A simple proof that the $(n^2 - 1)$-puzzle is hard

Erik D. Demaine[a,*], Mikhail Rudoy[a,1,**]

[a]*MIT Computer Science and Artificial Intelligence Laboratory, 32 Vassar St., Cambridge, MA 02139, USA*

Abstract

The 15 puzzle is a classic reconfiguration puzzle with fifteen uniquely labeled unit squares within a 4×4 board in which the goal is to slide the squares (without ever overlapping) into a target configuration. By generalizing the puzzle to an $n \times n$ board with $n^2 - 1$ squares, we can study the computational complexity of problems related to the puzzle; in particular, we consider the problem of determining whether a given end configuration can be reached from a given start configuration via at most a given number of moves. This problem was shown NP-complete in [1]. We provide an alternative simpler proof of this fact by reduction from the rectilinear Steiner tree problem.

Keywords: $(n^2 - 1)$-puzzle, complexity of games, NP-hardness

1. Introduction

The 15 puzzle is a classic puzzle consisting of fifteen sliding 1×1 squares labeled with the numbers 1 through 15 within a 4×4 board. The remaining spot is taken up by a hole, whose presence allows the other squares to slide. The goal of the puzzle is to slide the squares in such a way that they end up arranged in order, while avoiding collisions. The number of moves required to solve this puzzle is at most 80, as realized by six “hardest” starting configurations [2].

We can naturally generalize the 15 puzzle to the $(n^2 - 1)$-puzzle, where $n^2 - 1$ sliding squares labeled 1 through $n^2 - 1$ and one hole fill an $n \times n$ board of cells. Given a pair of configurations (injective mappings from squares to cells), we can decide whether one can be reached from the other by any sequence of moves in polynomial time by checking a simple parity condition (see for example [3]). A further generalization is to consider a graph where $N - 1$ out of the N vertices have tokens labeled 1 through $N - 1$, and we want to permute the tokens by repeatedly moving a token along an edge onto the empty vertex. (In addition to the $n^2 - 1$ puzzle, this generalization captures a variety of real-world puzzles with sliding pieces in tracks.) Such puzzles can also be solved in polynomial time by characterizing their group structure [4, p. 866]. A further generalization is to have k tokens labeled 1 through k for any $k < N$. This generalization can also be solved in polynomial time, and the worst-case number of moves is $\Theta(N^3)$ [5]. A related problem is *token swapping* where any pair of tokens can be swapped, not just a token with an empty “token”, which makes any puzzle solvable and using just $O(N^2)$ swaps [6]. See [7, 8] for other related puzzles, including some restrictions on the moves that make solving a puzzle PSPACE-complete.

But what if we want to minimize the number of moves to solve a given puzzle configuration? In this paper, we consider this problem for the basic $n^2 - 1$ puzzle, which we can phrase as the following decision problem bounding the number of moves:

Problem 1. *The* (n^2-1)-puzzle problem *asks, for a given pair of puzzle configurations s and t and a given number k, whether it is possible to convert configuration s into configuration t using at most k moves.*

[*] Corresponding author

[**] Principal corresponding author

Email addresses: edemaine@mit.edu (Erik D. Demaine), mrudoy@gmail.com (Mikhail Rudoy)

[1] Now at Google.

This problem was shown NP-complete in [1] by reduction from a rather complicated variant of SAT introduced in that paper. The purpose of this paper is to provide a simpler proof of the same fact. In particular, we provide a reduction from the rectilinear Steiner tree problem to the $(n^2 - 1)$-puzzle problem.

Problem 2. *The* rectilinear Steiner tree problem *asks, for a given number l and a given set of points with positive integer coordinates in the plane, whether it is possible to find a tree in the plane whose total length does not exceed l, whose edges are all rectilinear (i.e., horizontal or vertical), and which passes through each of the given points.*

The rectilinear Steiner tree problem was shown strongly NP-hard in [9], so a reduction from this problem to the $(n^2 - 1)$-puzzle problem is sufficient for our desired result.

2. Reduction

Suppose we are given a rectilinear Steiner tree instance consisting of a set of points $P = (p_1, p_2, \ldots, p_{|P|})$ and a value l. We wish to systematically convert the pair (P, l) into a $(n^2 - 1)$-puzzle instance consisting of a pair of puzzle configurations s and t and a value k.

Let $c = 18 \cdot |P|$, and let m be the maximum coordinate of any point in P. Then the puzzle configurations s and t that we produce will be $n \times n$ puzzles with $n = (m + 1) \cdot c$.

Now we define a correspondence between the points in P and some of the cells of the puzzle board. In particular, we correspond point (x, y) with the puzzle board cell at location (cx, cy). Notice that this location is within the board because, by design, cx and cy are both bounded above by $m \cdot c$.

