Optimally Solving Cooperative Path-Finding Problems Without Hole on Rectangular Boards with Heuristic Search

Bruno Bouzy

Paris Descartes University

WoMPF 2016

July 10, 2016

Outline

- Cooperative Path-Finding without Hole (CPFwH)
- Numbers, Databases and Heuristic Search
- Rubik cube : a 2-stage algorithm for CPFwH ?
- Stage-one and stage-two heuristics
- Tuple-based heuristics
- Experimental results
  - In the size of the boards
  - Heuristic functions' evaluation
- Conclusions

CPF rules

- On a grid (graph)
  - 4-connectivity (or 8-connectivity)
  - (obstacles)
- Rules:
  - An agent can move on the neighbouring cell or stay
  - Rule 0: no two agents can be on the same cell
  - Rule 1: no two agents can swap
  - Circularity is possible
- A = MxN
  - A : number of agents
  - M : width of the board
  - N : height of the board

CPF example (0/6)

Start

4	8	0
7	5	3
2	1	6

Goal

0	1	2
3	4	5
6	7	8

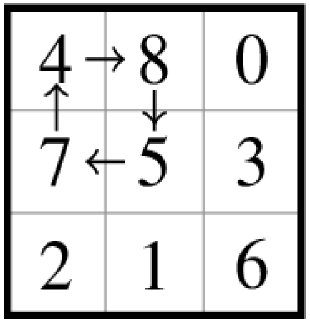
CPF example (1/6)

Start

4	8	0
7	5	3
2	1	6

Goal

0	1	2
3	4	5
6	7	8



4 → 8		0
↑ 7 ← 5 ↓		3
2	1	6

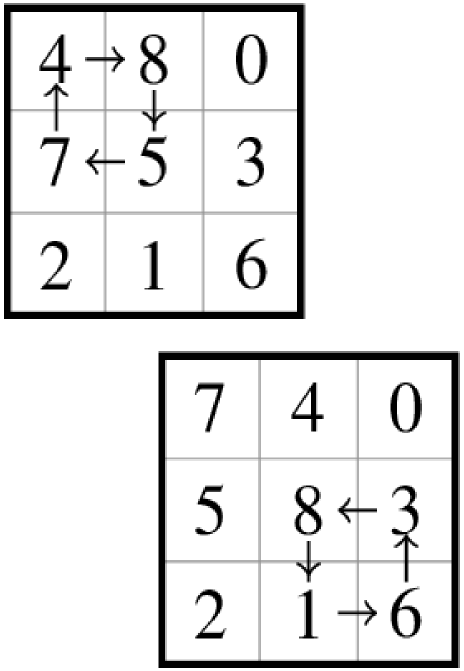
CPF example (2/6)

Start

4	8	0
7	5	3
2	1	6

Goal

0	1	2
3	4	5
6	7	8



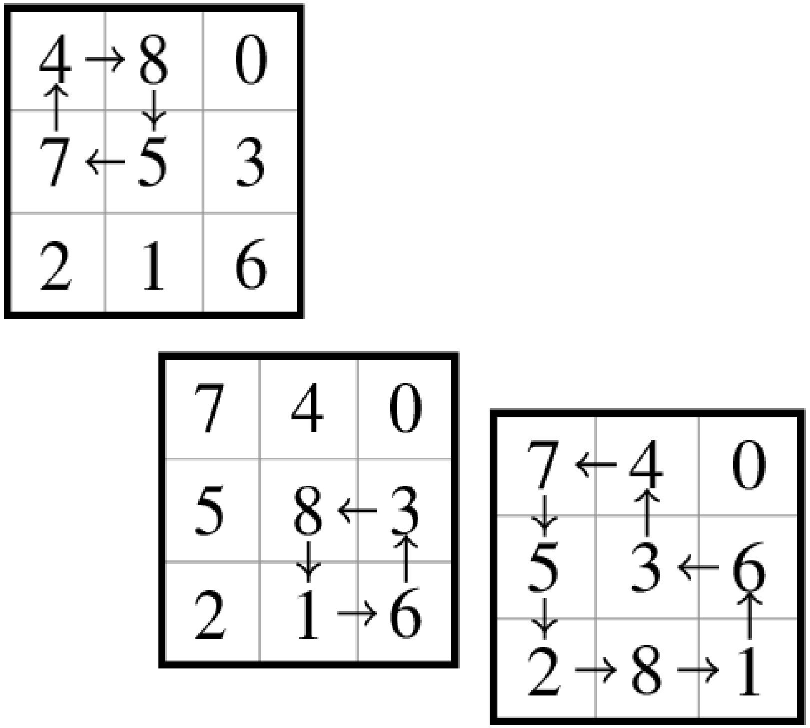
CPF example (3/6)

