

Permutation Match Puzzles: How Young Tanvi Learned About Computational Complexity

Kshitij Gajjar ✉ 

IIIT Hyderabad, India

Neeldhara Misra ✉ 

IIT Gandhinagar, India

Abstract

We study a family of sorting match puzzles on grids, which we call permutation match puzzles. In this puzzle, each row and column of a $n \times n$ grid is labeled with an ordering constraint – ascending (A) or descending (D) – and the goal is to fill the grid with the numbers 1 through n^2 such that each row and column respects its constraint.

We provide a complete characterization of solvable puzzles: a puzzle admits a solution if and only if its associated constraint graph is acyclic, which translates to a simple “at most one switch” condition on the A/D labels. When solutions exist, we show that their count is given by a hook length formula. For unsolvable puzzles, we present an $O(n)$ algorithm to compute the minimum number of label flips required to reach a solvable configuration. Finally, we consider a generalization where rows and columns may specify arbitrary permutations rather than simple orderings, and establish that finding minimal repairs in this setting is NP-complete by a reduction from feedback arc set.

2012 ACM Subject Classification Theory of computation → Design and analysis of algorithms; Mathematics of computing → Combinatorics

Keywords and phrases sorting match puzzles, permutation match puzzles, grid puzzles, constraint satisfaction, directed acyclic graphs, hook length formula, standard Young tableaux, NP-completeness, feedback arc set

Digital Object Identifier 10.4230/LIPIcs.FUN.2026.20

Acknowledgements We would like to thank the Center for Creative Learning (CCL) at IIT Gandhinagar for introducing us to this puzzle. We also acknowledge the use of an AI assistant (Claude) for the preparation of this draft. We are also grateful to the reviewers of FUN 2026 for their feedback.



■ **Figure 1** A 3×3 sorting match puzzle: the physical edition that inspired this work.



© Kshitij Gajjar and Neeldhara Misra;

licensed under Creative Commons License CC-BY 4.0

13th International Conference on Fun with Algorithms (FUN 2026).

Editor: John Iacono; Article No. 20; pp. 20:1–20:16



Leibniz International Proceedings in Informatics

LIPIcs Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

1 Introduction

Tanvi, while wandering the corridors of IIT Gandhinagar, ambles into the Centre for Creative Learning (CCL), where she is intrigued by a really simple looking puzzle – also shown in Figure 1. From the looks of it, it is a 3×3 grid on a foam board with nine removable pieces labeled from 1 to 9. It reminds her of a magic square, but here each row and column is labelled by the word *ascending* or *descending* – it turns out these are also removable bits that can be flipped over to change the label, so the board appears to hold sixty-four puzzles in all!

She wonders what the puzzle demands. After asking around, she finds that the goal is to reorganize the numbers in the grid such that each row and column is in the appropriate order. She wonders if there is a way to solve each of the sixty-four puzzles in a systematic way. A playable version of this puzzle can be found here.

In this work, we explore general versions of this puzzle, providing a fairly complete picture of questions of solvability, counting solutions, and the issue of optimally perturbing unsolvable puzzles to make them solvable.

Consider the following natural generalization of the 3×3 puzzle found at CCL: given an $n \times n$ grid where each row and column is labeled either A (ascending) or D (descending), fill the grid with numbers from 1 to n^2 such that each row (respectively, column) labeled A must contain numbers in ascending order and each row (respectively, column) labeled D must contain numbers in descending order.

	A	D	D		D	D	D		D	D	A		A	A	D
A	1	6	8	D	9	7	3	D	9	8	4	D	7	4	3
D	7	5	4	D	8	5	2	A	3	5	6	D	8	5	2
D	9	3	2	D	6	4	1	A	1	2	7	D	9	6	1

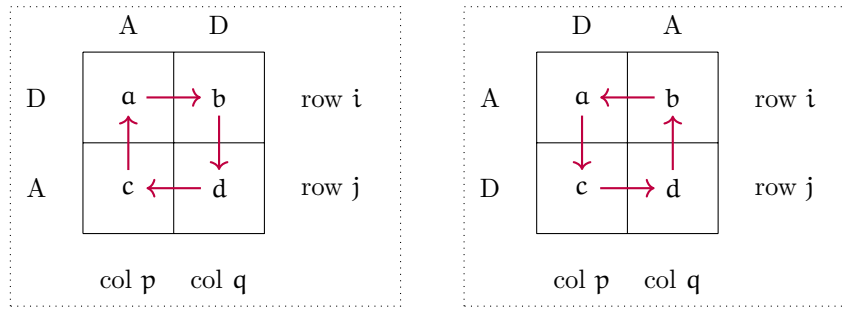
■ **Figure 2** Examples of 3×3 ascending-descending grid puzzles with valid solutions.

	D	D	A		A	A	D		D	A	D		A	D	A
D				A				D				A			
A				D				A				D			
D				A				D				A			

■ **Figure 3** Four examples of 3×3 ascending-descending grid puzzles with no solution.

The examples in Figure 3 show configurations that have no valid solutions. To understand why these puzzles are unsolvable, we first observe a key structural property that creates an impossible situation: if there exist rows i, j with $i < j$ labeled D and A (or A and D) respectively, and columns p, q with $p < q$ labeled A and D (or D and A) respectively, then we have a circular constraint, illustrated in Figure 3.

Consider the four cells at positions (i, p) , (i, q) , (j, p) , and (j, q) , containing values a , b , c , and d respectively. Following the arrows in the diagram, the constraints force a cyclic chain of inequalities: $a < b < c < d < a$ or $a < c < d < b < a$, depending on the labeling, which is clearly impossible.



■ **Figure 4** Two illustrations of circular constraints in unsolvable configurations. Left: rows labeled D,A with columns A,D, creating the cycle $a < b < d < c < a$. Right: rows labeled A,D with columns D,A, creating the cycle $a < c < d < b < a$. Both cycles are impossible to satisfy with distinct values.

We now formally define the notion of a *sorting match puzzle*.

► **Definition 1** (Sorting Match Puzzles). A *sorting match puzzle* of order n is a pair of words (r, c) , each of length n , over the alphabet $\{A, D\}$. A *sorting match puzzle* (r, c) is solvable if there exists a permutation of the integers $1, 2, \dots, n^2$ arranged in an $n \times n$ grid such that:

- For each $i \in \{1, \dots, n\}$, if $r_i = A$, the entries in the i^{th} row are in ascending order; if $r_i = D$, the entries are in descending order.
- For each $j \in \{1, \dots, n\}$, if $c_j = A$, the j^{th} column is in ascending order; if $c_j = D$, the entries are in descending order.

