Post-hoc Optimization for the Sliding Tile Puzzle

Damian Knuchel <damian.knuchel@unibas.ch>

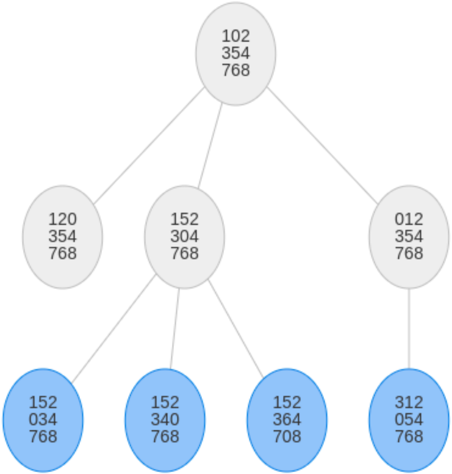
Department of Mathematics and Computer Science, University of Basel

October 14, 2021

Outline

1. Background
2. Sliding Tile Puzzle
3. Heuristics
4. Post-hoc Optimization Heuristic
5. Experimental Evaluation

Classical Planning, State Spaces and Heuristics



Sliding Tile Puzzle

9	5	12	3
0	15	6	11
8	13	1	10
2	4	7	14

→

0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15

Move

9	5	12	3
0	15	6	11
8	13	1	10
2	4	7	14

Move

9	5	12	3
0	15	6	11
8	13	1	10
2	4	7	14

Move

0	5	12	3
9	15	6	11
8	13	1	10
2	4	7	14

Heuristics

1	0	2
3	5	4
7	6	8

Manhattan Distance

1	0	2
3	5	4
7	6	8

Manhattan Distance

1. Manhattan Distance: 1

1	0	2
3	5	4
7	6	8

Manhattan Distance

1. Manhattan Distance: 1

1	0	2
3	5	4
7	6	8

Manhattan Distance

1. Manhattan Distance: 1

1	0	2
3	5	4
7	6	8

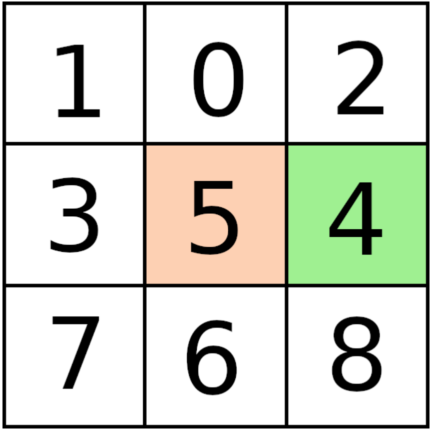
Manhattan Distance

1. Manhattan Distance: 2

1	0	2
3	5	4
7	6	8

Manhattan Distance

1. Manhattan Distance: 3



