Introduction to Artificial Intelligence

cuu duong than cong . com

Chapter 2: Solving Problems by Searching (4)
Heuristic Functions

cuu duong than cong . com

Nguyễn Hải Minh, Ph.D
nhminh@fit.hcmus.edu.vn

CuuDuongThanCong.com https://fb.com/tailieudientucntt

Outline

1. The 8-puzzle problem
2. Relaxed problems
3. Pattern databases
4. Learning heuristics from experience

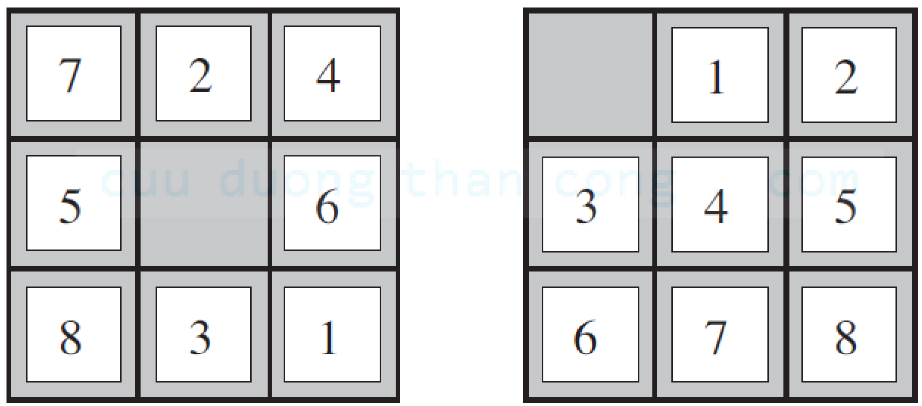
cuu duong than cong . com

cuu duong than cong . com

CuuDuongThanCong.com https://fb.com/tailieudientucntt

The (n^2-1)-Puzzle problem

- ❑Slide the tiles horizontally or vertically into the empty space until the configuration matches the goal state



Start State　　Goal State

- ❑States in (n^2-1)-Puzzle problem: $(n^2)!/2$
  - 181,440 possible states for 8-Puzzle
  - 1.05 x 10^{13} possible states for 15-Puzzle

cuu duong than cong . com

CuuDuongThanCong.com https://fb.com/tailieudientucntt

The 8-puzzle

- One of the earliest heuristic problems
- Average solution cost:
  - 22 steps
  - $b \approx 3$
  - A graph search: exhaustive search about 170k states (see HW 3.4)
- How about 15-puzzle?
  - About 10^{13} states
  - → Need a good heuristic to find the shortest solutions
  - → Never overestimates the number of steps to the goal

cuu duong than cong . com

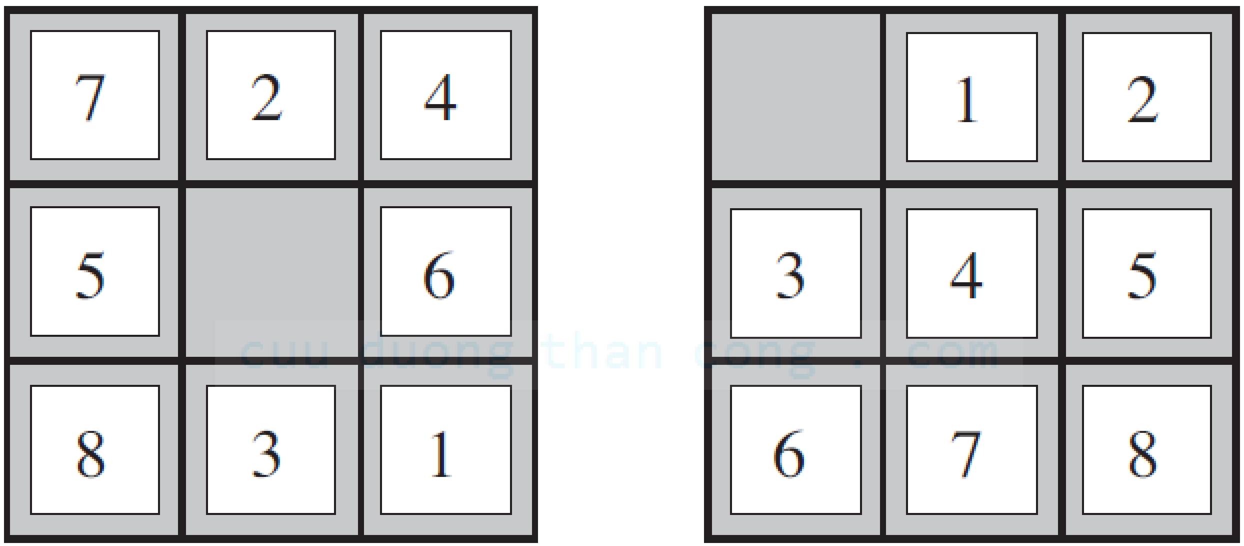
CuuDuongThanCong.com https://fb.com/tailieudientucntt