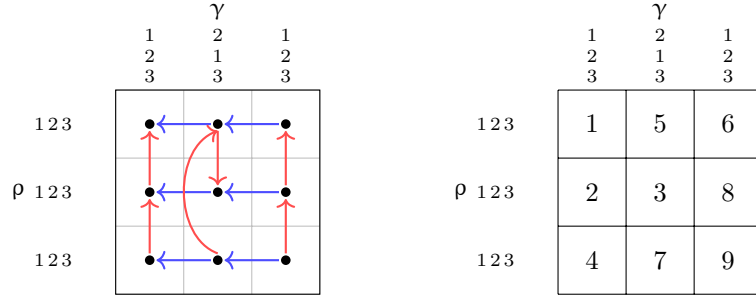
We also call a sorting match puzzle row (respectively, column) uniform if all its row (resp., column) labels are identical.

► **Definition 2** (Uniform Sorting Match puzzles). A *sorting match puzzle* (r, c) is *row-uniform* if $r = A^n$ or $r = D^n$; a *sorting match puzzle* (r, c) is *col-uniform* if $c = A^n$ or $c = D^n$; and a *sorting match puzzle* (r, c) is *uniform* if it is both row-uniform and col-uniform.

A first natural question we want to address is the following: when is a sorting match puzzle solvable? Our first result is that the scenarios described in Figure 4 are the *only* obstructions to solvability. In particular, solvable puzzles are either row or column uniform, or are of the form $r = A^k D^{n-k}$ and $c = A^\ell D^{n-\ell}$ for some $k, \ell \in \{1, \dots, n-1\}$ or $r = D^k A^{n-k}$ and $c = D^\ell A^{n-\ell}$ for some $k, \ell \in \{1, \dots, n-1\}$. This characterization also tells us that there are $4 \cdot (2^n) + 2 \cdot (n-1)^2$ solvable puzzles.

Next, we want to explore both solvable and unsolvable puzzles further. For solvable puzzles, we would like to count the number of solutions. For unsolvable puzzles, we would like to know the minimum number of label flips required to make the puzzle solvable. We discover that the answer to the first question involves the hook length formula, while for the second question notice that by guessing the “shape” of the nearest solvable puzzle, we can find the answer in $O(n^2)$ time by brute force. We improve this to a linear time algorithm using a simple dynamic programming approach. We note that our ideas here extend to d -dimensional grids too.

Notice that ascending and descending orders can be thought of as two special permutations over $[n]$. This suggests the following natural generalization of the puzzle above: where the labels can be arbitrary permutations over $[n]$, and a solution is a permutation of the integers $1, 2, \dots, n^2$ that satisfies the ordering constraints imposed by the labels – see Figure 5 for an illustration. This motivates our next definition.



■ **Figure 5** A generalized sorting match puzzle with row permutations $\rho_1 = \rho_2 = \rho_3 = (1, 2, 3)$ (all ascending) and column permutations $\gamma_1 = \gamma_3 = (1, 2, 3)$ (ascending) but $\gamma_2 = (2, 1, 3)$. Left: The constraint graph where an arrow $P \rightarrow Q$ indicates $P > Q$ (blue for rows, red for columns); arrows implied by transitivity are omitted for clarity. For column 2, the permutation $(2, 1, 3)$ requires: middle entry < top entry < bottom entry. Right: A valid solution where column 2 has $3 < 5 < 7$.

► **Definition 3** (Permutation Match Puzzles). A permutation match puzzle of order n is a pair (ρ, γ) , where $\rho = (\rho_1, \dots, \rho_n)$ and $\gamma = (\gamma_1, \dots, \gamma_n)$ are sequences of permutations, with each $\rho_i, \gamma_j \in S_n$. A generalized sorting match puzzle (ρ, γ) is solvable if there exists a permutation of the integers $1, 2, \dots, n^2$ arranged in an $n \times n$ grid such that:

- For each $i \in \{1, \dots, n\}$, if the entries in the i^{th} row are $a_1, a_2, \dots, a_n$ (from left to right), then $a_{\rho_i^{-1}(1)} < a_{\rho_i^{-1}(2)} < \dots < a_{\rho_i^{-1}(n)}$.
- For each $j \in \{1, \dots, n\}$, if the entries in the j^{th} column are $b_1, b_2, \dots, b_n$ (from top to bottom), then $b_{\gamma_j^{-1}(1)} < b_{\gamma_j^{-1}(2)} < \dots < b_{\gamma_j^{-1}(n)}$.

Sorting match puzzles are a special case of permutation match puzzles where each ρ_i and γ_j is either the identity permutation (corresponding to A) or the reversal permutation (corresponding to D).

By interpreting the permutation constraints as directed edges on a grid, we observe that determining if a permutation match puzzle is solvable or not reduces to cycle detection. Permutation match puzzles inspire the following question: given a directed graph G on the vertex set $\{(i, j) \mid 1 \leq i, j \leq n\}$, where the subgraph induced by the vertices $R_i := \{(i, j) \mid 1 \leq j \leq n\}$ and $C_j := \{(i, j) \mid 1 \leq i \leq n\}$ is acyclic for each $i, j \in [n]$, what is the minimum number of edges we need to flip to make the G acyclic? This is essentially a feedback arc set problem on a restricted class of directed graphs. Our final result is that this problem is NP-complete.

Related Work. Permutation match puzzles belong to a rich tradition of combinatorial puzzles with algebraic and algorithmic underpinnings. The most classical example is the 15-puzzle, whose solvability was characterized by Johnson and Story [8] using permutation parity: a configuration is reachable if and only if it corresponds to an even permutation. While checking solvability is polynomial-time, Ratner and Warmuth [9] showed that finding the shortest solution for generalized $n \times n$ sliding puzzles is NP-hard.

Our counting results connect to the theory of posets and Young tableaux. The problem of counting linear extensions of a poset is #P-complete in general [3], making our closed-form formula for sorting match puzzles particularly notable. The formula we derive is reminiscent of the celebrated hook length formula of Frame, Robinson, and Thrall [6], which counts standard Young tableaux of a given shape. Greene, Nijenhuis, and Wilf [7] gave an elegant probabilistic proof of this formula using a “hook walk” argument, which bears conceptual similarity to our recursive decomposition approach.

The structural constraints in permutation match puzzles also relate to geometric grid classes of permutations studied by Albert, Atkinson, Bouvel, Ruškuc, and Vatter [2], where permutations are constrained by monotonicity conditions on grid cells. Grid-based constraint satisfaction more broadly connects to Latin squares and Sudoku puzzles [5, 10], though our puzzles differ in using global ordering constraints rather than local distinctness conditions.

Our NP-completeness result for the minimum repair problem relates to feedback arc set problems. While minimum feedback arc set is NP-hard even for tournaments [4], and approximation algorithms exist for general instances [1], our reduction shows hardness persists even when the underlying graph has the restricted structure arising from grid-based permutation constraints.

2 Sorting Match Puzzles

We first establish a characterization of solvable sorting match puzzles.