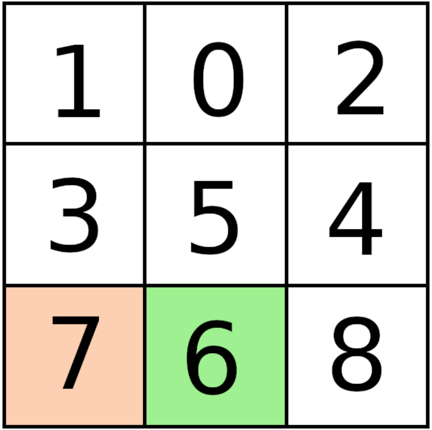
Manhattan Distance

1. Manhattan Distance: 4

1	0	2
3	5	4
7	6	8

Manhattan Distance

1. Manhattan Distance: 5



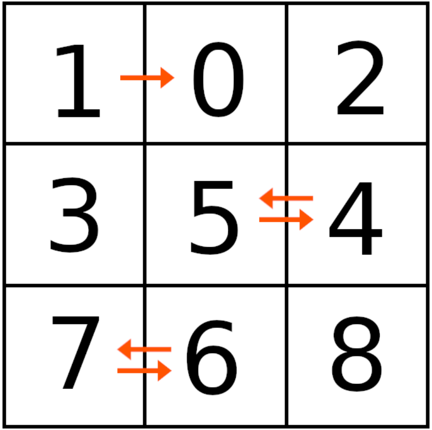
Manhattan Distance

1. Manhattan Distance: 5

1	0	2
3	5	4
7	6	8

Manhattan Distance

1. Manhattan Distance: 5



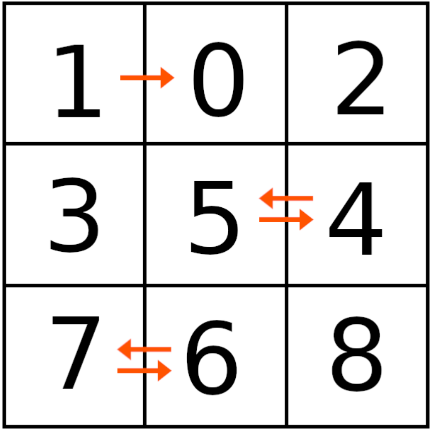
Linear Conflict

1. Manhattan Distance: 5
2. Linear Conflicts: $5 + x * 2$

1	0	2
3	5	4
7	6	8

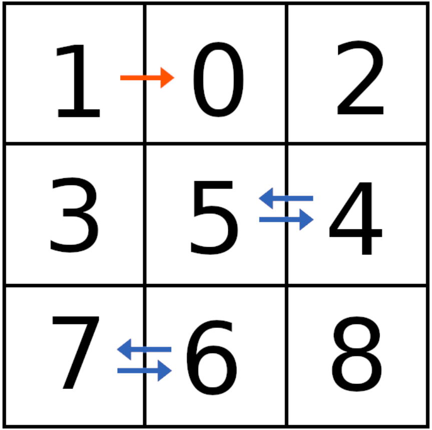
Linear Conflict

1. Manhattan Distance: 5
2. Linear Conflicts: $5 + x * 2$



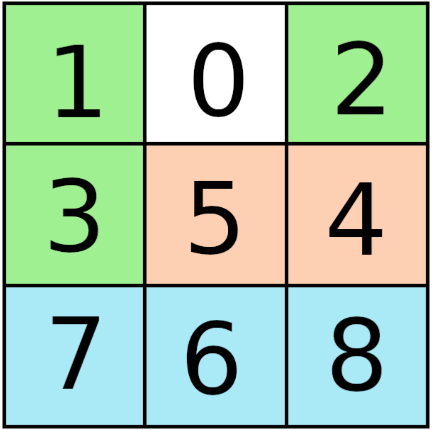
Linear Conflict

1. Manhattan Distance: 5
2. Linear Conflicts: $5 + 2 * 2 = 9$



Statically-Partitioned Additive Pattern Database Heuristic

1. Manhattan Distance: 5
2. Linear Conflicts: 9
3. Statically:

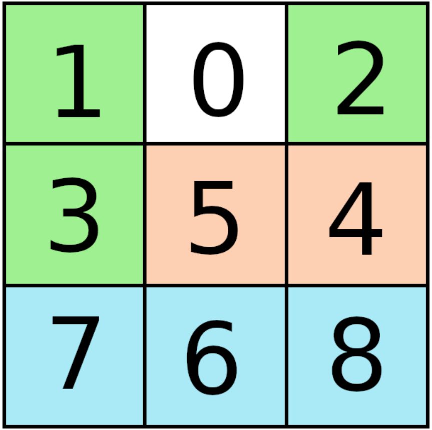


Statically-Partitioned Additive Pattern Database Heuristic

`https://sliding-puzzle-solver.herokuapp.com/`

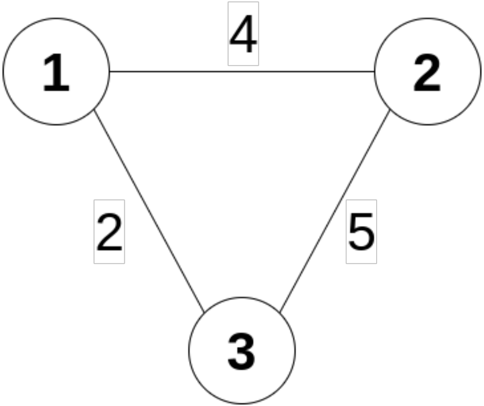
Statically-Partitioned Additive Pattern Database Heuristic

