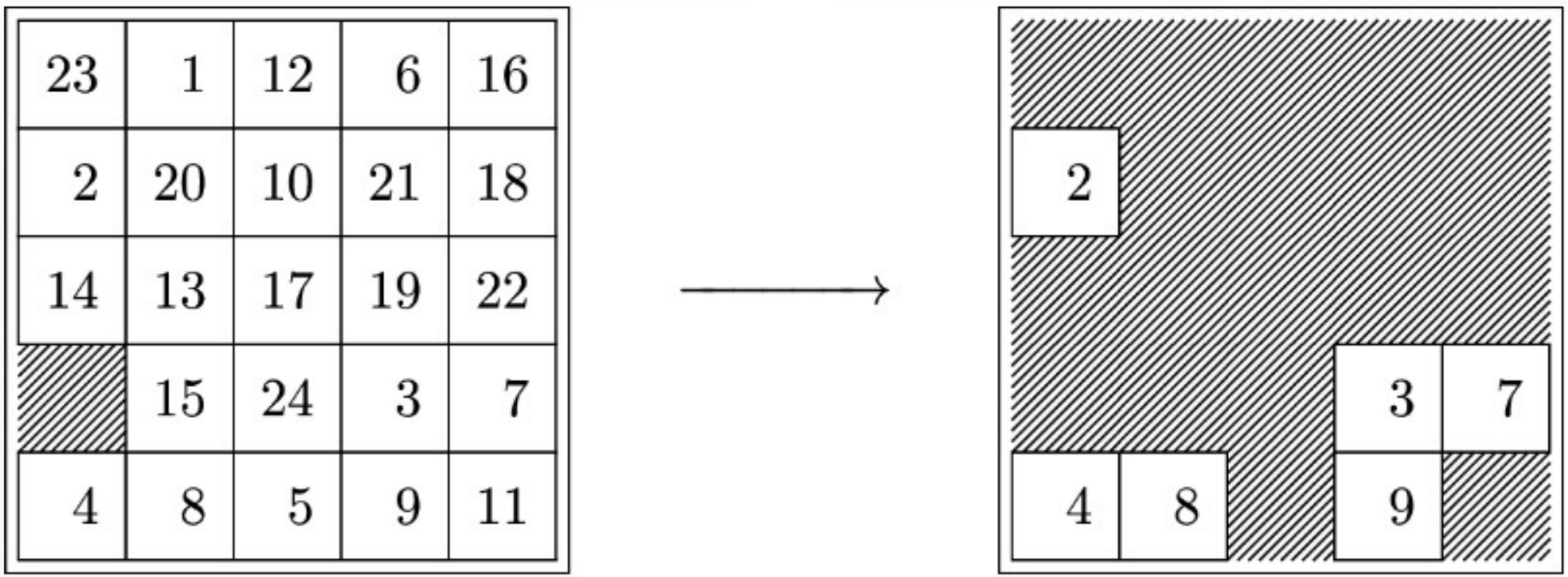
Better PDB Heuristics for Sliding Tile Puzzles

Robert Clausecker
<clausecker@zib.de>

motivation

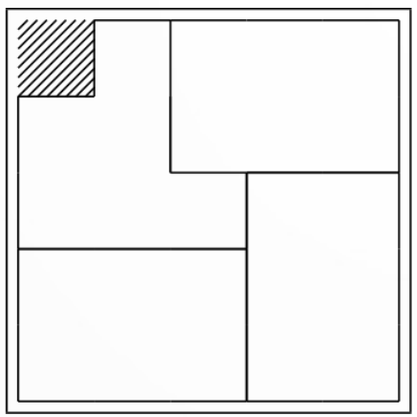
- sliding tile puzzles are a testbed problem for heuristic search
- currently best known heuristics: *additive pattern databases* (APDBs)
- can we do better?