► **Theorem 4.** *A sorting match puzzle (r, c) is solvable if and only if it is either row-uniform or col-uniform, or satisfies one of the following conditions:*

1. *There exists $k \in \{1, \dots, n-1\}$ and $\ell \in \{1, \dots, n-1\}$ such that:*
 - *The first k rows are labeled A and the remaining rows are labeled D, and*
 - *The first ℓ columns are labeled A and the remaining columns are labeled D.*
2. *There exists $k \in \{1, \dots, n-1\}$ and $\ell \in \{1, \dots, n-1\}$ such that:*
 - *The first k rows are labeled D and the remaining rows are labeled A, and*
 - *The first ℓ columns are labeled D and the remaining columns are labeled A.*

Proof. If (r, c) is col-uniform, say with $c = A^n$, then fill the grid from top-left to bottom-right, row by row, with the entries in the i^{th} row being the numbers $\{(i-1)n+1, \dots, (i-1)n+n\}$ in ascending order if $r_i = A$ and in descending order if $r_i = D$. If $c = D^n$, then fill the grid from bottom-right to top-left, row by row, with the entries in the i^{th} row from the bottom being the numbers $\{(i-1)n+1, \dots, (i-1)n+n\}$ in ascending order if $r_i = A$ and in descending order if $r_i = D$. An analogous reasoning applies if (r, c) is row-uniform.

Now consider the case when there exists $k \in \{1, \dots, n-1\}$ and $\ell \in \{1, \dots, n-1\}$ such that the first k rows are labeled A and the remaining rows are labeled D, and the first ℓ columns are labeled A and the remaining columns are labeled D. Let:

- $P^{\nwarrow}$ denote the cells included in the intersection of the first k rows and the first ℓ columns
- $P^{\nearrow}$ denote the cells included in the intersection of the first k rows and the remaining columns
- $P^{\swarrow}$ denote the cells included in the intersection of the remaining rows and the first ℓ columns
- $P^{\searrow}$ denote the cells included in the intersection of the remaining rows and the remaining columns

One solution to this puzzle is the following: populate $P^{\nwarrow}$ with the first $k\ell$ numbers in ascending order, filling this sub-grid row by row, top-left to bottom-right, and populate $P^{\searrow}$ with the next $(n-k)(n-\ell)$ numbers in descending order, filling this sub-grid row by row, top-left to bottom-right. Then we fill the cells in $P^{\nearrow}$ with the next $k(n-\ell)$ numbers in descending order, filling this sub-grid row by row, top-left to bottom-right, however keeping the numbers in each row ascending; and finally, the cells in $P^{\swarrow}$ with the next $(n-k)\ell$ numbers in ascending order, filling this sub-grid row by row, top-left to bottom-right, however keeping the numbers in each row descending.

We refer the reader to Figure 7 for an example of this algorithm with $n = 9, k = 5, \ell = 3$. The other case in the theorem has a symmetric argument. ◀

20:6 Permutation Match Puzzles

	D	A	D		D	A	D		A	A	A		D	D	D			
A	3	4	9		D	9	4	3		D	3	2	1		D	9	8	7
A	2	5	8		D	8	5	2		A	4	5	6		A	4	5	6
A	1	6	7		D	7	6	1		D	9	8	7		D	3	2	1

■ **Figure 6** Examples of row-uniform and column-uniform puzzles with non-uniform labels DAD in the other direction. All examples show valid solutions constructed according to the method described in the proof.

	A	A	A	D	D	D	D	D	D
A	1	2	3	64	65	66	67	68	69
A	4	5	6	58	59	60	61	62	63
A	7	8	9	52	53	54	55	56	57
A	10	11	12	46	47	48	49	50	51
A	13	14	15	40	41	42	43	44	45
D	72	71	70	39	38	37	36	35	34
D	75	74	73	33	32	31	30	29	28
D	78	77	76	27	26	25	24	23	22
D	81	80	79	21	20	19	18	17	16

■ **Figure 7** A valid solution for a 9×9 grid with $k = 5$ and $\ell = 3$. The grid is partitioned into four subgrids: $P^{\nwarrow}$ (1-15), $P^{\nearrow}$ (40-69), $P^{\swarrow}$ (70-81), and $P^{\searrow}$ (39-16).

► **Corollary 5.** *There are*

$$4 \cdot (2^n - 1) + 2 \cdot (n - 1)^2$$

sorting match puzzles of order n .

Proof. By Theorem 4, solvable puzzles are either: (i) row-uniform puzzles (where $r = A^n$ or $r = D^n$), giving $2 \cdot 2^n$ puzzles (two choices for r , and 2^n choices for c), or (ii) column-uniform puzzles (where $c = A^n$ or $c = D^n$), giving $2 \cdot 2^n$ puzzles (two choices for c , and 2^n choices for r), or (iii) non-uniform (of two types, which we will see later).

This sums up to $4 \cdot 2^n$. However, we need to make a small correction for over-counting: we have to subtract four to account for the fact that the four puzzles $(r = A^n, c = A^n)$, $(r = A^n, c = D^n)$, $(r = D^n, c = A^n)$, $(r = D^n, c = D^n)$ are double-counted as they are both row-uniform and column-uniform. This makes it $4 \cdot 2^n - 4$ uniform puzzles total.

Finally, let us count the non-uniform puzzles. By Theorem 4, we have two types of non-uniform puzzles: (i) $r = A^k D^{n-k}$ and $c = A^\ell D^{n-\ell}$ for some $k, \ell \in \{1, \dots, n-1\}$, or (ii) $r = D^k A^{n-k}$ and $c = D^\ell A^{n-\ell}$ for some $k, \ell \in \{1, \dots, n-1\}$. For each type, there are $(n-1)^2$ solvable puzzles (one puzzle for each choice of k and ℓ). This makes it $2 \cdot (n-1)^2$ non-uniform puzzles total. ◀

The first few terms of the series are 4, 14, 36, 78, 156. This sequence is not present on OEIS, and we plan to put it up soon.

Graph-Theoretic Formulation. We observe that this puzzle can be reformulated as a problem of topologically sorting a directed graph. This view is redundant in the light of the characterization above, but will be useful when we consider the general permutation match puzzles later.

Given an instance of a sorting match puzzle, we can create a vertex for each cell in the grid, and add the following directed edges based on the A/D labels: for each row labeled A (respectively, D): add edges from left to right (respectively, right to left); for each column labeled A (respectively, D): add edges from bottom to top (respectively, top to bottom).

► **Theorem 6.** *An instance of the ascending-descending grid puzzle has a solution if and only if its corresponding directed graph is acyclic.*

Proof. ($\Rightarrow$) If there exists a solution, then the numbers in the grid provide a valid topological ordering of the vertices, which implies the graph must be acyclic.