Start

4	8	0
7	5	3
2	1	6

Goal

0	1	2
3	4	5
6	7	8



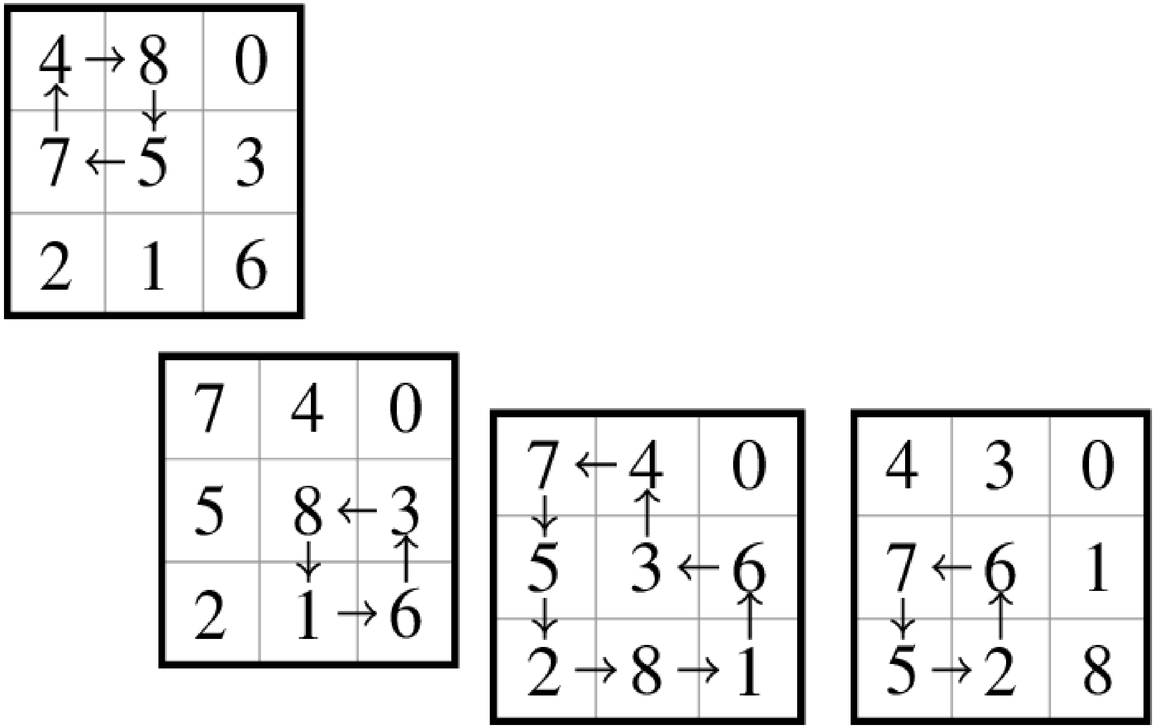
CPF example (4/6)

Start

4	8	0
7	5	3
2	1	6

Goal

0	1	2
3	4	5
6	7	8



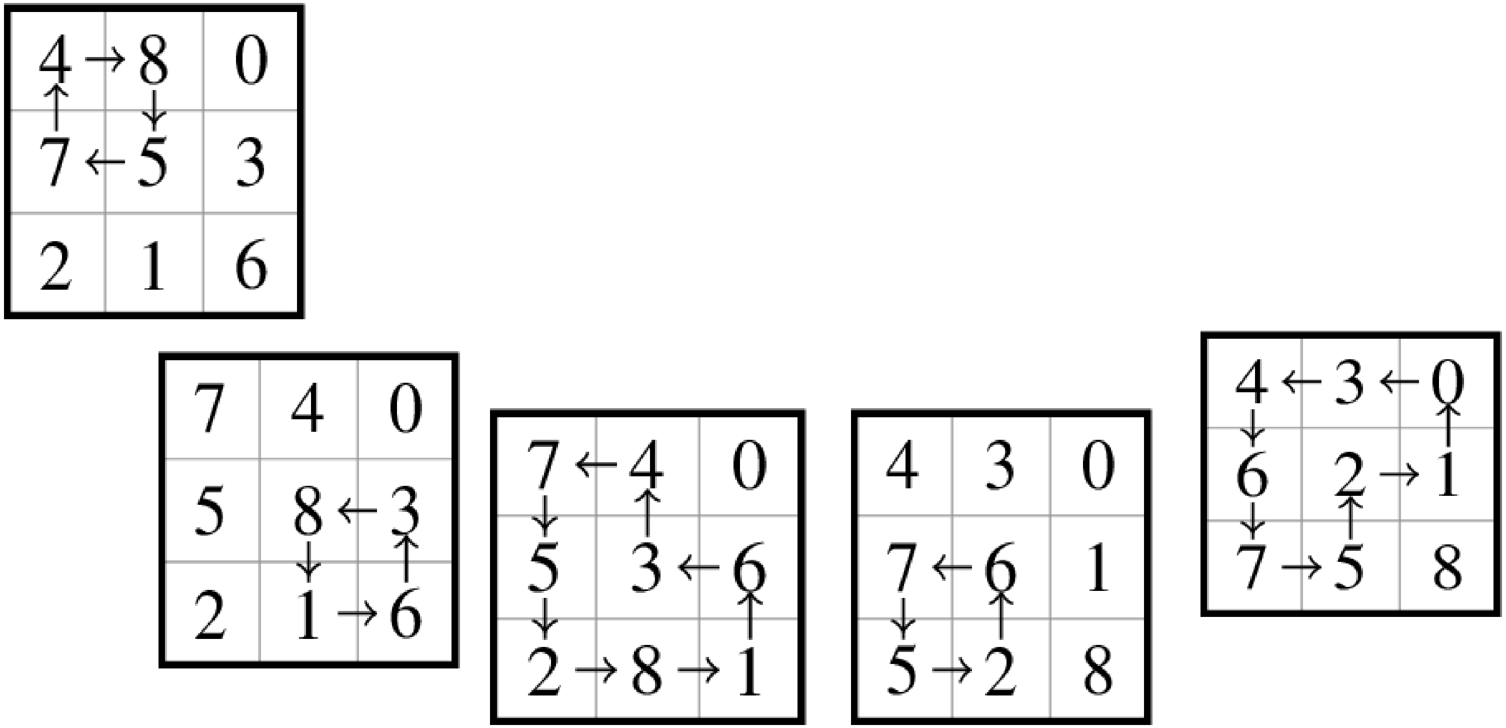
CPF example (5/6)

