Pattern Databases by Joseph C. Culberson and Jonathan Schaeffer

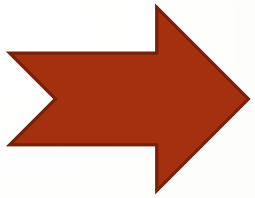
Presented by Patrick von Reth

The 15-Puzzle

- 15-Puzzle is a permutation problem.
  - Problems where a set of operators converts one permutation into another
  - Reach goal permutation with as few operators as possible
- Another prominent example is the Rubik's Cube.

The 15-Puzzle

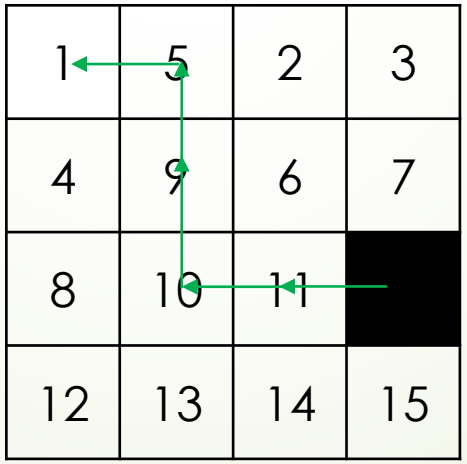
1	5	6	3
12	9	4	2
15	11	10	7
13	14		8



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

Manhattan Distance

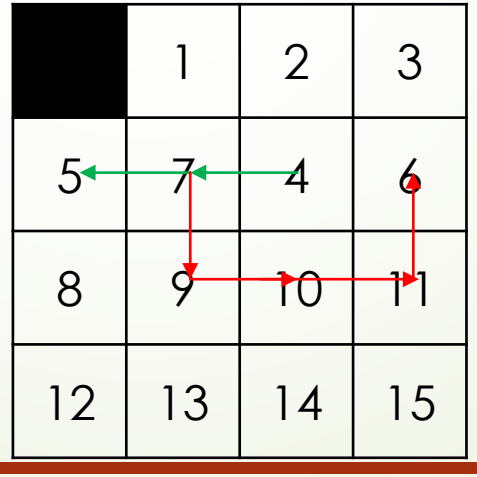
- Distance between two positions in a grid
- Sum of the vertical and horizontal distance
- Example: Distance for the empty tile to its goal position.
- Manhattan heuristic: Sum of distances for all tiles to their goal position.



Linear conflict heuristic

- Extension of the Manhattan heuristic.
- If two tiles are in linear conflict they have to surround each other.

  → Increase the heuristic by two.
- Example: The Manhattan distance for tile 4 and for tile 7 is two.



Pattern databases

- Human-like approach to solve a complex permutation problems:
  - Find a partial solution and thus reduce the complexity of the problem.
- Find an optimal path to a partial solution
- Distance to the partial solution is a lower bound for the complete solution.
- Precompute the distance to the partial solution, for all permutations, and store it in a database.

Fringe and Corner pattern

- The Fringe pattern

			3
			7
			11
12	13	14	15

- The Corner pattern

8	9	10	
12	13	14	15

Upper bounds

- Used to eliminate the last iteration of the IDA* search.
- Upper bound for the 15-Puzzle
  - For the fringe pattern the upper bound is 61 moves.
  - For the remaining 8-Puzzle the upper bound is 31 moves.
  - This results in an upper limit of 92 moves to completely solve the 15-Puzzle.
- Different patterns result in different upper bounds.

Solution Databases

- If obtaining the optimal solution path is to expensive.
  - Precompute optimal solution for sub-goals together with the moves.

Symmetry

- Can be used to simplify the search tree.
- For example reflections can be used to obtain additional cost bounds.
  - The maximum of all lower bounds retrieved is then used.
  - Two diagonal reflections , (0,15) and (12,3).
  - The vertical and the horizontal reflection.

Symmetry - Reflections

- Applying the diagonal reflection (0,15) on a path yields the mirrored path.
- Can be achieved by remapping the operators
  - d->r, r->d, l->u, u->l
- Used to retrieve the cost for the reflected state.

- With some modifications this can also be done for the other reflections.

Symmetry - Cost retrieved using reflections

- Goal state and its horizontal mirror.
  - Distance of the two state is three.
  - Distance of three is used as penalty on the retrieved cost for the mirror state.

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

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

Symmetry – Reflection penalties

- Cost for diagonal reflection (0,15) and its mirrored state are the same.
- Penalty for the horizontal and the vertical reflection is 3.
- Penalty for the diagonal (12,3) reflection is 6.

Combination of patterns

- Both pattern databases are used for the heuristic calculation.
  - Using maximum over both databases would be expensive (memory)
  - Split up in 16 databases each.
  - Lookup defined by empty tile and its mirrors.
  - Only one half of each database is loaded.

F	C	F	C
C	F	C	F
F	C	F	C
C	F	C	F

Linear conflict heuristic compared to pure Manhattan distance heuristic

- Presented results are based on the Manhattan distance.
  - Linear conflict heuristic:
    - Search tree size is reduced noticeable.
  - Pattern database with linear conflict heuristic:
    - Only minor improvement.

Results (1)

- Combination of both pattern databases is better than individual improvements.
- Many enhancements possible
  - Don't use the diagonal (12,3) because of the penalty of 6.
  - Don't lookup every possible reflection.

Results (2)

- Without those improvements:
  - 12 times faster than a naïve implementation only using the Manhattan distance.
  - 1.5 times faster than using the linear conflict heuristic

Conclusion

- Computational workload increased.
  - Computation is done before the actual search.
  - Computation is done once, cost will amortise over time.
- Speedup of 12 and reduction of the search tree size by 1038-fold is impressive.
- Pattern databases are better than the linear conflict heuristic.
- Adaptable for different problems.

Questions ?