The 8-puzzle heuristics

- Admissible heuristic: $h(n) \leq h^*(n)$
- Which of the following heuristics is admissible?
  - $h_1(n) =$ total number of misplaced tiles
  - $h_2(n) =$ total Manhattan distance
  - $h_3(n) = 0$
  - $h_4(n) = 1$
  - $h_5(n) = h^*(n)$
  - $h_6(n) = \min(2, h^*(n))$
  - $h_7(n) = \max(2, h^*(n))$

CuuDuongThanCong.com https://fb.com/tailieudientucntt

The 8-puzzle: Admissible heuristics



Start State Goal State

cuu duong than cong . com

- $h_1(S) = 8$
- $h_{2(S)} = 3 + 1 + 2 + 2 + 2 + 3 + 3 + 2 = 18$

→ True solution cost: $h^*(S) = 26$

cuu duong than cong . com

CuuDuongThanCong.com https://fb.com/tailieudientucntt

Effective branching factor b*

- The branching factor that an uniform tree of depth *d* would have in order to contain *N+1* nodes

$$N + 1 = 1 + b^* + {b^*}^2 + \cdots + {b^*}^d$$

- Measure is fairly constant for sufficiently hard problems.
  - Can thus provide a good guide to the heuristic's overall usefulness.
  - A well-designed heuristic would have a value of b* close to 1

CuuDuongThanCong.com https://fb.com/tailieudientucntt

Search costs vs branching factors

	Search Cost (nodes generated)			Effective Branching Factor		
d	IDS	$A^*(h_1)$	$A^*(h_2)$	IDS	$A^*(h_1)$	$A^*(h_2)$
2	10	6	6	2.45	1.79	1.79
4	112	13	12	2.87	1.48	1.45
6	680	20	18	2.73	1.34	1.30
8	6384	39	25	2.80	1.33	1.24
10	47127	93	39	2.79	1.38	1.22
12	3644035	227	73	2.78	1.42	1.24
14	–	539	113	–	1.44	1.23
16	–	1301	211	–	1.45	1.25
18	–	3056	363	–	1.46	1.26
20	–	7276	676	–	1.47	1.27
22	–	18094	1219	–	1.48	1.28
24	–	39135	1641	–	1.48	1.26

CuuDuongThanCong.com https://fb.com/tailieudientucntt

Dominance

- If $h_2(n) \geq h_1(n)$ for all n (both admissible) then h_2 dominates h_1
  - h_2 is better for search
- Typical search costs (average number of nodes expanded):
  - $d=12$ IDS = 3,644,035 nodes
    $A^*(h_1)$ = 227 nodes
    $A^*(h_2)$ = 73 nodes
  - $d=24$ IDS = too many nodes
    $A^*(h_1)$ = 39,135 nodes
    $A^*(h_2)$ = 1,641 nodes