Start

4	8	0
7	5	3
2	1	6

Goal

0	1	2
3	4	5
6	7	8



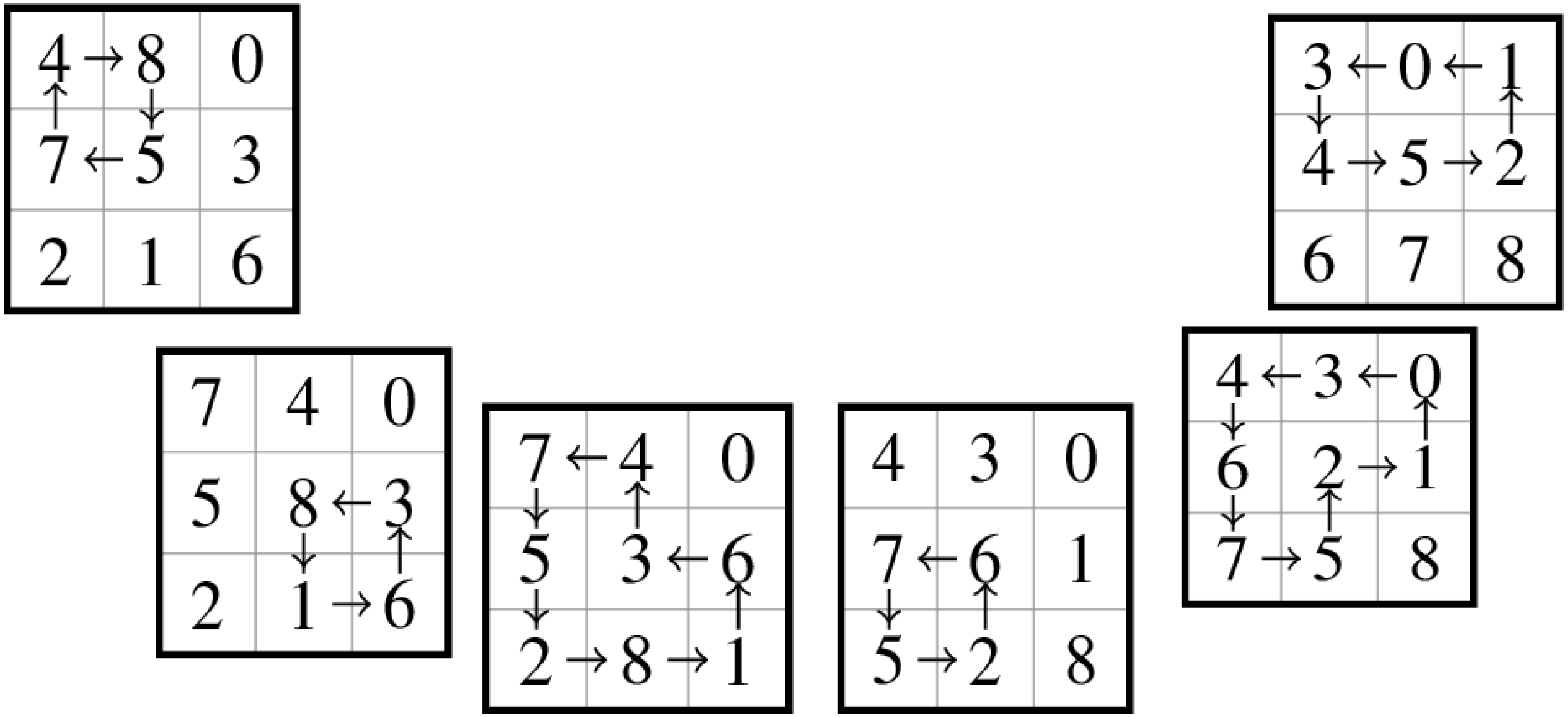
CPF example (6/6)

Start

4	8	0
7	5	3
2	1	6

Goal

0	1	2
3	4	5
6	7	8



CPF optimization target

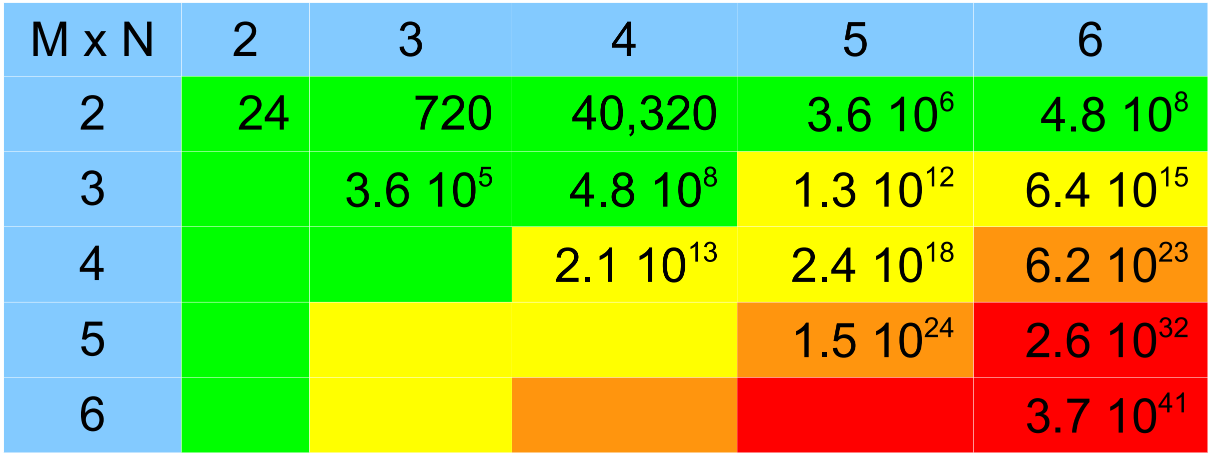
- Sequentiality
  - sum of individual costs
  - i. e. Total number of elementary actions
  - Most of work in the literature
- Simultaneity
  - Global elapsed time (T) = Number of timesteps
  - Our work + TOMPP (Yu 2012)