($\Leftarrow$) If the graph is acyclic, any topological ordering of the vertices provides a valid solution when we replace the vertices with numbers 1 through n^2 in order. ◀

Having established solvability, we now turn to counting solutions for sorting match puzzles that are not uniform.

2.1 Counting Solutions

Consider a $(n \times n)$ sorting match puzzle (r, c) , where there exist integers $k \in \{1, \dots, n-1\}$ and $\ell \in \{1, \dots, n-1\}$ such that:

- The first k rows are labeled A and the remaining rows are labeled D, and
- The first ℓ columns are labeled A and the remaining columns are labeled D.

Recall our notation from before: $P^{\nwarrow}$ for the top-left $k \times \ell$ subgrid, $P^{\nearrow}$ for the top-right $k \times (n - \ell)$ subgrid, $P^{\swarrow}$ for the bottom-left $(n - k) \times \ell$ subgrid, and $P^{\searrow}$ for the bottom-right $(n - k) \times (n - \ell)$ subgrid. Our first claim is that in any valid solution, the numbers from 1 to $k\ell + (n - k)(n - \ell)$ are used only in $P^{\nwarrow}$ and $P^{\searrow}$.

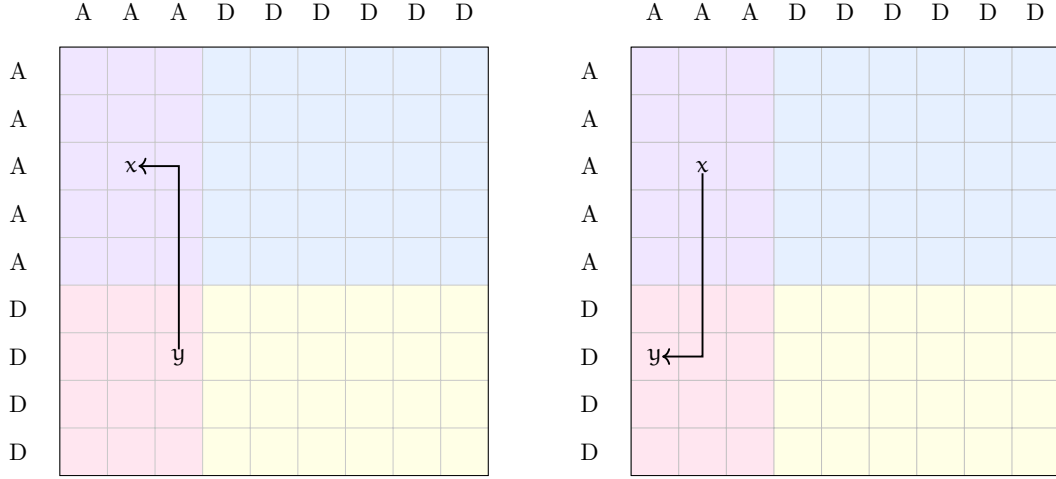
► **Lemma 7.** *In any valid solution to this puzzle, the numbers from 1 to $k\ell + (n - k)(n - \ell)$ must appear only in $P^{\nwarrow}$ and $P^{\searrow}$.*

Proof. We first argue that any number in $P^{\nwarrow} \cup P^{\searrow}$ must be smaller than any number in $P^{\swarrow} \cup P^{\nearrow}$. Then the claim follows, since the entries of the grid are a permutation of 1 to n^2 .

Consider a number x in $P^{\nwarrow} \cup P^{\searrow}$, located in the i^{th} row and j^{th} column; and a number y in $P^{\swarrow} \cup P^{\nearrow}$, located in the p^{th} row and q^{th} column. There are four cases, and we discuss one of them: the others are analogous.

Suppose $x \in P^{\nwarrow}$ and $y \in P^{\swarrow}$: if $p = q$, then we are in an ascending column and $y > x$. If $p < q$, then $x < z < y$, where z is the number in the i^{th} row and q^{th} column. If $p > q$, then $x < y < z$, where z is the number in the p^{th} row and j^{th} column (see Figure 8 for an illustration). ◀

An analogous result holds if the first k rows are labeled D and the remaining rows are labeled A, and the first ℓ columns are labeled D and the remaining columns are labeled A.



■ **Figure 8** Two examples showing paths from y to x in a 9×9 grid with $k = 5$ and $\ell = 3$. Left: L-shaped path in an ascending column and row. Right: L-shaped path in a descending column and row.

► **Theorem 8.** *The number of solutions to the sorting match puzzle with parameters k and ℓ (i.e., first k rows and first ℓ columns labeled A , rest labeled D) is:*

$$\binom{L}{k\ell} \cdot \binom{n^2 - L}{(n-k)\ell} \cdot H(k, \ell) \cdot H(n-k, n-\ell) \cdot H(k, n-\ell) \cdot H(n-k, \ell),$$

where $H(a, b) = \frac{\prod_{i=0}^{a-1} i!}{\prod_{i=0}^{b-1} (b+i)!} \cdot (ab)!$ is the number of standard Young tableaux of rectangular shape $a \times b$ and $L = k\ell + (n-k)(n-\ell)$.

Proof. By the preceding lemma, numbers 1 to $k\ell + (n-k)(n-\ell)$ fill $P^{\nwarrow} \cup P^{\searrow}$, while the remaining numbers fill $P^{\nearrow} \cup P^{\swarrow}$. Within each subgrid, the ordering constraints (ascending/-descending in rows and columns) make filling equivalent to constructing a standard Young tableau of rectangular shape. The binomial coefficient counts ways to distribute the smallest $k\ell + (n-k)(n-\ell)$ numbers between $P^{\nwarrow}$ and $P^{\searrow}$. Once this choice is made, the hook length formula gives the count for each rectangular subgrid independently. ◀

2.2 Unique Solutions

Note that Theorem 4 characterizes solvability, and gives us conditions for when a permutation grid puzzle has *no* solution. We would now like to characterize when a permutation grid puzzle has *exactly one* solution. Notice that non-uniform grids have multiple solutions as suggested by our counting argument from Theorem 8: also see Figure 9 for examples on puzzles of order two.

On the other hand, if we fix a row-uniform or column-uniform puzzle, we can characterize when it has a unique solution: this happens precisely when the non-uniform side is alternating, and is filled out by a so-called “boustrophedon” pattern (see Figure 10).

► **Theorem 9.** *A sorting match puzzle of order $n \geq 2$ has a unique solution if and only if it satisfies one of the following:*

1. Row-uniform with alternating columns: $r \in \{A^n, D^n\}$ and c alternates between A and D .
2. Column-uniform with alternating rows: $c \in \{A^n, D^n\}$ and r alternates between A and D .

	A	D		A	D		D	A		D	A
A	1	3	A	1	4	D	4	1	D	4	2
D	4	2	D	3	2	A	2	3	A	1	3
	Solution 1			Solution 2			Solution 1			Solution 2	

■ **Figure 9** Non-uniform puzzles with multiple solutions. Left pair: the AD/AD puzzle has two valid solutions. Right pair: the DA/DA puzzle also has two valid solutions. In both cases, the non-uniform structure in rows and columns creates incomparable pairs in the poset.