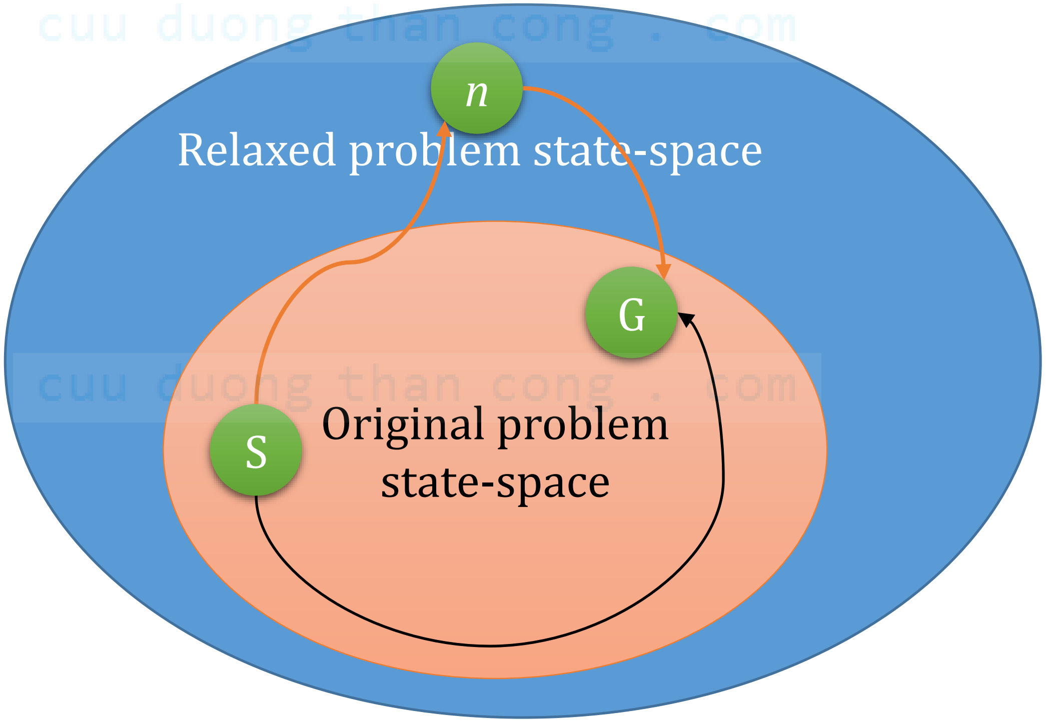
cuu duong than cong . com
cuu duong than cong . com

CuuDuongThanCong.com https://fb.com/tailieudientucntt

Relaxed problems

- A problem with fewer restrictions on the actions is called a relaxed problem
- The cost of an optimal solution to a relaxed problem is an admissible heuristic for the original problem

cuu duong than cong . com



cuu duong than cong . com

Relaxed problems

- Original problem:
  - A tile canmove from square A to square B if A is horizontally or vertically adjacent to B **and** B is blank
- Relaxed problem:
  a) A tile can move from square A to square B if A is adjacent to B.
  b) A tile can move from square A to square B if B is blank.
  c) A tile can move from square A to square B.

Manhattan distance

Misplaced tiles

it is crucial that the relaxed problems generated by this technique can be solved essentially *without search*

cuu duong than cong . com

cuu duong than cong . com

Relaxed problems

❑A collection of admissible heuristics: $h_1, h_2, \ldots, h_m$

→*Composite heuristic function:*

$$h(n) = \max\{h_1(n), h_2(n), \ldots, h_m(n)\}$$

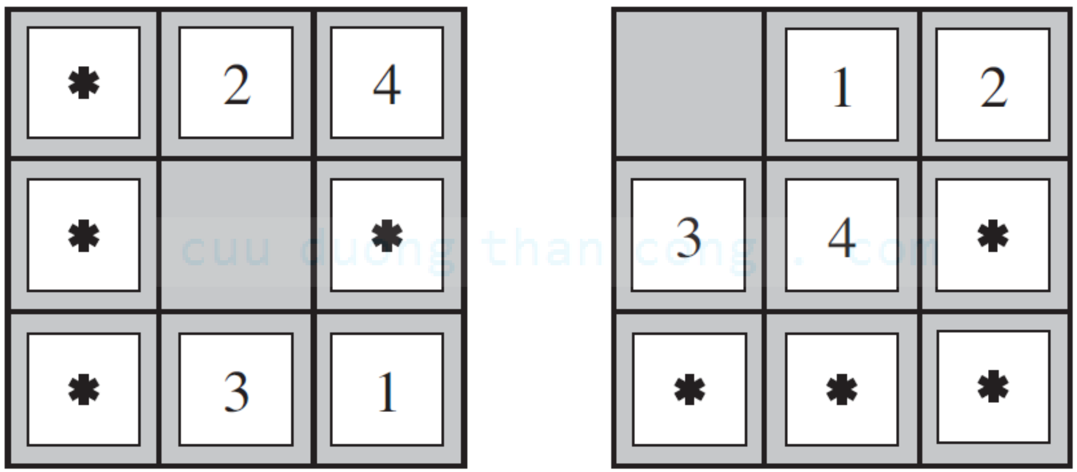
→ It is easy to prove that h is consistent. Furthermore, h dominates all of its component heuristics

cuu duong than cong . com

CuuDuongThanCong.com https://fb.com/tailieudientucntt

Pattern database

- ❑A subproblem of the 8-puzzle instance



Start State   Goal State

- ❑Solution cost of this subproblem ≤ real cost of the original problem
  - Some cases, it is more accurate than Manhattan distance