CPF previous work

- WHCA* (Silver 2006)
- BIBOX (Surynek 2009)
- A* + OD (Standley 2010), A* + ID (Standley 2011)
- Multi-Agent Path Planning (MAPP) (Botea 2011)
- Push & Swap (Luna 2011), TASS (Khorshid 2011)
- M* (Wagner 2011)
- Time-Optimal Multi-Agent Path Planning (TOMPP) (Yu 2012)
    - Linear Programming (LP) approach
- CPF video game benchmarks (Sturtevant 2012)
- Conflict-Based Search (CBS) (Sharon 2012)
- Pebble Motion on Graphs with Rotations (Yu 2015)

Number of states

- #states == A !

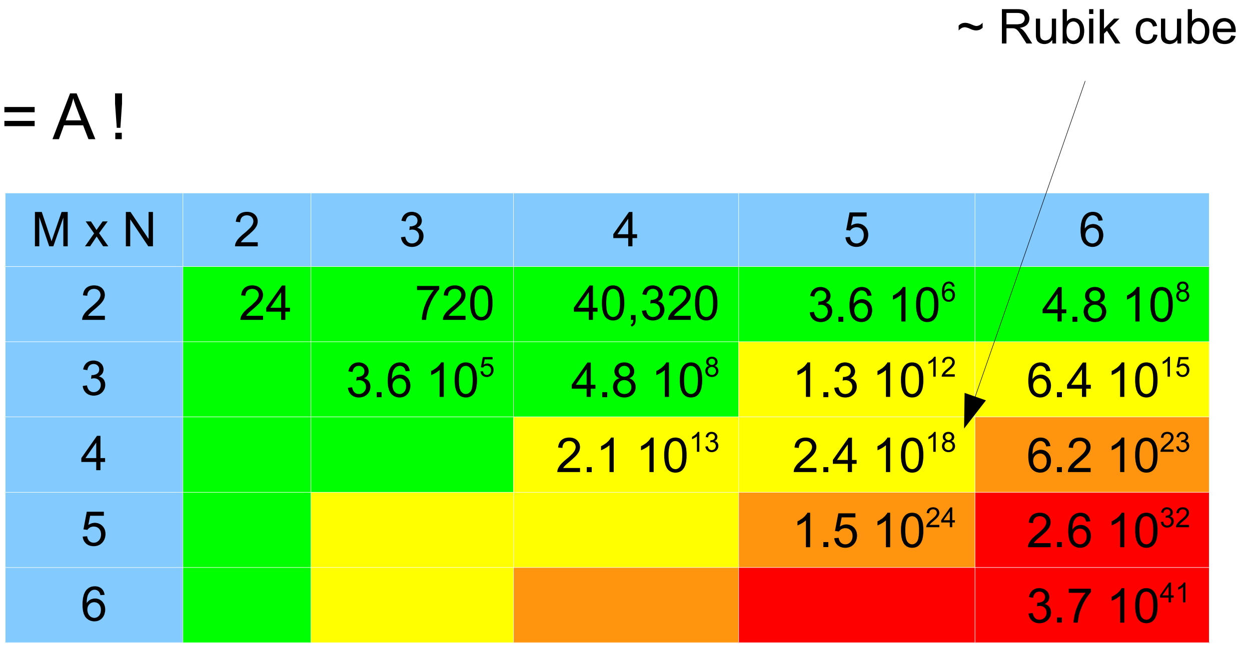


M x N	2	3	4	5	6
2	24	720	40,320	3.6 10^{6}	4.8 10^{8}
3	🟩	3.6 10^{5}	4.8 10^{8}	1.3 10^{12}	6.4 10^{15}
4	🟩	🟩	2.1 10^{13}	2.4 10^{18}	6.2 10^{23}
5	🟩	🟨	🟨	1.5 10^{24}	2.6 10^{32}
6	🟩	🟨	🟧	🟥	3.7 10^{41}

Simple	Doable	Hard	Very Hard
Database	Heuristic Search	Heuristic Search	Approximate Methods

Number of states

- #states == A !



~ Rubik cube

M x N	2	3	4	5	6
2	🟩 24	🟩 720	🟩 40,320	🟩 $3.6\ 10^{6}$	🟩 $4.8\ 10^{8}$
3	🟩	🟩 $3.6\ 10^{5}$	🟩 $4.8\ 10^{8}$	🟨 $1.3\ 10^{12}$	🟨 $6.4\ 10^{15}$
4	🟩	🟩	🟨 $2.1\ 10^{13}$	🟨 $2.4\ 10^{18}$	🟧 $6.2\ 10^{23}$
5	🟩	🟨	🟨	🟧 $1.5\ 10^{24}$	🟥 $2.6\ 10^{32}$
6	🟩	🟨	🟧	🟥	🟥 $3.7\ 10^{41}$