additive pattern databases

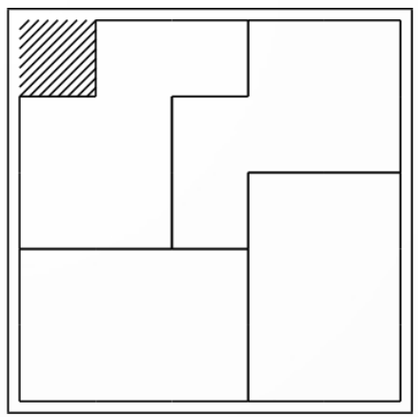


a 24 puzzle and its { 2, 3, 4, 7, 8, 9 } APDB

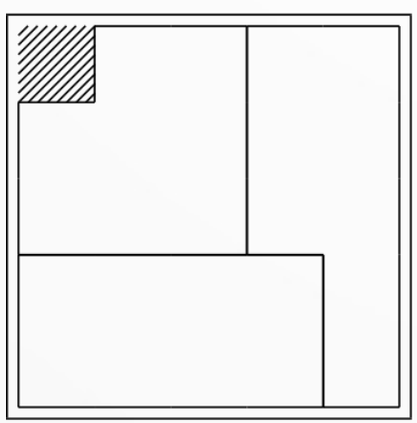
additive pattern databases



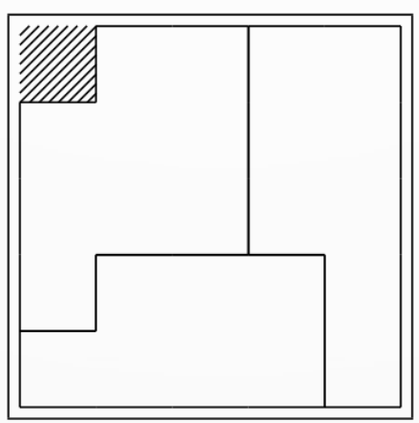
6-6-6-6 orig.