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

■ **Figure 10** The boustrophedon (snake) pattern for a 5×5 puzzle with $r = A^5$ and $c = ADADA$. The alternating column constraints force a unique traversal order, shown by the arrows.

Proof. We first show that an alternating constraint pattern on an uniform puzzle has a unique solution. Consider the case $r = A^n$ with $c = (AD)^{n/2}$ (or $(AD)^{(n-1)/2}A$ if n is odd). Every row is ascending left-to-right. Column 1 is ascending (bottom-to-top), column 2 is descending, and so on. This creates a snake-like path: starting from cell $(1, 1)$, we traverse up column 1, then across row n to column 2, down column 2, across row 1 to column 3, and so forth. We now argue that each step in this path is *forced* by the constraints, making the solution unique.

We show by induction that value k is at cell $(k, 1)$ for each $k \in \{1, \dots, n\}$. Since all rows are ascending, the value 1 cannot appear in any column other than the first. Since column 1 is ascending, the value 1 must be in row 1. Thus, cell $(1, 1)$ contains 1, this is the base case. For the inductive step, suppose values $1, \dots, k$ occupy cells $(1, 1), \dots, (k, 1)$ for some $k < n$. We show that value $k+1$ must be at $(k+1, 1)$.

Suppose $k+1$ is at cell (i, j) with $j \geq 2$. Since row i is ascending, $(i, 1) < (i, j) = k+1$, so $(i, 1) \leq k$. By the inductive hypothesis, the cells containing values $1, \dots, k$ are exactly $(1, 1), \dots, (k, 1)$, hence $i \leq k$. Now consider column 2, which is descending. All cells $(i', 2)$ with $i' > i$ satisfy $(i', 2) < (i, 2) \leq k+1$ (the first inequality by column 2 descending, the second by row i ascending). Thus all such cells need values from $\{1, \dots, k\}$, but these values are already in column 1. Since there are $n-i \geq n-k \geq 1$ such cells, we have a contradiction.

Therefore $k+1$ is in column 1. Since cells $(1, 1), \dots, (k, 1)$ are occupied and column 1 is ascending, we have $(k+1, 1) = k+1$.

After column 1 is filled with $1, \dots, n$, we show $n+1$ must be at $(n, 2)$. If $n+1$ were at (i, j) with $j \geq 3$, then $(i, 2) < (i, j) = n+1$ by row i ascending. But all values less than $n+1$ are in column 1, so $(i, 2)$ cannot be filled. Thus $n+1$ must be in column 2. A similar

20:10 Permutation Match Puzzles

inductive argument shows that values $n + 1, \dots, 2n$ occupy cells $(n, 2), (n - 1, 2), \dots, (1, 2)$ respectively: the descending constraint on column 2 forces us to fill it from bottom to top. The pattern alternates: odd columns fill upward, even columns fill downward, creating the boustrophedon pattern. The case $r = D^n$ with alternating columns, and the symmetric cases with c uniform and r alternating, follow by analogous arguments.

Now suppose $r = A^n$ (row-uniform) but c contains two adjacent columns with the same label, say columns j and $j + 1$ are both descending. Consider the 2×2 subgrid formed by rows $i, i + 1$ and columns $j, j + 1$. Let the four cells be:

w	x
y	z

where $w = (i + 1, j)$, $x = (i + 1, j + 1)$, $y = (i, j)$, $z = (i, j + 1)$.

The constraints require $y < z$ and $w < x$ and $y < w$ and $z < x$. These constraints form a partial order where y is minimum and x is maximum, but w and z are *incomparable* – neither is constrained to be less than the other. This means there exist valid solutions with $w < z$ and other valid solutions with $z < w$, giving at least two distinct solutions. As a concrete example, consider a 4×4 puzzle with $r = AAAA$ and $c = DDAD$. The following are both valid solutions:

	D	D	A	D
A	4	8	9	16
A	3	7	10	15
A	2	6	11	14
A	1	5	12	13

	D	D	A	D
A	7	8	9	16
A	5	6	10	15
A	3	4	11	14
A	1	2	12	13

For the mixed case (where neither r nor c is uniform), the hook length formula from the previous section shows the count is always greater than 1. ◀

2.3 Nearest Solvable Puzzles

Imagine Tanvi encounters an unsolvable sorting match puzzle. Unwilling to give up, she plans to change some of the row and column labels to make the puzzle solvable. However, she does not want to be conspicuous in doing so. Hence she wonders: what is the minimum number of label flips needed to make the puzzle solvable?

Given an arbitrary sorting match puzzle $(r, c) \in \{A, D\}^n \times \{A, D\}^n$, we may ask: what is the “nearest” solvable puzzle? Using the notation $\|a, b\|$ to denote the Hamming distance between two strings, we define the *nearest solvable puzzle problem*:

NEAREST SOLVABLE SORTING MATCH PUZZLE

Input: A sorting match puzzle $(r, c) \in \{A, D\}^n \times \{A, D\}^n$

Output: A solvable puzzle (r^*, c^*) minimizing $\|r, r^*\| + \|c, c^*\|$

Recall from Theorem 4 that a puzzle is solvable if and only if it avoids certain forbidden 2×2 patterns. A key observation is that if either r or c is *uniform* (all A ’s or all D ’s), then no forbidden pattern can arise, since such patterns require transitions in *both* rows and columns at adjacent positions. More generally, a puzzle is solvable if both r and c have at most one transition point (from A to D or vice versa), provided these transitions are *compatible* – both must be $A \rightarrow D$ transitions or both must be $D \rightarrow A$ transitions.

Notice that by guessing the “shape” of the nearest solvable puzzle, we can find the answer in $O(n^2)$ time by brute force. But we can do better. For a string $s \in \{A, D\}^n$, define:

- $X(k) = |\{i \leq k : s_i = A\}|$, the number of A’s in the first k positions
- $\text{total}_A(s) = X(n)$, the total number of A’s in s

The cost to transform s into the pattern $A^k D^{n-k}$ (first k positions are A, remaining are D) is:

$$\text{cost}_{A \rightarrow D}(k) = \underbrace{(k - X(k))}_{\# \text{ D's to flip in prefix}} + \underbrace{(\text{total}_A - X(k))}_{\# \text{ A's to flip in suffix}} = k + \text{total}_A - 2 \cdot X(k)$$

Similarly, the cost to transform s into $D^k A^{n-k}$ is:

$$\text{cost}_{D \rightarrow A}(k) = X(k) + (n - k - \text{total}_A + X(k)) = 2 \cdot X(k) + n - k - \text{total}_A$$

We compute $X(k)$ incrementally via the recurrence $X(0) = 0$ and $X(k) = X(k-1) + [s_k = A]$. This yields recurrences for the costs:

$$\begin{aligned} \text{cost}_{A \rightarrow D}(0) &= \text{total}_A, & \text{cost}_{A \rightarrow D}(k) &= \text{cost}_{A \rightarrow D}(k-1) + \begin{cases} -1 & \text{if } s_k = A \\ +1 & \text{if } s_k = D \end{cases} \\ \text{cost}_{D \rightarrow A}(0) &= n - \text{total}_A, & \text{cost}_{D \rightarrow A}(k) &= \text{cost}_{D \rightarrow A}(k-1) + \begin{cases} +1 & \text{if } s_k = A \\ -1 & \text{if } s_k = D \end{cases} \end{aligned}$$

As we scan from $k = 0$ to n , the cost changes by ± 1 at each step, so we can track the running minimum in $O(n)$ time for each string.