🟩 Simple	🟨 Doable	🟧 Hard	🟥 Very Hard
🟩 Database	🟨 Heuristic Search	🟧 Heuristic Search	🟥 Approximate Methods

Number of joint actions

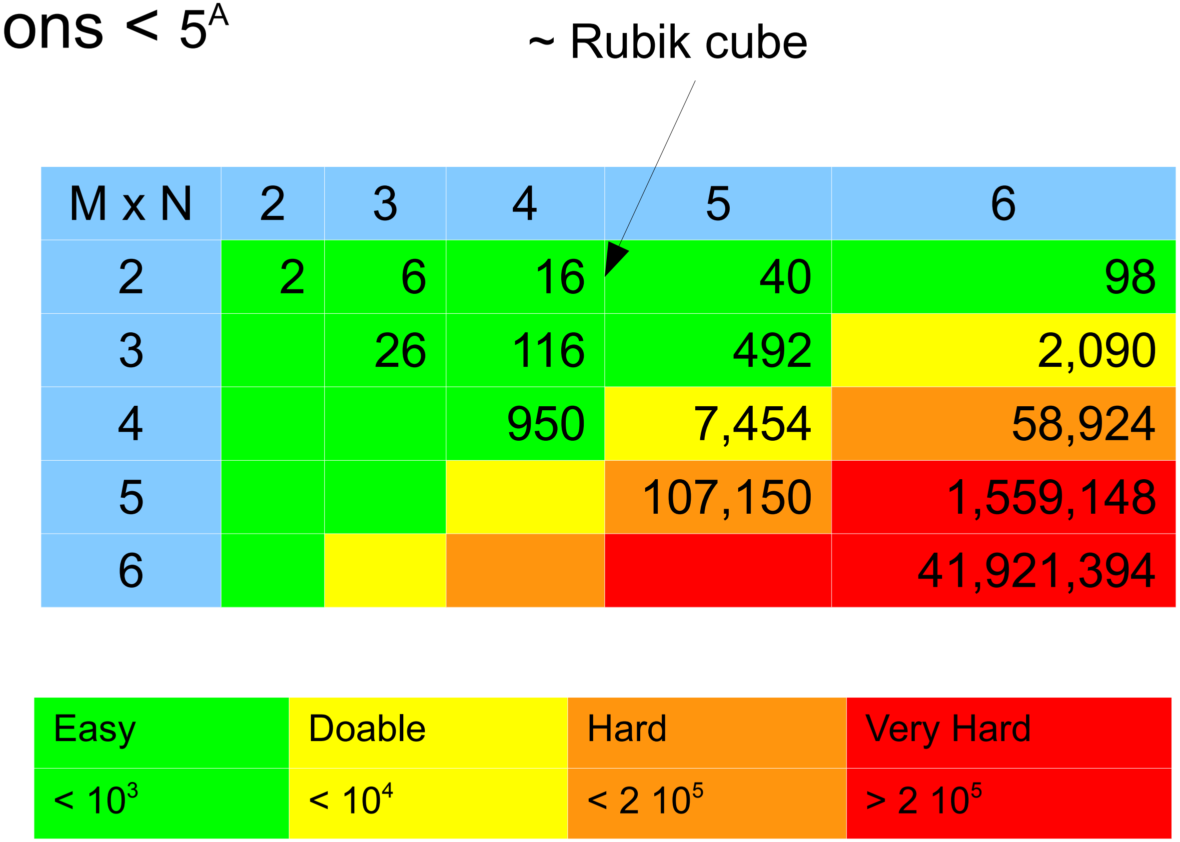
- #jointActions < 5^A

M x N	2	3	4	5	6
2	2	6	16	40	98
3		26	116	492	2,090
4			950	7,454	58,924
5				107,150	1,559,148
6					41,921,394

Easy	Doable	Hard	Very Hard
$< 10^3$	$< 10^4$	$< 2\ 10^5$	$> 2\ 10^5$

Number of joint actions

- #jointActions < 5^A



~ Rubik cube

M x N	2	3	4	5	6
2	2	6	16	40	98
3	🟩	26	116	492	2,090
4	🟩	🟩	950	7,454	58,924
5	🟩	🟩	🟨	107,150	1,559,148
6	🟩	🟨	🟧	🟥	41,921,394

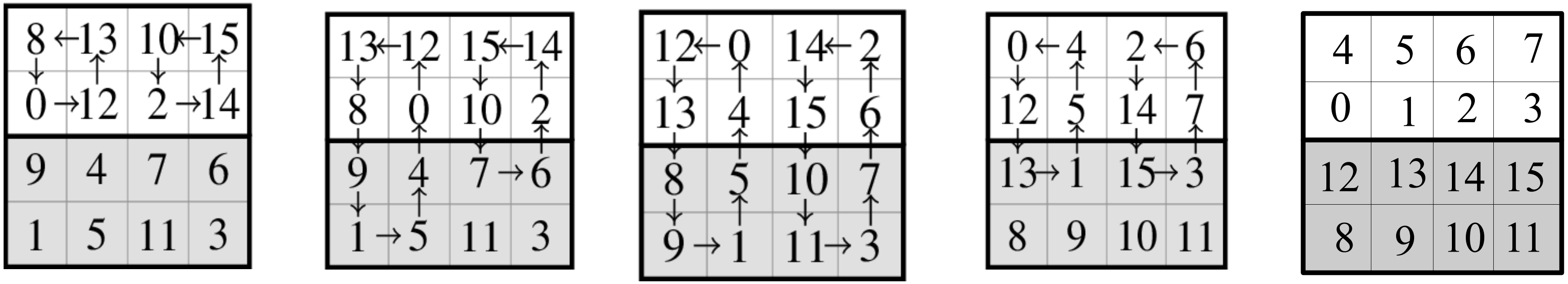
Easy	Doable	Hard	Very Hard
$< 10^3$	$< 10^4$	$< 2\ 10^5$	$> 2\ 10^5$