CuuDuongThanCong.com https://fb.com/tailieudientucntt

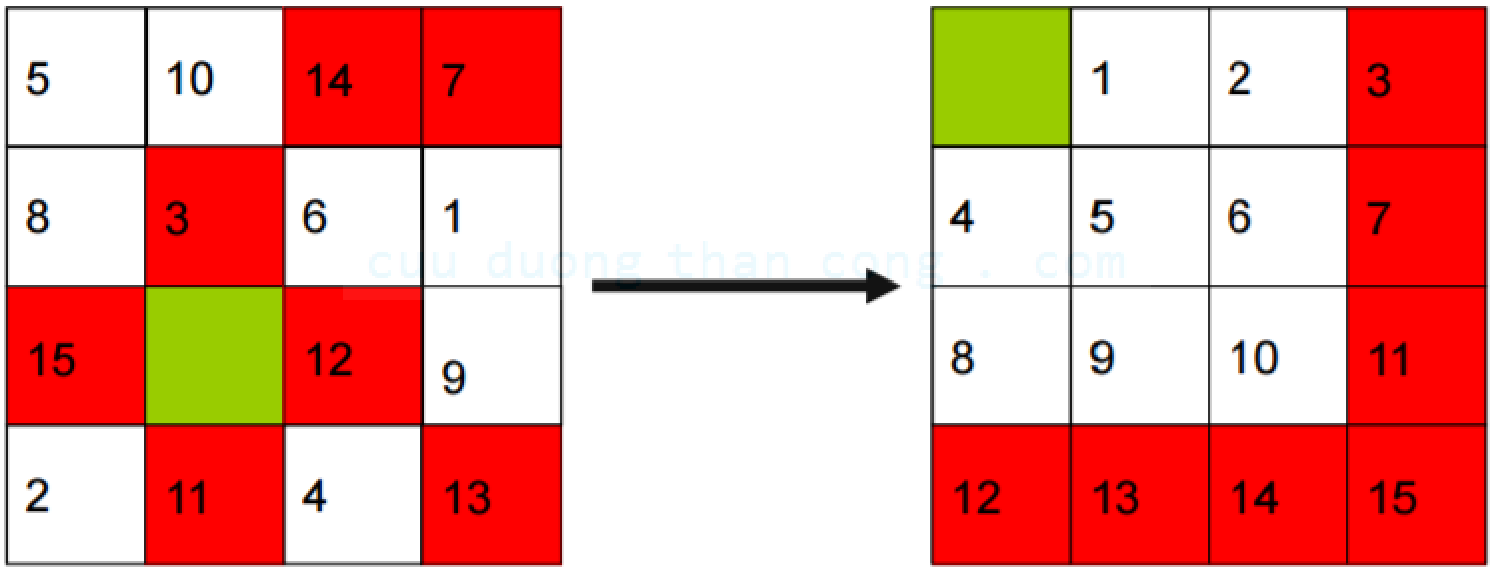
Pattern database

- ❑Admissible heuristics can also be derived from the solution cost of a subproblem of a given problem.
- ❑This cost is a lower bound on the cost of the real problem.
- ❑Idea of pattern databases:
  - Store the exact solution costs for every possible subproblem instance.
  - The complete heuristic is constructed using the patterns in the databases