► **Theorem 10.** *The nearest solvable sorting match puzzle can be computed in $O(n)$ time.*

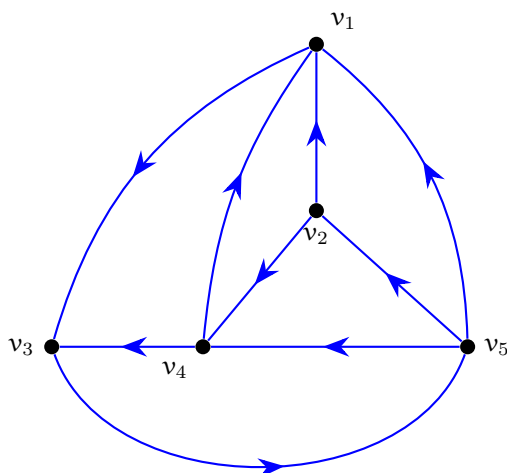
Proof. Let $\min_{A \rightarrow D}(s)$ and $\min_{D \rightarrow A}(s)$ denote the minimum costs over all k for the respective patterns. The answer is:

$$\min \begin{pmatrix} \min(\text{total}_A(r), n - \text{total}_A(r)), \\ \min(\text{total}_A(c), n - \text{total}_A(c)), \\ \min_{A \rightarrow D}(r) + \min_{A \rightarrow D}(c), \\ \min_{D \rightarrow A}(r) + \min_{D \rightarrow A}(c) \end{pmatrix}.$$

The first two options make one string uniform (cost = minimum of flipping all to A or all to D), leaving the other unchanged. The last two options transform both strings to have at most one compatible transition. Each quantity is computed in a single $O(n)$ pass. ◀

3 Permutation Match Puzzles

In this section, we turn to permutation match puzzles and establish that determining the minimum number of constraint modifications needed to make such a puzzle solvable is NP-complete. Recall that a permutation match puzzle is solvable if and only if its associated constraint graph is acyclic. This motivates the following graph-theoretic problem.



■ **Figure 11** A directed graph H on five vertices v_1, v_2, v_3, v_4, v_5 . (Note that all cycles of H pass through v_3 and v_5 ; hence, deleting any one of these vertices makes the graph acyclic.) The grid graph G corresponding to H is shown in Figure 12.

GRID ACYCLIFICATION

Input: A positive integer n , a non-negative integer k , and a directed graph G on the vertex set $V = \{(i, j) \mid 1 \leq i, j \leq n\}$ such that for each $i \in [n]$, the subgraph induced by the row $R_i := \{(i, j) \mid 1 \leq j \leq n\}$ is an acyclic tournament, and for each $j \in [n]$, the subgraph induced by the column $C_j := \{(i, j) \mid 1 \leq i \leq n\}$ is an acyclic tournament.

Question: Can G be made acyclic by deleting at most k vertices?

Deleting a vertex removes it along with all incident edges. The constraint that each row and column induces an acyclic tournament reflects the structure arising from permutation constraints: each permutation $\pi \in S_n$ defines a total order on n elements, which corresponds to a transitive tournament on the row or column.

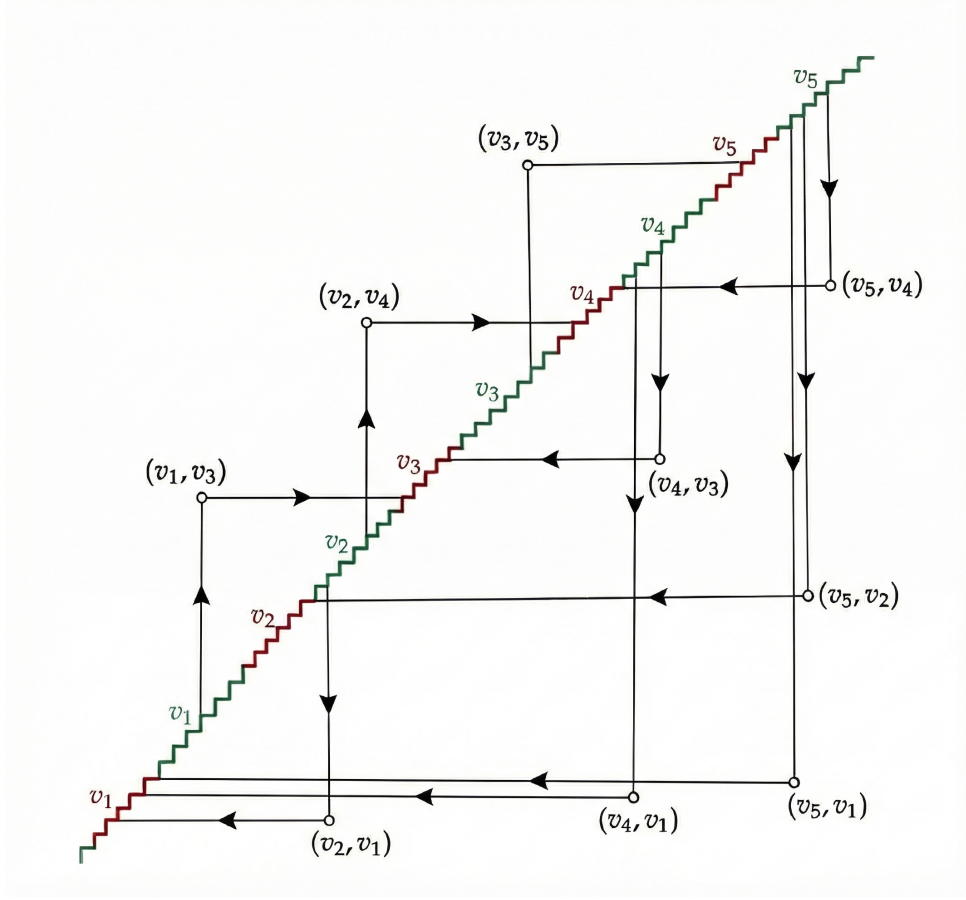
The GRID ACYCLIFICATION problem is closely related to the classical FEEDBACK VERTEX SET problem, where one seeks to remove a minimum number of vertices to eliminate all directed cycles. Our variant restricts attention to graphs with the grid structure described above.