The 2-stage algorithm for Rubik cube

- Stage 1
  - Goal : reach a coset
    - Up and Down facets located in the Up and Down faces
    - Intermediate cubies located in the intermediate slice
  - All actions
- Stage 2
  - Goal : solve the Rubik cube
  - 10 Actions : Left2, Right2, Front2, Back2, Up+, Up-, Down+, Down-, Up2, Down2
- Good properties
  - When in stage 2, distance to the goal is admissible
  - When in stage 1, distance to the coset is admissible
  - Fitting computer memory : #states = 10^{18} #statestage1 ≈ 10^{9} #stage2 ≈ 10^{9}
  - (Provide a domain-dependent simulation for Monte-Carlo)
- (Kociemba 1990')

2-stage algorithm for CPFwH

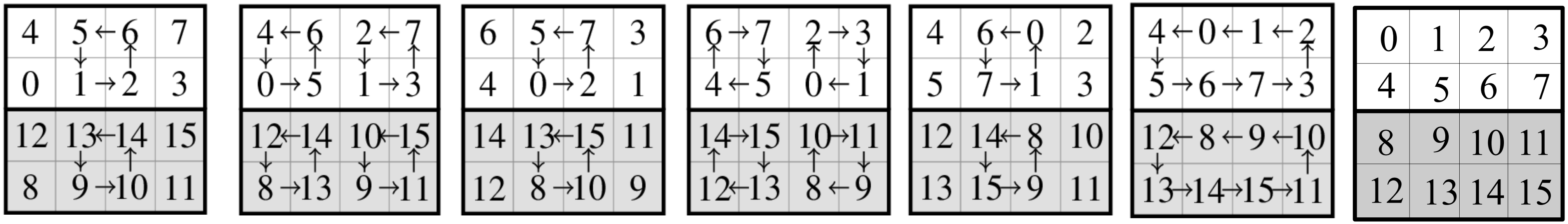
- Divide the board into two sub-boards.
- Stage one



- The first 8 agents are moving up to the white area and the last 8 agents are moving down to the grey area.
- Goal : the first 8 agents are situated in the white area
- 4 moves

2-stage algorithm for CPFwH

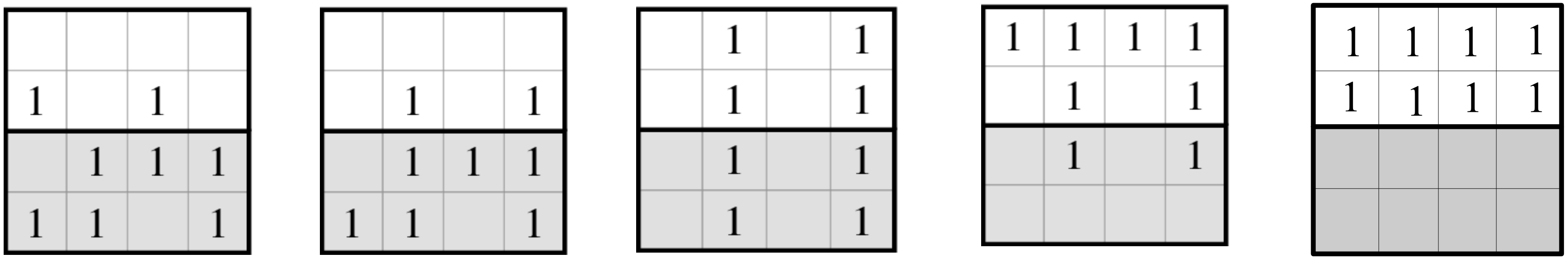
- Solve each sub-board separately
- Stage two



- The 2x4 board at the top and the 2x4 board at the bottom are solved separately and independently.
- 6 moves for each 2x4 board ==> 6 moves for stage two.

Stage-one heuristics

- HSO = distance to the stage one goal



1		1	
	1	1	1
1	1		1

	1		1
	1	1	1
1	1		1

	1		1
	1		1
	1		1
	1		1

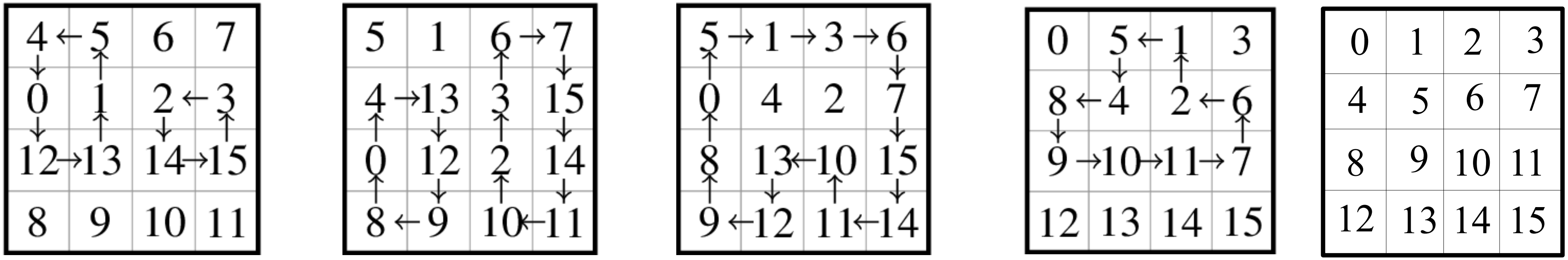
1	1	1	1
	1		1
	1		1

1	1	1	1
1	1	1	1

- HSO==4 HSO==3 HSO==2 HSO==1 HSO==0
- Size S of the database for stage one : C_{nxN}^{Nxb}
- B==2, N==4 ==> S = 12,870
- B==2, N==5 ==> S = 3,268,760
- Building the database : Breadth First Search
- Admissible

