Hardness of Rubiks Table Problems

Corbet Elkins

July 16, 2025

Purdue University

Rubiks Table

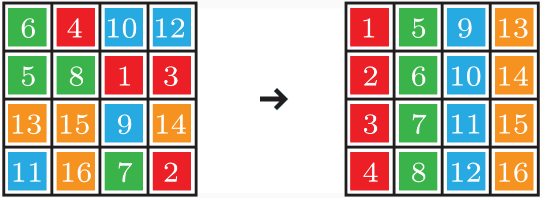


Figure 1: Rubik Table Problem [1, Fig. 2]

- Two main variations, labeled and unlabeled Rubiks Tables.
- Our goal is to minimize the number of column/row shuffles to reach the desired state.

Background

- Primary interest is for multi-robot path planning on a grid

Lemma
An arbitrary Rubiks Table problem on an $m_1 \times m_2$ *table can be solved using* $m_1 + m_2$ *shuffles. The labeled Rubiks Table problem can be solved using* $2m_1 + m_2$ *shuffles.*

- These shuffles are of the form m_1 column shuffles, then m_2 row shuffles, followed by m_1 column shuffles

Parallel (Labeled) Rubiks Table Algorithm

- Find a perfect matching of a bipartite graph
- Shuffle column according to matching
- Shuffle rows into correct positions
- If it is a labeled Rubiks Table shuffle columns again

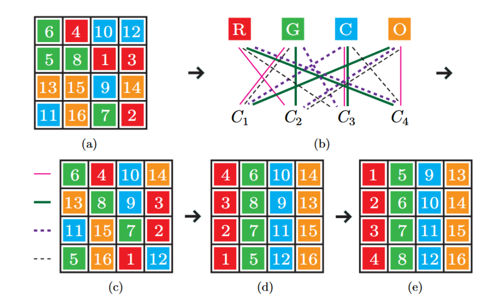


Figure 2: Rubik Table Diagram [1, Fig. 4]

Hardness questions

Problem
Is minimizing the number of shuffles in the Parallel Rubiks Table Scheme NP-hard?

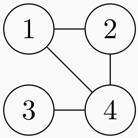
Problem
Is minimizing the number of shuffles in the Parallel Labeled Rubiks Table Scheme NP-hard?

Problem
Is minimizing the number of shuffles to solve a Rubiks Table NP-hard?

Conflict Graph

C_1	C_2	C_3	C_4
1	2	3	3
1	1	2	2
4	4	3	4
3	4	1	2

Rubiks Table

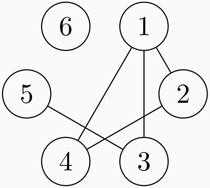


Conflict Graph

- Conflict graphs give a general understanding of the table

Lemma
The number of column shuffles in the Parallel Rubiks Table Scheme is bounded below by the min vertex cover of the conflict graph.

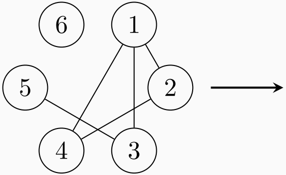
- Goal: Given a graph G, generate a Rubiks table, T, such that the conflict graph of T is isomorphic to G.
- For every edge, we want to place the same number in the two column associated with the vertices connected via that edge



1	1				
2		2			
3			3		
	4		4		
		5		5	

Reduction cont.

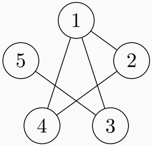
- We cannot however easily fill the table so it becomes a Rubiks table.
- To fix this we fill the table with the next sequential numbers.



1	1	6	7	8	9
2	10	2	11	12	13
3	14	15	3	16	17
18	4	19	4	20	21
22	23	5	24	5	25

Reduction cont.

- We then construct our Rubiks table by appending several of these small tables with slight variations.



		C_1					C_2		
1	1	6	7	8	99	100	98	97	98
2	9	2	10	11	6	6	7	8	1
3	12	13	3	14	9	10	9	11	2
15	4	16	4	17	12	13	14	12	3
18	19	5	20	5	16	15	17	15	4
21	21	26	27	28	19	20	18	5	18
22	29	22	30	31	26	26	27	28	21
23	32	33	23	34	29	30	29	31	22
35	24	36	24	37	32	33	34	32	23
38	39	25	40	25	36	35	37	35	24

$\cdots$

$\vdots$ $\vdots$ $\ddots$

Reduction cont.

- Our Rubiks table, T, no longer has a conflict graph isomorphic to G but to several copies of G.
- For any vertex cover of G, we can shuffle the respective columns of T to the desired result.
- We then can relabel the numbers so that there are no cells in the correct column.
- Thus if we can calculate the minimum number of shuffles, we can determine the size of the min vertex set of G.

Results

Theorem
Minimizing the number of shuffles in the Parallel Rubiks Table Scheme is NP-hard.

Theorem
Minimizing the number of shuffles in the Parallel Labeled Rubiks Table Scheme is NP-hard.

Theorem
Minimizing the number of shuffles to solve a Rubiks Table is NP-hard.

Acknowledgements

- I'd like to thank my mentor Dr. Jingjin Yu
- Additionally, thanks to the DIMACS REU 2025 program
- I would like to thank the NSF, with this research being supported

by NSF grant CCF-2447342

References

[1] Mario Szegedy and Jingjin Yu. *Rubik Tables and Object Rearrangement*. 2023. arXiv: 2002.04979 [cs.RO]. URL: https://arxiv.org/abs/2002.04979.