1. Manhattan Distance: 5
2. Linear Conflicts: 9
3. Statically: $1 + 4 + 6 = 11$

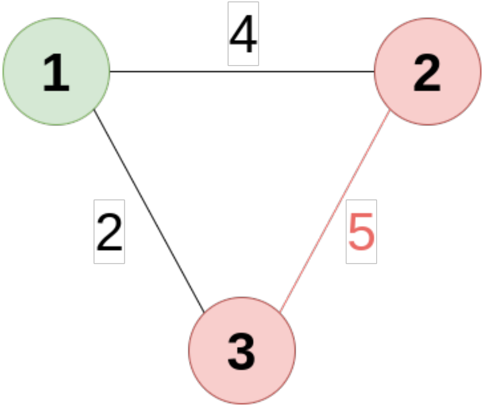


Dynamically-Partitioned Additive Pattern Database Heuristic

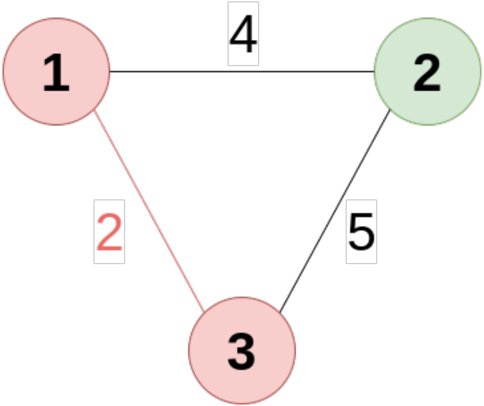
Dynamically-Partitioned Additive Pattern Database Heuristic



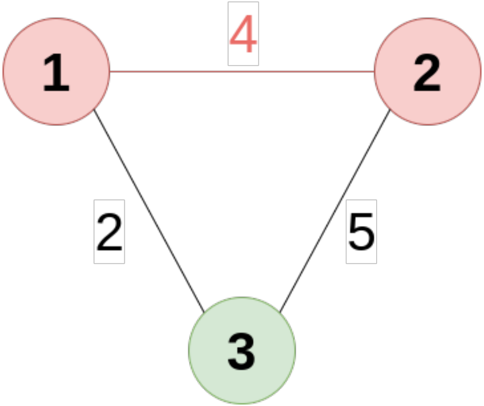
Dynamically-Partitioned Additive Pattern Database Heuristic



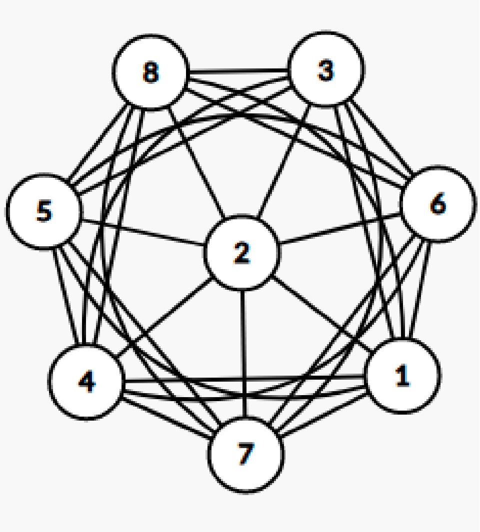
Dynamically-Partitioned Additive Pattern Database Heuristic