cuu duong than cong . com

CuuDuongThanCong.com https://fb.com/tailieudientucntt

Heuristic from PDB

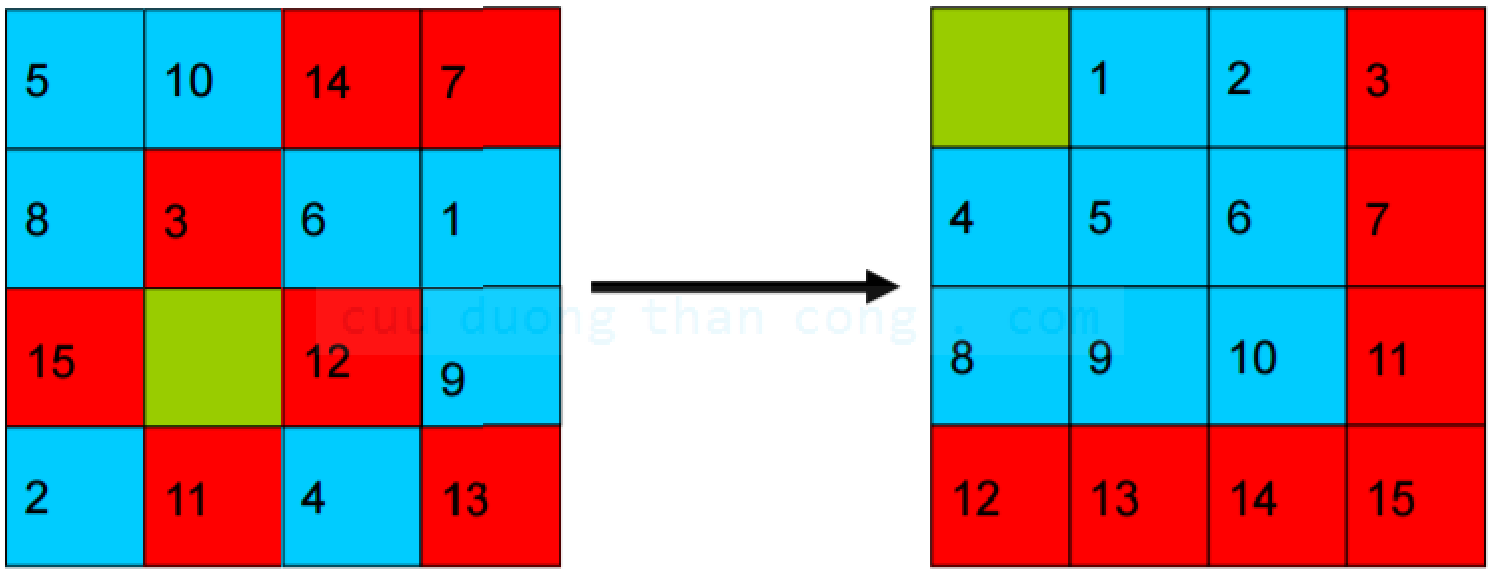


31 moves is a lower bound on the total number of moves needed to solve this particular state

https://courses.cs.washington.edu/courses/cse473/12sp/slides/04-heuristics.pdf

CuuDuongThanCong.com https://fb.com/tailieudientucntt

Heuristic from PDB



cuu duong than cong . com

31 moves needed to solve red tiles

22 moves needed to solve blue tiles

→ Overall heuristic is maximum of 31 moves

https://courses.cs.washington.edu/courses/cse473/12sp/slides/04-heuristics.pdf

CuuDuongThanCong.com https://fb.com/tailieudientucntt

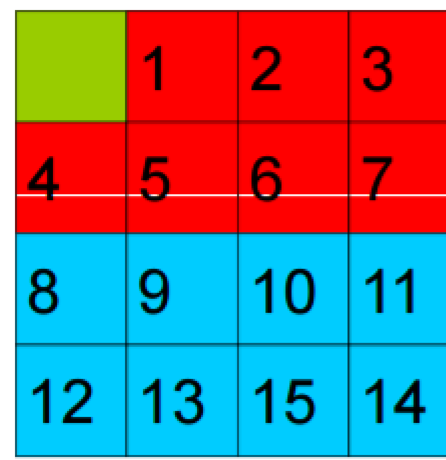
Additive Pattern Databases

- Count only moves of the pattern tiles, ignoring non-pattern moves.
  - If no tile belongs to more than one pattern, then we can add their heuristic values.
  - Manhattan distance is a special case of this, where each pattern contains a single tile.