Stage-two heuristics

- Not admissible
- Unfortunately solving two sub-problems separately is sub-optimal.
- Example : optimal sequence on the starting position of stage two :



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

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

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

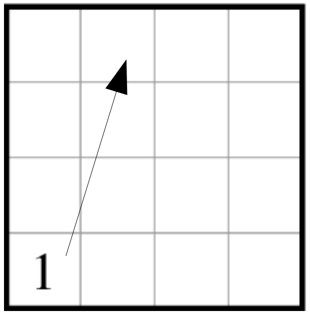
0	5	1	3
8	4	2	6
9	10	11	7
12	13	14	15

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

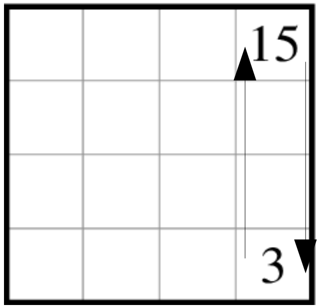
- 4 moves < 6 moves

Tuple-based heuristics

- Maxspan over the single agents is a better heuristic than HSO
- HMT_T = Maxspan over Tuples of T agents with T : size of the tuple.
- Example :

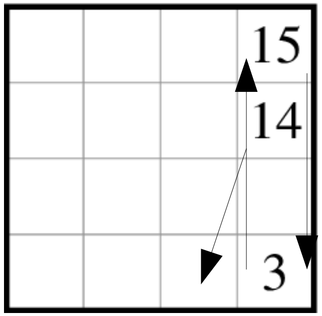


$HMT_1 == 4$



$HMT_1 == 3$

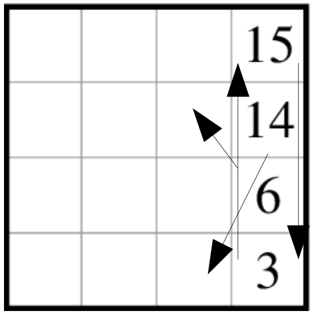
$HMT_2 == 5$



$HMT_1 == 3$

$HMT_2 == 5$

$HMT_3 == 5$



$HMT_1 == 3$

$HMT_2 == 5$

$HMT_3 == 5$

$HMT_4 == 5$

Experiments

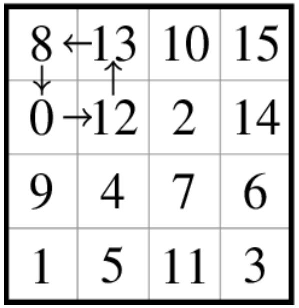
- Set of problems
  - 4x4 : 100 problems drawn at random
  - 5x5 : 10 easy problems set up manually + the Yu's problem.

- 3.2 Ghz CPU with 6 Gbytes memory

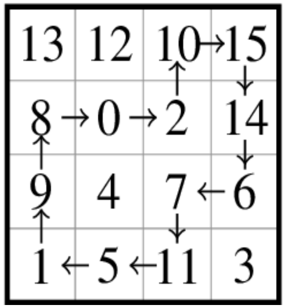
Optimal solution on a 4x4 board

- L==6
- Few seconds (IDA* + HMT$_3$)