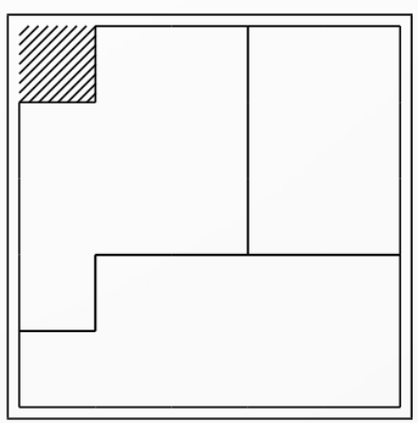
6-6-6-6 new



8-8-8



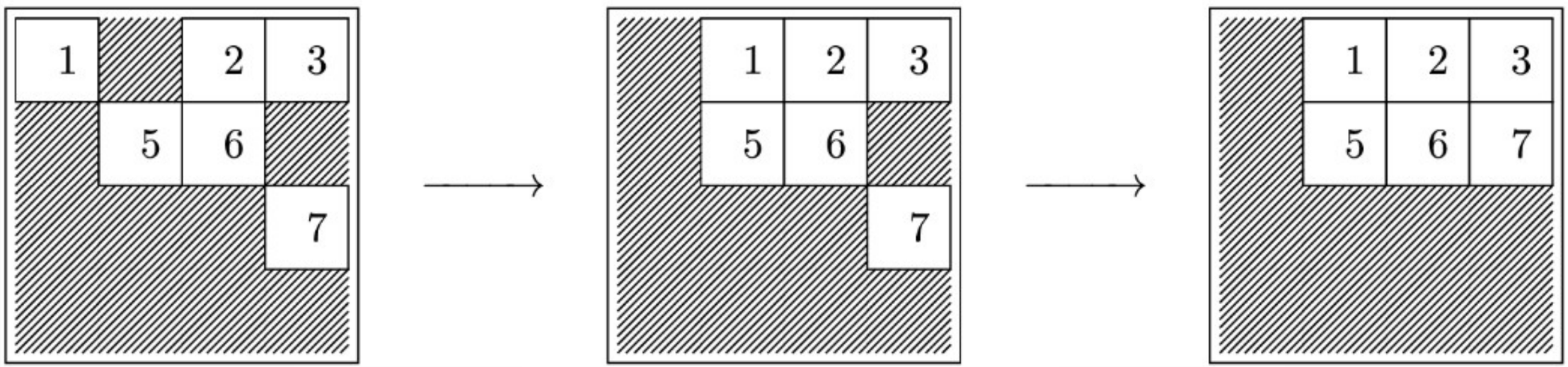
9-8-7



9-9-6

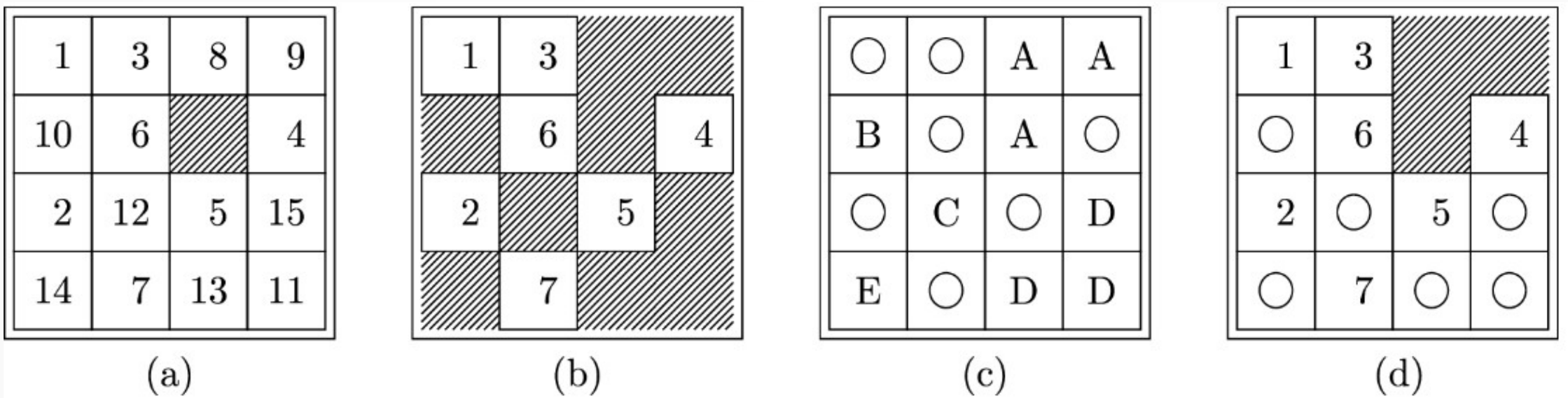
some partitionings of the 24-puzzle

limitations



h = 2 is predicted by the APDB but 2 moves are not sufficient

tracking the blank (zero tile)