► **Theorem 11.** *GRID ACYCLIFICATION is NP-complete.*

We establish this by a reduction from FEEDBACK VERTEX SET on directed graphs. Before presenting the formal construction, we illustrate the key ideas through an example.

Example. Consider the directed graph H on five vertices shown in Figure 11. The graph contains several cycles, including $v_2 \rightarrow v_4 \rightarrow v_3 \rightarrow v_5 \rightarrow v_2$ and $v_1 \rightarrow v_3 \rightarrow v_5 \rightarrow v_1$. The minimum feedback vertex set of H has size one, since deleting one vertex (v_3 or v_5) breaks all cycles.

We construct a grid graph G from H as follows. The construction has three types of vertices (see Figure 12):



■ **Figure 12** The grid graph G constructed from the directed graph H of Figure 11. Each vertex v_i contributes a red segment (incoming), a black key step κ_i , and a green segment (outgoing) to the staircase. Edge vertices (the circular dots lying outside the staircase) connect green segments to red segments via orthogonal paths. All other vertices of the grid are inactive (that is, they do not participate in any cycle). (Note that deleting κ_3 or κ_5 makes this grid acyclic.).

1. **Staircase vertices:** A diagonal staircase runs from the bottom-left to the top-right corner (edges of the staircase are oriented alternately upwards and rightwards). For each vertex v_i of H , the staircase contains a *red segment* (where edges can only enter), followed by a *connector vertex* (shown in black), followed by a *green segment* (where edges can only leave). The connector vertex between the red and green segments of v_i is called the *key vertex* for v_i .
2. **Edge vertices:** For each edge (v_i, v_j) in H , we place an *edge vertex* at a unique grid position.
3. **Inactive vertices:** All remaining grid cells contain *inactive vertices*. These are configured as sinks in every row and column tournament, ensuring they never participate in any cycle.

The key insight is that **each row and column contains at most three active (non-sink) vertices**: at most two from the staircase (where the row/column crosses the diagonal) and at most one edge vertex. This sparse structure allows us to carefully control the edge orientations within each row and column tournament.

A cycle $v_1 \rightarrow v_3 \rightarrow v_5 \rightarrow v_1$ in H corresponds to a cycle in G that traverses:

green $v_1 \rightarrow$ edge vertex $(v_1, v_3) \rightarrow$ red $v_3 \rightarrow$ (upwards along staircase) $\rightarrow$ green $v_3 \rightarrow$ edge vertex $(v_3, v_5) \rightarrow$ red $v_5 \rightarrow$ (upwards along staircase) $\rightarrow$ green $v_5 \rightarrow$ edge vertex $(v_5, v_1) \rightarrow$ red $v_1 \rightarrow$ (upwards along staircase) $\rightarrow$ green v_1 .

Crucially, deleting the *key vertex* for v_i (the black connector between red v_i and green v_i) disconnects all paths through v_i , which corresponds to removing v_i from the feedback vertex set in H .

Formal Construction. Let $H = (V_H, E_H)$ be a directed graph with $V_H = \{v_1, \dots, v_n\}$ and $m = |E_H|$ edges. We construct a grid graph G of size $N \times N$ where $N = 4n + m$.

Staircase structure. The diagonal of G contains a staircase pattern. For each vertex $v_i \in V_H$, we allocate a segment of the diagonal consisting of:

- A *red segment* of $d^-(v_i) + 1$ consecutive diagonal cells, where $d^-(v_i)$ is the in-degree of v_i in H .
- A single *key vertex* κ_i .
- A *green segment* of $d^+(v_i) + 1$ consecutive diagonal cells, where $d^+(v_i)$ is the out-degree of v_i .

The segments are arranged along the diagonal in order: red v_1 , key κ_1 , green v_1 , red v_2 , key κ_2 , green v_2 , and so on.

Edge vertices. For each edge $e = (v_i, v_j) \in E_H$, we designate a unique cell ε_e in the grid. This cell is positioned such that:

- It shares a row with exactly one cell from the green segment of v_i .
- It shares a column with exactly one cell from the red segment of v_j .

Since each vertex's segment has enough cells to accommodate all incident edges, and each edge uses a distinct cell from each segment, the edge vertices are all distinct and well-defined.

All grid cells not designated as staircase vertices or edge vertices are *inactive vertices*.

Tournament orientations. We now specify the edge orientations within each row and column. Let $\mathcal{A}$ denote the set of *active vertices*: all staircase vertices and edge vertices. For each row R_i and column C_j :

- **Inactive vertices are sinks:** For every inactive vertex u and every other vertex w in the same row or column, the edge is directed $w \rightarrow u$.
- **Staircase edges follow the diagonal:** Within the staircase, edges are directed from lower-left to upper-right: if (a, a) and (b, b) are staircase vertices with $a < b$ in the same row or column, then $(a, a) \rightarrow (b, b)$.
- **Edge vertex orientations:** For an edge vertex ε_e corresponding to edge (v_i, v_j) :
 - In its row (shared with green v_i): the edge goes from the green segment cell to ε_e .
 - In its column (shared with red v_j): the edge goes from ε_e to the red segment cell.
- **Direct edges between distant active vertices:** If two active vertices share a row or column but are not adjacent in the staircase and neither is an edge vertex connecting them, orient the edge from left-to-right (in rows) or bottom-to-top (in columns).

These orientations ensure that each row and column induces an acyclic tournament on its vertices. We now claim that inactive vertices never participate in any cycle of G . Indeed, every inactive vertex is a sink with respect to vertices that are staircase and edge vertices. Further, the subgraph induced by inactive vertices is acyclic by construction. This implies the claim.

► **Lemma 12.** *The graph G contains a cycle if and only if H contains a cycle. Moreover, every cycle in G passes through the key vertex κ_i for some $v_i \in V_H$.*

Proof. ($\Rightarrow$) Let C be a cycle in G . Note that C contains only active vertices. Since staircase edges are directed from lower-left to upper-right, any path along the staircase must eventually leave the staircase (at a green segment) to continue as a cycle. Such a path must use an edge vertex to jump to a later portion of the staircase.

Tracing the cycle C : starting from any key vertex κ_i , the path proceeds through the green segment of v_i , exits via some edge vertex $\varepsilon_{(v_i, v_j)}$, enters the red segment of v_j , proceeds to κ_j , and continues. The sequence of key vertices visited corresponds exactly to a cycle in H .