Dynamically-Partitioned Additive Pattern Database Heuristic

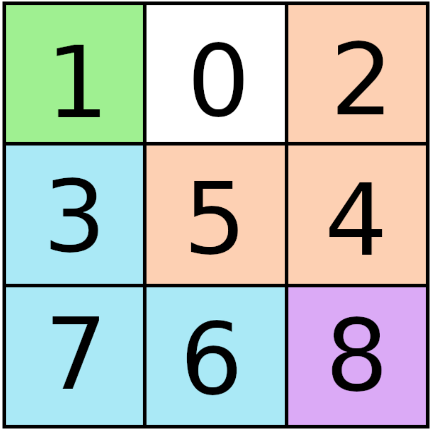


Dynamically-Partitioned Additive Pattern Database Heuristic



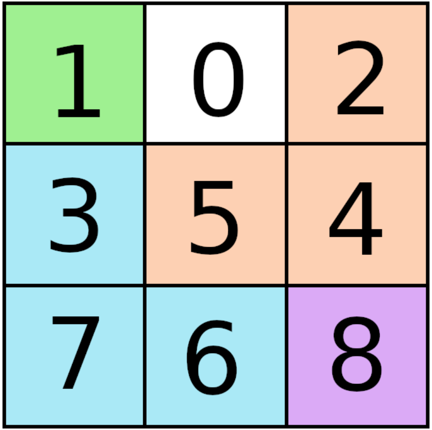
Dynamically-Partitioned Additive Pattern Database Heuristic

1. Manhattan Distance: 5
2. Linear Conflicts: 9
3. Statically: 11
4. Dynamically:



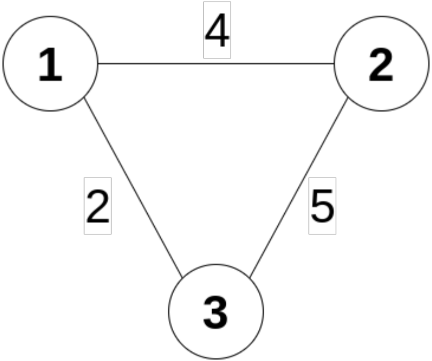
Dynamically-Partitioned Additive Pattern Database Heuristic

1. Manhattan Distance: 5
2. Linear Conflicts: 9
3. Statically: 11
4. Dynamically: $1 + 6 + 6 + 0 = 13$



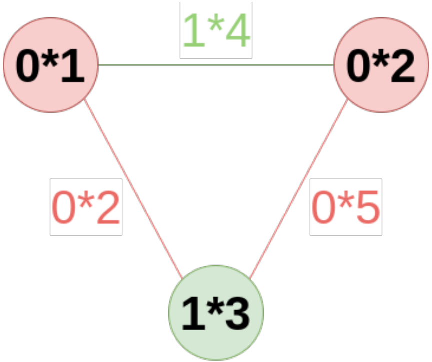
Post-hoc Optimization Heuristic

1. Manhattan Distance: 5
2. Linear Conflicts: 9
3. Statically: 11
4. Dynamically: 13
5. PhO:



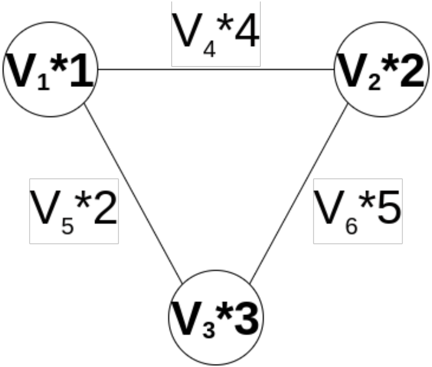
Post-hoc Optimization Heuristic

1. Manhattan Distance: 5
2. Linear Conflicts: 9
3. Statically: 11
4. Dynamically: 13
5. PhO:



Post-hoc Optimization Heuristic

1. Manhattan Distance: 5
2. Linear Conflicts: 9
3. Statically: 11
4. Dynamically: 13
5. PhO:



Post-hoc Optimization Heuristic

$$\textit{Maximize } V_1 * 1 + V_2 * 2 + V_3 * 3 + V_4 * 4 + V_5 * 2 + V_6 * 5 \textit{ subject to}$$

$$V_1 + V_4 + V_5 \leq 1$$

$$V_2 + V_4 + V_6 \leq 1$$

$$V_3 + V_5 + V_6 \leq 1$$

$$V_i \geq 0 \textit{ for all } i \in \{1, 2, 3, 4, 5, 6\}.$$

Post-hoc Optimization Heuristic

Linear program for 8-Puzzle with patterns up to the size of 3 variables:

92 Variables (Heuristics)

8 Constraints (Tiles)

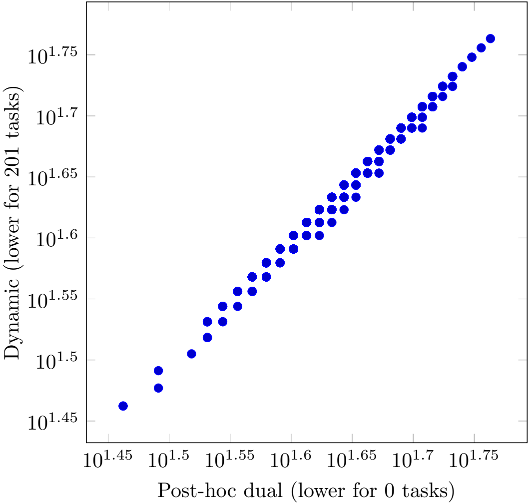
29 Variables/Constraint

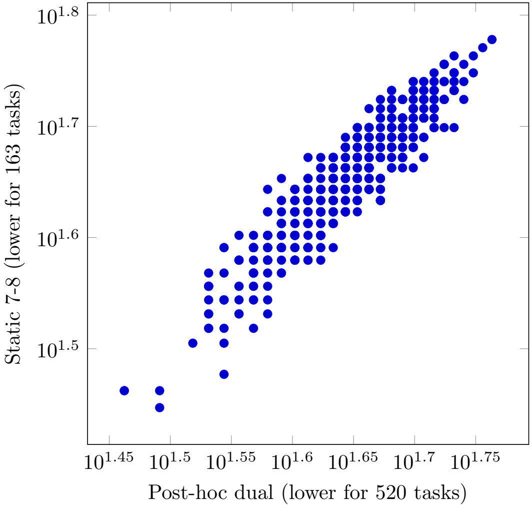
Post-hoc Optimization Heuristic

1. Manhattan Distance: 5
2. Linear Conflicts: 9
3. Statically: 11
4. Dynamically: 13
5. PhO: 15

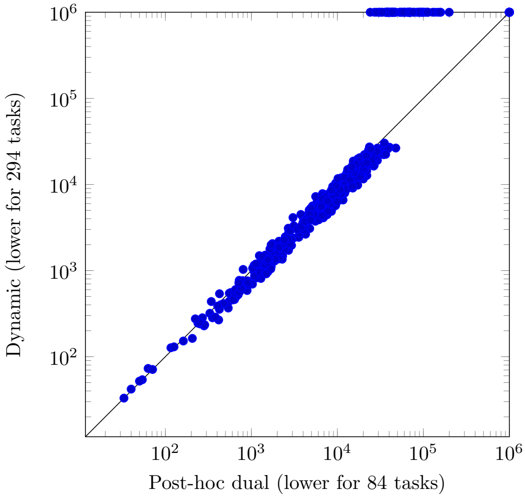
1	0	2
3	5	4
7	6	8

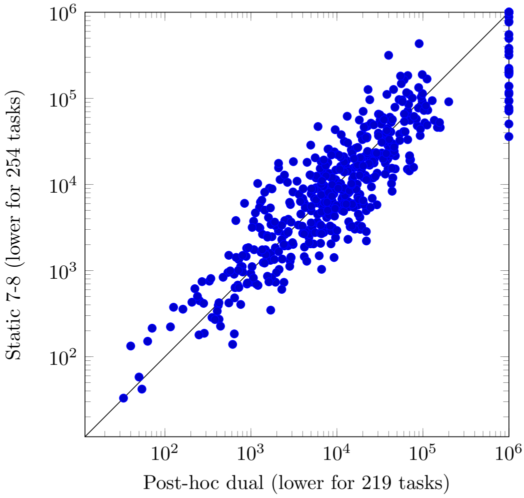
Experimental Evaluation: Initial Heuristic





Experimental Evaluation: Expansion





Questions?

damian.knuchel@unibas.ch