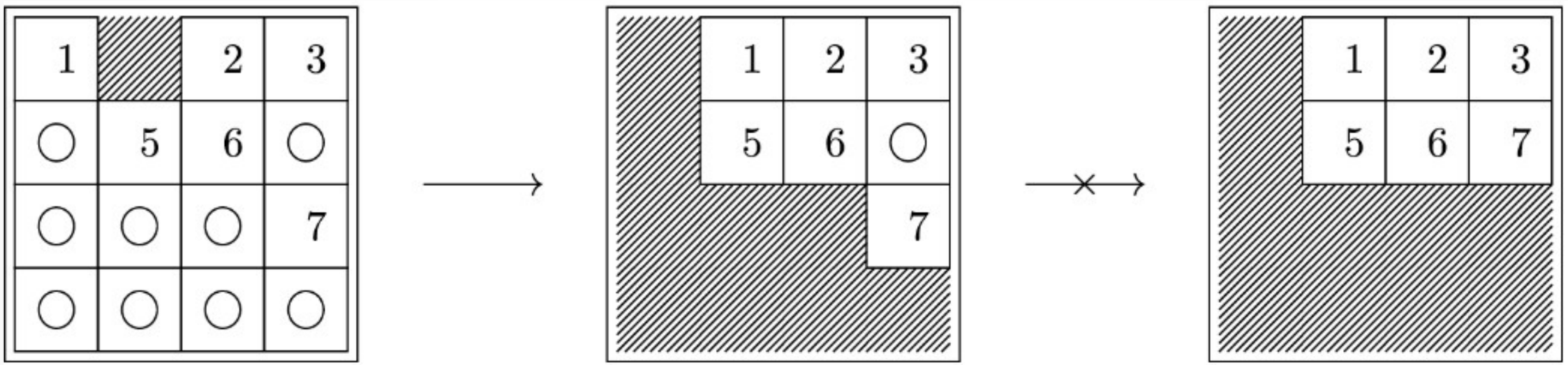
(a) a configuraration of the 15 puzzle

(b) as seen by the { 1, 2, 3, 4, 5, 6, 7 } APDB

(c) its zero-tile regions

(d) as seen by the { 1, 2, 3, 4, 5, 6, 7 } ZPDB

limitations revisited

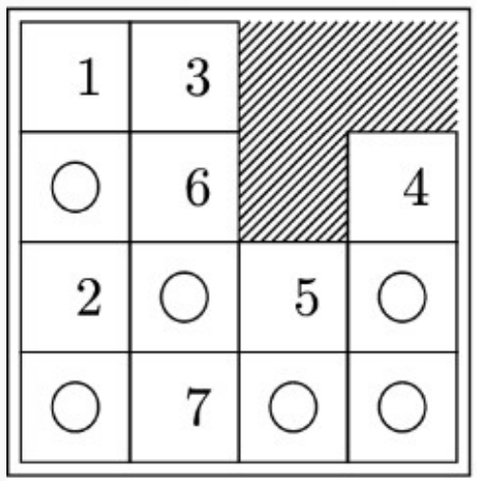


the ZPDB does not predict a 2 move solution

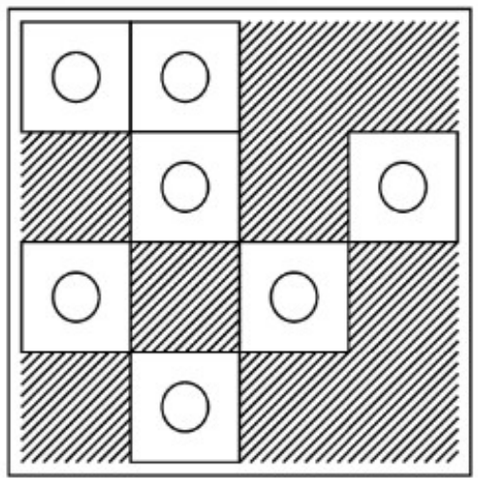
space considerations

k	APDB size	ZPDB size	avg	max
2	600	608	1.01	2
3	13 800	14 472	1.04	2
4	303 600	339 048	1.12	3
5	6 375 600	7 871 280	1.23	4
6	127 512 000	181 008 000	1.42	5
7	2 422 728 000	4 066 655 040	1.68	6
8	43 609 104 000	87 358 400 640	2.00	7
9	741 354 768 000	1 759 513 674 240	2.37	8
10	11 861 676 288 000	32 787 717 580 800	2.76	10
11	177 925 144 320 000	560 680 553 664 000	3.15	11
12	2 490 952 020 480 000	8 749 801 518 796 800	3.51	13

representation



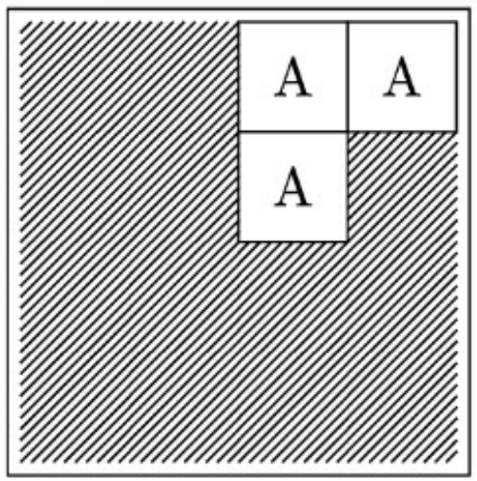
$e = (\mu, \pi, \rho)$



$\mu = 2027$

$$\begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 & 7 \\ 1 & 3 & 6 & 4 & 2 & 5 & 7 \end{pmatrix}$$

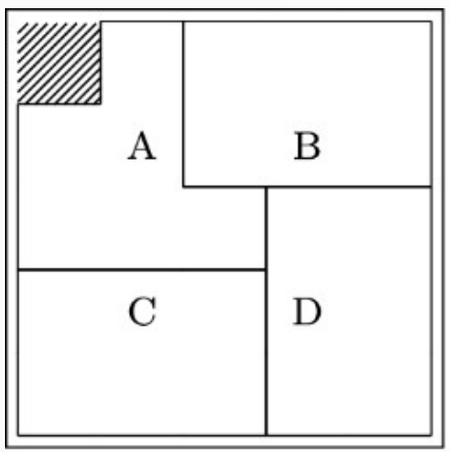
$\pi = 198$



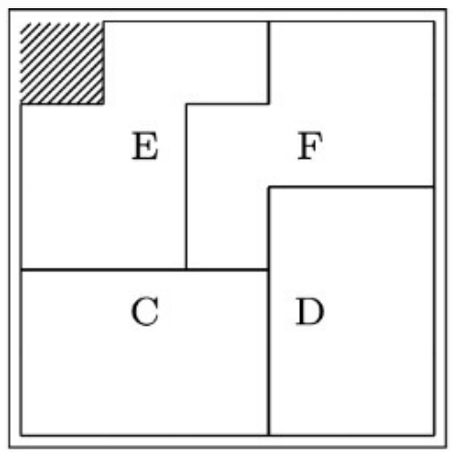
$\rho = 0$

*representing a ZPDB entry by tile **m**ap, **p**ermutation, and zero tile **r**egion*

