Pattern Databases
Culberson and Schaeffer, 1998

Silvan Sievers

Artificial Intelligence
Departement Informatik
Universität Basel

November 15, 2012

UNI BASEL

Introduction

- Motivation:
  - A^* search algorithm of fundamental importance in AI
  - Try to find "good" heuristics
- Contribution: introduction of pattern databases
  - New approach for enhancing single-agent search
  - Evaluated on the 15-Puzzle

Outline

The 15-Puzzle

- Specification:
  - Set of 15 labeled tiles, labeled 1 to 15, and an empty tile, labeled 0
  - 4×4 square grid with positions labeled from 0 to 15
  - State: permutation of $(0 \ldots 15)$ mapping locations to tiles
  - Move: swaps the empty tile with one of its neighbors, denoted by the direction the empty tile moves, i.e. $move \in \{l, r, u, d\}$

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

Goal state τ

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

State p induced by *rrdldruu*

The 15-Puzzle ctd.

- More definitions:
  - Distance between two states: minimum number of moves required to produce one from the other
  - Cost of state s: distance from s to τ
  - Path from state p to state q: move sequence transforming p into q
- Notation:
  - p is obtained from τ by the move sequence *rrdldruu*:

$$\begin{pmatrix} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 \\ 1 & 2 & 0 & 3 & 4 & 9 & 6 & 7 & 8 & 10 & 5 & 11 & 12 & 13 & 14 & 15 \end{pmatrix}$$

  - More compact: (1 2 0 3 4 9 6 7 8 10 5 11 12 13 14 15)

Outline

Pattern Databases

- Definitions:
  - Pattern: partial specification of a permutation (or state), i.e. some locations are occupied by unspecified tiles, called blanks
  - Target pattern: partial specification of τ
  - Pattern database (PDB): set of all patterns that can be obtained by permutations of a target pattern
  - Cost associated with each pattern in a PDB: distance to the target pattern

Pattern Databases ctd.

Lemma

For any state s, for any PDB, the cost of the pattern induced by s obtained from the PDB is a lower bound on the cost of s.

- Using the lower bound yields an admissible heuristic

Example

- Two example target patterns:

[redacted]			3
			7
			11
12	13	14	15

Fringe target pattern

[redacted]			
8	9	10	
12	13	14	15

Corner target pattern

- Consider state s:

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

- Cost of s in the fringe: 6
- Cost of s in the corner: 2

Outline

Symmetries

- Symmetries:
  - Use symmetries to identify "equivalent states"
  - Here: only consider reflections
- The mirror symmetry D:
  - Reflection across the main diagonal $(0, 15)$
  - Maps the puzzle onto an equivalent puzzle with the empty cell in the goal state remaining in the top left corner
  - Path P has an equivalent mirror path P' obtained by replacing moves: $l \leftarrow u, u \leftarrow l, r \leftarrow d, d \leftarrow r$

Computing the Mirror Symmetry

- Computing the mirror state p' of p (without computation of the mirror path!):
  - Apply state p to the mirror D, i.e. compute $p \circ D$
  - "Undo" the mirror operation, i.e. apply D^{-1} on the result
  - Overall computation: $p' = D^{-1} \circ p \circ D$

Lemma

Any upper or lower bound on the cost of a state p or its mirror p' applies to both p and p'.

Using the Mirror Symmetry

- Using the mirror symmetry:
  - PDBs only need to store information about a state and not its mirror (need a canonic definition of state/mirror state)
  - Save nearly 50% of space in PDBs (some states are self-reflections)
  - Fringe PDB symmetric w.r.t. the mirror, thus $cost(p) = cost(p')$
  - Same not true for corner PDB

Other symmetries

- Using other reflections:
  - More involved because empty tile not reflected on itself in the goal state
  - Possibly yields lower bounds for mirrored states different from the original one
  - Combining several reflections: lookup different reflections of state s in the same PDB and use the maximal retrieved value

Summary symmetries

- PDBs:
  - Use the mirror symmetry
  - Only store states and not their mirrors (reduction of one half)
- Other symmetries:
  - Given a state, compute several reflections
  - Use different reflections to retrieve additional information from PDBs

Outline

Experiments

- Setup for IDA*:
  - Manhatten Distance heuristic (MD)
  - MD and fringe PDB (FR)
  - MD and corner PDB (CO)
  - MD and a combination of FR and CO (FC)

Experiment	Total nodes	Tree size (%)	Improvement
MD	36, 302, 808, 031	100.00	1
MD&FR	105, 067, 478	0.29	346
MD&CO	83, 125, 633	0.23	437
MD&FC	34, 987, 894	0.10	1038

Outline

Summary

- Conclusion:
  - New approach for computation of lower bounds: PDBs
  - Domain specific implementation for 15-Puzzle
  - Using symmetry of the problem allows for great performance improvements
- Outlook:
  - Generalize idea to non-permutation problems and general search problems
  - Find an algorithmic way to determine good (target) patterns other than fringe and corner

Using the Mirror Symmetry

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

$\xrightarrow{D}$

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

$\downarrow p$

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

$\xleftarrow{D}$

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

Combination of FR and CO

- Combination of FR and CO:
  - PDBs often produced complementary results
  - Taking the maximum of both required too much memory
  - Split the PDBs instead and look up a value in a PDB depending on the position of the empty tile

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