($\Leftarrow$) Let $v_{i_1} \rightarrow v_{i_2} \rightarrow \dots \rightarrow v_{i_\ell} \rightarrow v_{i_1}$ be a cycle in H . We construct a corresponding cycle in G : from κ_{i_1} , follow the staircase through green v_{i_1} , take the edge vertex $\varepsilon_{(v_{i_1}, v_{i_2})}$, enter red v_{i_2} , proceed to κ_{i_2} , and continue. This traces a valid directed path that returns to κ_{i_1} . $\blacktriangleleft$

Proof of Theorem 11. Membership in NP is clear: given a set S of at most k vertices, we can verify in polynomial time that $G - S$ is acyclic.

For NP-hardness, we reduce from FEEDBACK VERTEX SET on directed graphs, which is NP-complete. Given an instance (H, k) of FEEDBACK VERTEX SET, construct the grid graph G as described above. We claim that H has a feedback vertex set of size at most k if and only if G can be made acyclic by deleting at most k vertices.

($\Rightarrow$) Suppose $S \subseteq V_H$ is a feedback vertex set of H with $|S| \leq k$. Let $S' = \{\kappa_i : v_i \in S\}$ be the corresponding key vertices in G . We claim $G - S'$ is acyclic. By Lemma 12, every cycle in G passes through some key vertex κ_i . If $v_i \in S$, then $\kappa_i \in S'$ is deleted. If $v_i \notin S$, then any cycle through κ_i corresponds to a cycle in H through v_i , but since S is a feedback vertex set, such a cycle must also pass through some $v_j \in S$, meaning the cycle in G passes through $\kappa_j \in S'$. Thus, all cycles are broken.

($\Leftarrow$) Suppose S' is a set of at most k vertices in G such that $G - S'$ is acyclic. We construct a feedback vertex set S for H with $|S| \leq k$ using a *pushing argument*.

First, recall that we may assume S' contains no inactive vertices (removing an inactive vertex from S' cannot create new cycles).

For each vertex $u \in S'$, we “push” it to a nearby key vertex:

- If $u = \kappa_i$ is already a key vertex, include v_i in S .
- If u is in the red segment of v_i , include v_i in S . (Any cycle through u must continue to κ_i , so deleting κ_i also breaks this cycle.)
- If u is in the green segment of v_i , include v_i in S . (Any cycle through u must have come from κ_i .)
- If $u = \varepsilon_{(v_i, v_j)}$ is an edge vertex, include v_i in S . (Any cycle through u must pass through either κ_i or κ_j ; we choose v_i .)

This maps each vertex in S' to a vertex in S (possibly with repetitions), so $|S| \leq |S'| \leq k$. We claim S is a feedback vertex set. Suppose for contradiction that $H - S$ contains a cycle $v_{i_1} \rightarrow \dots \rightarrow v_{i_\ell} \rightarrow v_{i_1}$. By Lemma 12, G contains a corresponding cycle through $\kappa_{i_1}, \dots, \kappa_{i_\ell}$. Since $v_{i_j} \notin S$ for all j , no key vertex κ_{i_j} is the image of any vertex in S' under our pushing map. But then the original set S' did not break this cycle in G , contradicting that $G - S'$ is acyclic.

The construction of G from H is polynomial in $|V_H| + |E_H|$, completing the reduction. $\blacktriangleleft$

4 Concluding Remarks

We have presented a comprehensive study of permutation match puzzles, a family of constraint satisfaction problems on grids inspired by a physical puzzle. Our main contributions include a complete characterization of solvability via the acyclicity of an associated constraint graph,

which reduces to an elegant “at most one switch” condition on the row and column labels. When solutions exist, we derived an exact counting formula using hook lengths, connecting these puzzles to the rich combinatorics of standard Young tableaux. For unsolvable instances, we provided a linear-time algorithm to compute the minimum number of label flips required to restore solvability. These results extend naturally to three-dimensional grids and partially filled puzzles, where the solvability criterion generalizes in a clean manner.

Several directions remain open for further investigation. First, while we established NP-completeness for the generalized setting where rows and columns specify arbitrary permutations, it would be interesting to identify tractable special cases – for instance, when the permutations come from a restricted family or when the constraint graph has bounded treewidth. Second, our results focus on complete grids; understanding the complexity of partial filling problems where only a subset of cells must be filled could lead to connections with constraint propagation and arc consistency. Third, the hook length formula suggests deeper algebraic structure; exploring whether representation-theoretic techniques can yield further insights into the solution space would be worthwhile. The interplay between simple local constraints and global structure makes permutation match puzzles a fertile ground for both recreational mathematics and algorithmic exploration.

References

- 1 Nir Ailon, Moses Charikar, and Alanthia Newman. Aggregating inconsistent information: Ranking and clustering. In *Proceedings of the 37th Annual ACM Symposium on Theory of Computing (STOC)*, pages 684–693, 2005. doi:10.1145/1060590.1060692.
- 2 Michael H. Albert, M. D. Atkinson, Mathilde Bouvel, Nik Ruškuc, and Vincent Vatter. Geometric grid classes of permutations. *Transactions of the American Mathematical Society*, 365(11):5859–5881, 2013. doi:10.1090/S0002-9947-2013-05804-7.
- 3 Graham Brightwell and Peter Winkler. Counting linear extensions. *Order*, 8(3):225–242, 1991. doi:10.1007/BF00383444.
- 4 Pierre Charbit, Stéphan Thomassé, and Anders Yeo. The minimum feedback arc set problem is NP-hard for tournaments. *Combinatorics, Probability and Computing*, 16(1):1–4, 2007. doi:10.1017/S0963548306007887.
- 5 Charles J. Colbourn and Jeffrey H. Dinitz. *Handbook of Combinatorial Designs*. Chapman and Hall/CRC, 2nd edition, 2007. doi:10.1201/9781420010541.
- 6 J. S. Frame, G. de B. Robinson, and R. M. Thrall. The hook graphs of the symmetric group. *Canadian Journal of Mathematics*, 6:316–324, 1954. doi:10.4153/CJM-1954-030-1.
- 7 Curtis Greene, Albert Nijenhuis, and Herbert S. Wilf. A probabilistic proof of a formula for the number of Young tableaux of a given shape. *Advances in Mathematics*, 31(1):104–109, 1979. doi:10.1016/0001-8708(79)90023-9.
- 8 Wm. Woolsey Johnson and William E. Story. Notes on the “15” puzzle. *American Journal of Mathematics*, 2(4):397–404, 1879.
- 9 Daniel Ratner and Manfred Warmuth. The $(n^2 - 1)$ -puzzle and related relocation problems. *Journal of Symbolic Computation*, 10(2):111–137, 1990. doi:10.1016/S0747-7171(08)80001-6.
- 10 Takayuki Yato and Takahiro Seta. Complexity and completeness of finding another solution and its application to puzzles. In *Proceedings of the 14th International Symposium on Algorithms and Computation (ISAAC)*, pages 307–316, 2003. doi:10.1007/978-3-540-24587-2_33.