→Disjoint Pattern Databases

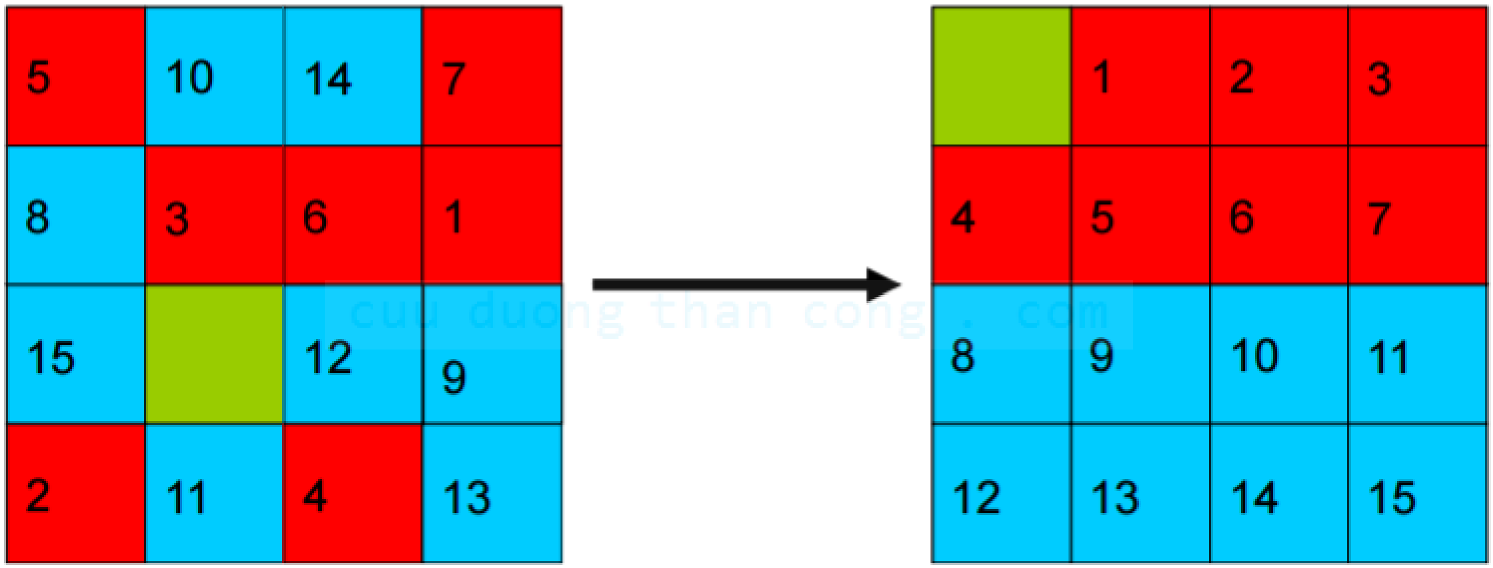
cuu duong than cong . com

- The 7-tile database contains 58 million entries.
- The 8-tile database contains 519 million entries.



CuuDuongThanCong.com https://fb.com/tailieudientucntt

Additive Pattern Databases



20 moves needed to solve red tiles

25 moves needed to solve blue tiles

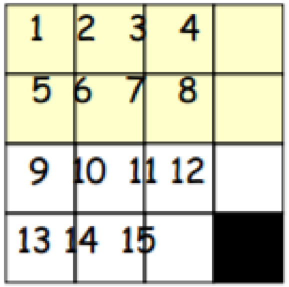
→ Overall heuristic is 20+25=45 moves

CuuDuongThanCong.com https://fb.com/tailieudientucntt

Performance

❑ 15 Puzzle:

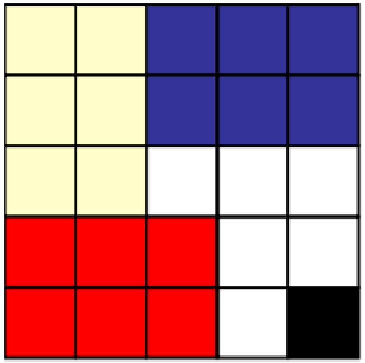
- 2000x speedup vs Manhattan distance
- IDA* with the two DBs solves 15 Puzzles optimally in 30 milliseconds



cuu duong than cong . com

❑ 24 Puzzle:

- 12 million x speedup vs Manhattan
- IDA* can solve random instances in 2 days.
- Requires 4 DBs as shown
- Each DB has 128 million entries
- Without PDBs: 65,000 years



cuu duong than cong . com

CuuDuongThanCong.com https://fb.com/tailieudientucntt

Learning heuristics from experience

- Experience means:
  - e.g, solving lots of 8-puzzles
  - Each optimal solution to an 8-puzzle problem provides examples from which $h(n)$ can be learned
- Learning algorithms:
  - Neural nets
  - Decision trees
  - Inductive learning
  - ...

cuu duong than cong . com

cuu duong than cong . com

CuuDuongThanCong.com https://fb.com/tailieudientucntt