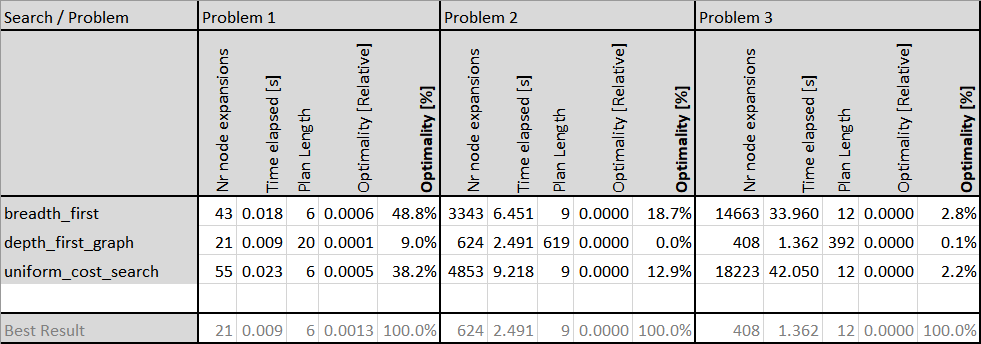
|  |  |
| --- | --- |
| Problem # | Plan |
| 1 | Load(C1, P1, SFO)  Fly(P1, SFO, JFK)  Load(C2, P2, JFK)  Fly(P2, JFK, SFO)  Unload(C1, P1, JFK)  Unload(C2, P2, SFO) |
| 2 | Load(C1, P1, SFO)  Fly(P1, SFO, JFK)  Load(C2, P2, JFK)  Fly(P2, JFK, SFO)  Load(C3, P3, ATL)  Fly(P3, ATL, SFO)  Unload(C3, P3, SFO)  Unload(C2, P2, SFO)  Unload(C1, P1, JFK) |
| 3 | Load(C2, P2, JFK)  Fly(P2, JFK, ORD)  Load(C4, P2, ORD)  Fly(P2, ORD, SFO)  Load(C1, P1, SFO)  Fly(P1, SFO, ATL)  Load(C3, P1, ATL)  Fly(P1, ATL, JFK)  Unload(C4, P2, SFO)  Unload(C3, P1, JFK)  Unload(C2, P2, SFO)  Unload(C1, P1, JFK) |

## Comparison non-heuristic search

Following table compares the performance of Breadth-First-, Depth-First-Graph- and Uniform-Cost-Search. The optimality is calculated by comparing each individual search with the hypothetical best search. The “best search” combines the best result of each individual search (minimal plan length, minimal time lapsed and minimal number node expansions of all searches).

The Optimality is defined by

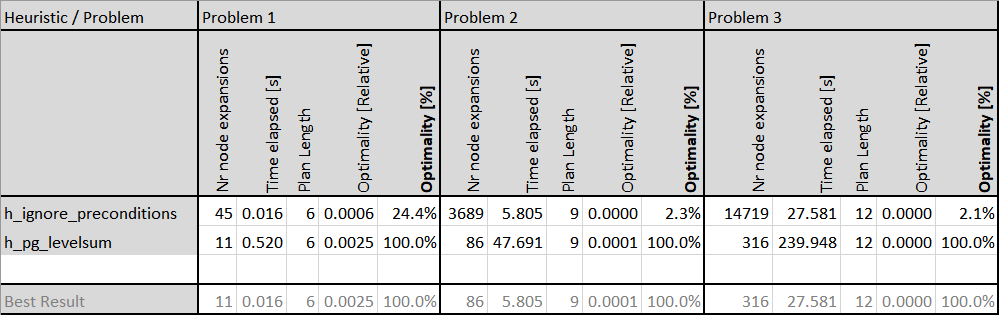
whereas N is “Nr node expansion” and P is “Plan Length”. P is to the power of 2 in order to give a small plan length a much higher weight. Time elapsed is just informative as it is the Wall time and not the CPU time.



Breadth-First was the most optimal search for all three problems.

## Comparison heuristic search

Following table compares the performance of heuristics Ignore-Preconditions and Levelsum. The optimality is defined the same as in “Comparison non-heuristic search”.



Because the Optimality is based on the number of node expansions the Levelsum-Heuristics performs best. This seems questionable since Time elapsed is for Levelsum approx. 10 times higher as it is for Ignore-Preconditions. This difference is massive and depending on the target application/system Levelsum might be too slow and Ignore-Preconditions could be a better choice. It’s therefore unclear which heuristic is better. Further measurements would have to analyse and compare CPU and memory consumption in more depth. It’s also possible that code optimisation could make the Levelsum heuristic more CPU efficient.

# Comparison non-heuristic with heuristic search