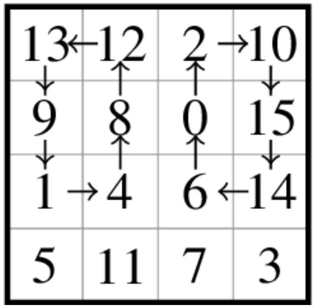
0	1	2	3
4	5	6	7
8	9	10	11
12	13	14	15



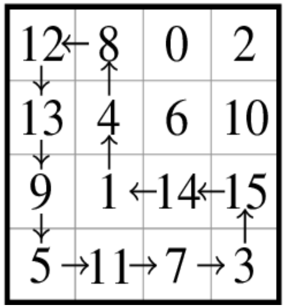
8	13	10	15
0	12	2	14
9	4	7	6
1	5	11	3



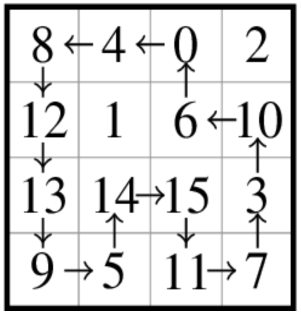
13	12	10	15
8	0	2	14
9	4	7	6
1	5	11	3



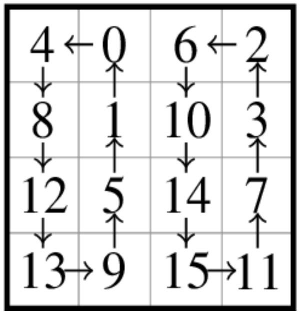
13	12	2	10
9	8	0	15
1	4	6	14
5	11	7	3



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



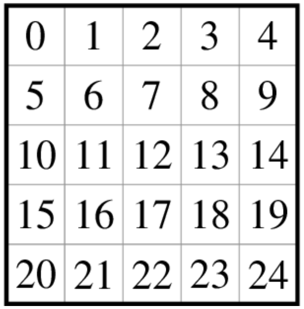
8	4	0	2
12	1	6	10
13	14	15	3
9	5	11	7

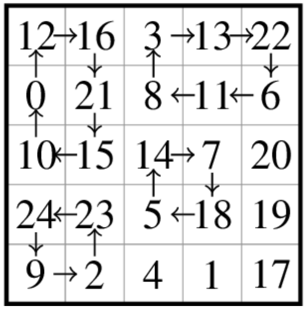


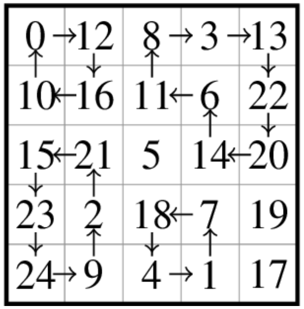
4	0	6	2
8	1	10	3
12	5	14	7
13	9	15	11

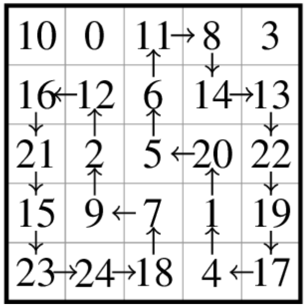
Optimal solution on a 5x5 board

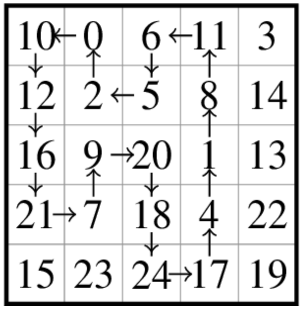
- The Yu's position : L==7
- One day (IDA* + HMT$_4$)

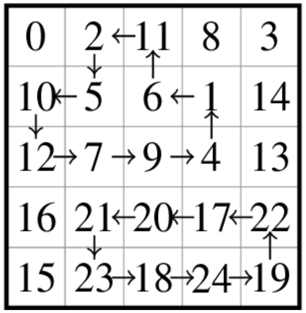


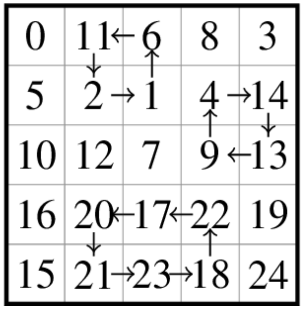


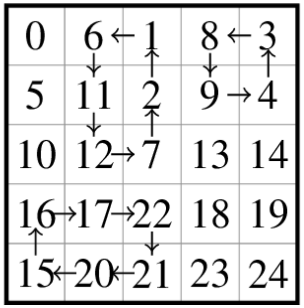












Assessing stage-one heuristics

- HSO alone does not help, i.e. HSO is worse than HMT_1
- HMT_1 + HSO versus HMT_1 alone
  - Number of nodes (nn) slightly decreased.
  - Time (t) is multipied by 8.

		HMT_1 alone		HMT_1 + HSO	
Pos.	L	t	nn	t	nn
B6	6	0.2s	259k	0.7s	257k
B1	6	0.7s	2.1M	5s	2.1M
S5	7	11s	42M	90s	39M
B7	7	16s	66.5M	150s	66M
C2	7	2m	434M	16m	433M
B3	6	12m	2.70G	1h40	2.70G
C1	7	8m	1.70G	1h	1.63G
S7	7	16m	3.77G	2h20	3.67G

Assessing tuple-based heuristics

- 4x4 boards, T = 2, 3, 4
  - Number of nodes (nn) decreases with T
  - Time (t) is minimal for T=3

		T = 2		T = 3		T = 4	
Pos.	L	t	nn	t	nn	t	nn
B6	6	0.2s	180k	0.2s	120k	0.2s	75k
B1	6	0.4s	960k	0.4s	670k	0.6s	430k
S5	7	5s	18M	5s	12M	5s	5.5M
B7	7	10s	38M	7s	19M	10s	10M
C2	7	30s	132M	25s	71M	40s	45M
B3	6	1m	210M	25s	59M	35s	34M
C1	7	2m	520M	90s	270M	150s	165M
S7	7	6m	1.3G	4m	590M	5m	340M

Assessing tuple-based heuristics

- 5x5 boards, T = 2, 3, 4
  - Number of nodes (nn) decreases with T
  - Time (t) is minimal for T=4

		T = 2		T = 3		T = 4	
Pos.	L	t	nn	t	nn	t	nn
SA2	3	8s	10M	4s	2.3M	3s	1.5M
SLC2	4	2h	9.1M	4m	420M	3m	380M
SL2	4			23m	2.2G	7m	730M
SL1	4			24m	2.0G	12m	1.7G
SLC1	4			1h	6.5G	24m	3.3G
SL5	5			26h	164G	10h	77G
SL3	5			33h	199G	8h	55G
YU	7			90h	900G	23h	210G

Conclusions and perspectives

- Heuristic Search can solve MxN CPFwH problems for M and N <= 5
- Transferring the Rubik's cube technique to CPFwH is disappointing.
  - Stage-two heuristics is not admissible.
  - Stage-one heuristics is admissible but not tight.
- Tuple-based heuristics are efficient.
- Heuristic Search seems dominated by LP approaches.
- Future work :
  - T > 4 ?
  - M or N > 5 ?
  - Obstacles ?
  - Embedding within a more general solver ?

Thank you for your attention!

- Questions ?
- bruno.bouzy@parisdescartes.fr