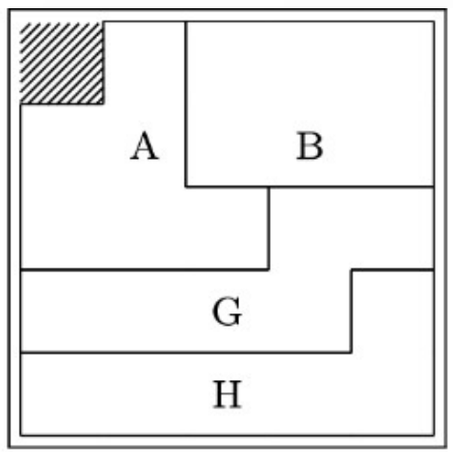
building catalogues



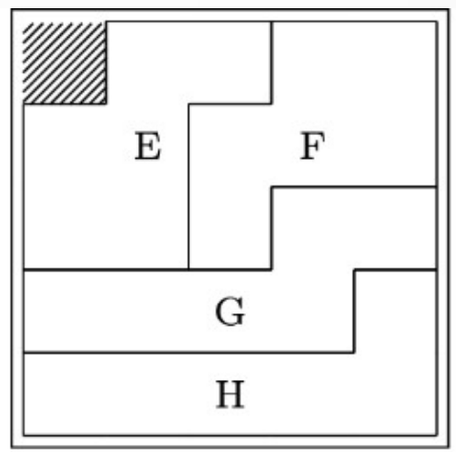
(a) 2.21%



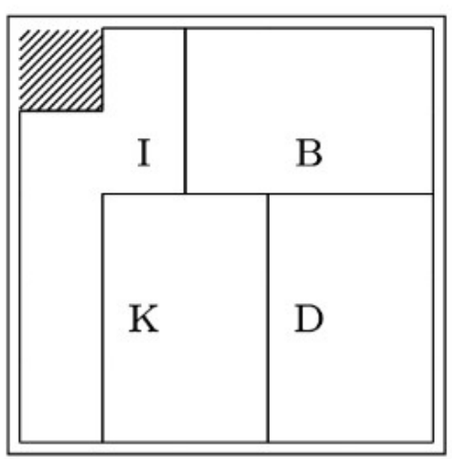
(b) 4.49%



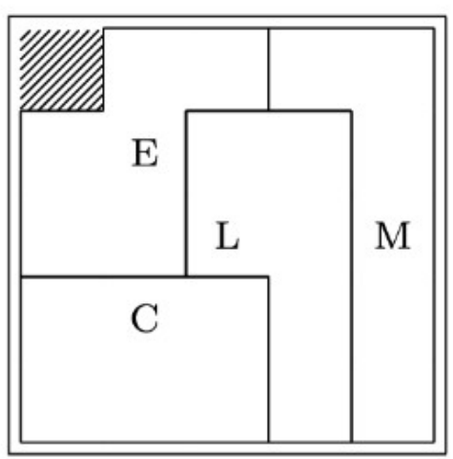
(c) 0.83%



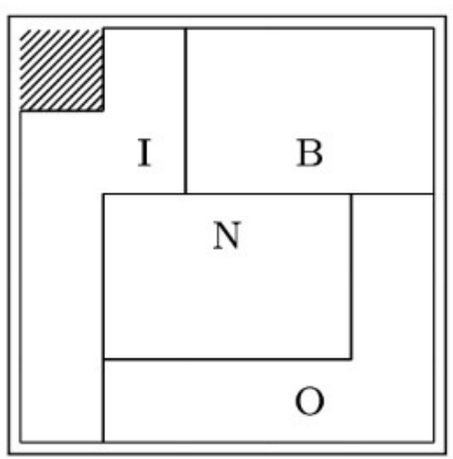
(d) 1.72%



(e) 5.18%

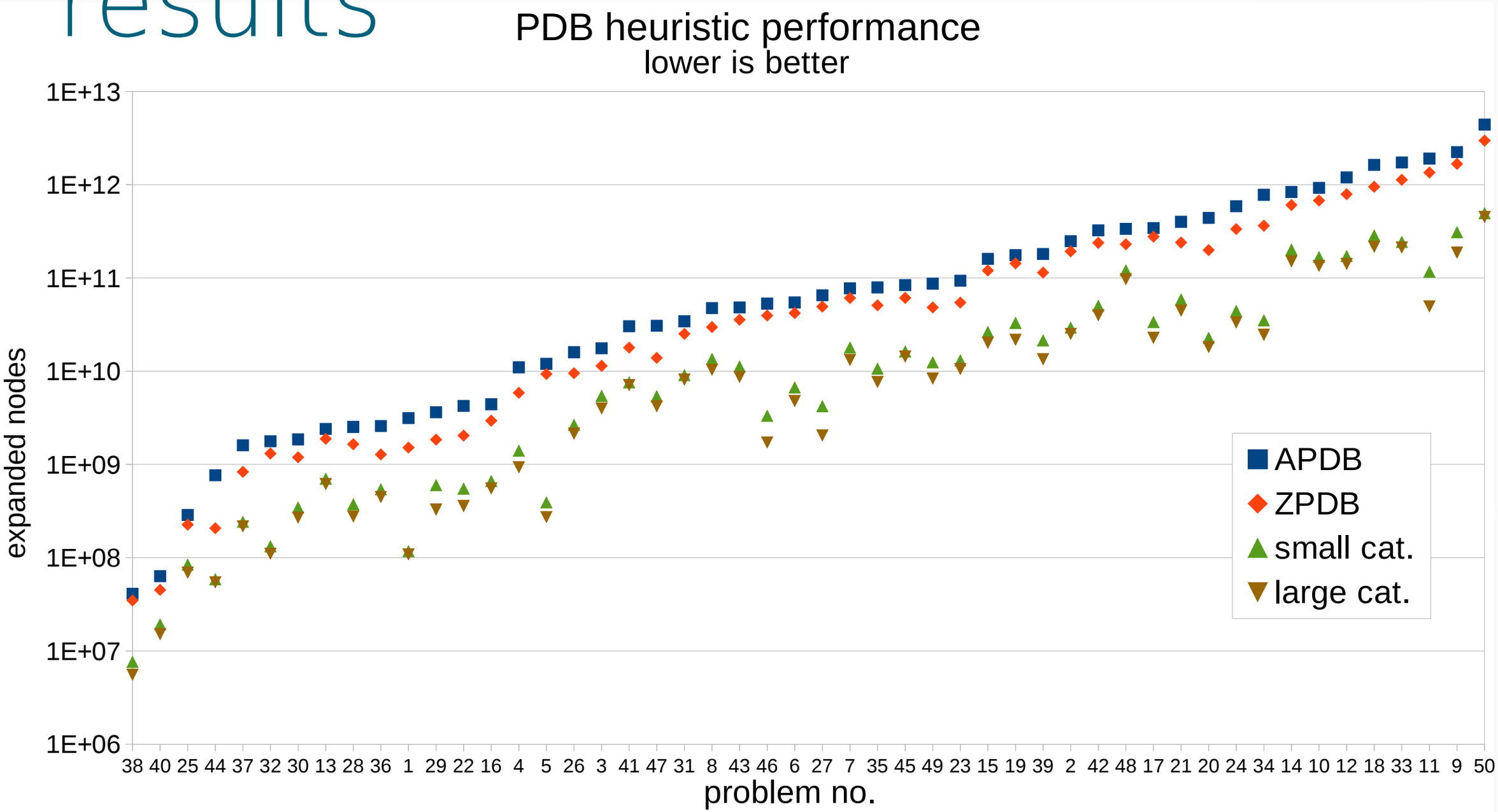


(f) 7.15%



(g) 4.77%

results


PDB heuristic performance
lower is better
expanded nodes
1E+13
1E+12
1E+11
1E+10
1E+09
1E+08
1E+07
1E+06
38 40 25 44 37 32 30 13 28 36 1 29 22 16 4 5 26 3 41 47 31 8 43 46 6 27 7 35 45 49 23 15 19 39 2 42 48 17 21 20 24 34 14 10 12 18 33 11 9 50
problem no.
APDB
ZPDB
small cat.
large cat.

conclusions

- the zero tile can be tracked explicitly at reasonable memory and performance costs
- tracking the zero tile explicitly reduces IDA* nodes by 37.8% on average
- a small catalogue of pattern databases additionally reduces IDA* nodes by 80.2% on average