With this correspondence in place, we can define configurations s and t. The hole in s is placed in the location corresponding to point p_1. We number every other cell in s arbitrarily. To construct t, we modify configuration s. For each point $p \in P \setminus \{p_1\}$, permute the three squares in the board locations directly above, directly to the right, and at the puzzle board cell corresponding to p in a clockwise cycle. This is shown in Figure 1. The result of doing this operation to s for every $p \in P \setminus \{p_1\}$ is the puzzle configuration t.

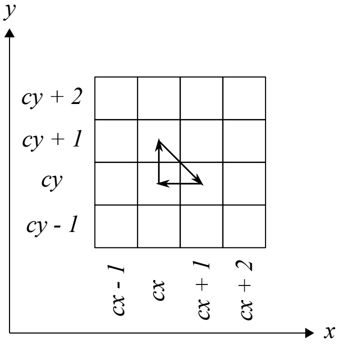

Figure 1: Configuration t is the same as configuration s, except near cells with coordinates $(cx, cy) \in P \setminus \{p_1\}$. For each such pair in $P \setminus \{p_1\}$, three squares in t are cycled relative to their positions in s as shown here.

Finally, define the move bound $k = (2l + 1) \cdot c$ to conclude the reduction.

3. Polynomial time

Notice that the board size is polynomial in the maximum coordinate value m and the number of points $|P|$. Because the rectilinear Steiner tree problem is strongly NP-hard, we can assume that both m and $|P|$—and therefore also the board size n—are polynomial in the size of the input.

Since the contents of every board cell in s or t can be computed easily, and since the number of board cells is polynomial in the size of the input, both s and t can be computed from (P, l) in polynomial time. Furthermore, k can be computed easily in polynomial time, so the whole reduction runs in polynomial time.

4. Rectilinear Steiner tree instance solution → $(n^2 - 1)$-puzzle instance solution

Suppose that there is a solution to the input rectilinear Steiner tree instance. Hanan's Lemma [10, Theorem 4] states that there exists an optimal-length rectilinear Steiner tree through the points in P whose edges all follow the grid formed by a horizontal and a vertical line through each point in P. In other words, there exists a rectilinear Steiner tree T with unit-length edges over at most $l + 1$ vertices with coordinates in $\{1, \ldots, m\}$ which touches each point in P.

We will use this tree to construct a sequence S' of moves solving the $(n^2 - 1)$-puzzle instance. We begin by building a simpler sequence of moves S. Construct an Euler tour of tree T rooted at p_1 (i.e., a traversal of the edges of T starting and ending at p_1 which passes through each edge exactly once in each direction). For example, one possible traversal for the tree in Figure 2 is

$$p_1, p_3, v_1, p_4, v_1, p_3, p_1, p_2, p_1, v_2, v_3, p_7, v_3, v_4, v_5, p_6, v_6, p_5, v_6, p_6, v_5, v_4, v_3, v_2, p_1.$$

For each edge in the Euler tour, add c moves to S which move the hole in correspondence to the direction of the traversal: if the traversal moves from vertex (x, y) to vertex $(x \pm 1, y)$, then move the hole directly right/left from cell (cx, cy) to cell $(c(x \pm 1), cy)$, and if the traversal moves from vertex (x, y) to vertex $(x, y \pm 1)$, then move the hole directly up/down from cell (cx, cy) to cell $(cx, c(y \pm 1))$. Notice that the hole starts at cell (cx, cy) where $(x, y) = p_1$ and that the traversal starts at p_1. Thus this definition of the moves in S is well defined.

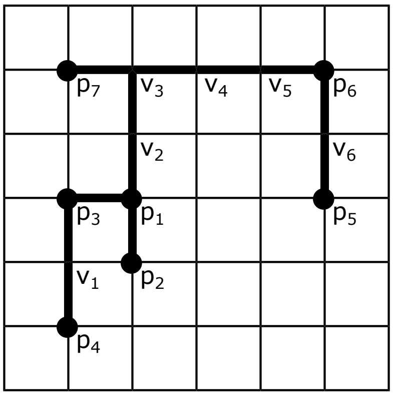

Figure 2: An example rectilinear Steiner tree for the points $p_1, p_2, \ldots, p_7$ which also has additional integer coordinate vertices $v_1, \ldots, v_6$. Every edge of this tree has length 1.

What is the effect of move sequence S? Just as the traversal passes each edge twice (once in each direction), the effect of the move sequence is that the hole passes each horizontal/vertical path of cells twice (once in each direction). The second time the hole passes some path of cells it exactly undoes the effect of the first pass. Thus the overall effect of S is the identity transformation. Below, we modify S into the $(n^2 - 1)$-puzzle instance solution S' by adding moves.

Consider any point $p_i = (x, y) \in P \setminus \{p_1\}$. For any such point, we can consider the 4×4 window of cells with x coordinate in the set $\{cx - 1, cx, cx + 1, cx + 2\}$ and y coordinate in the set $\{cy - 1, cy, cy + 1, cy + 2\}$. Call this window w_i. Under move sequence S, the hole visits window w_i but does not end inside it. Therefore there is some move in S which moves the hole out of window w_i for the final time. We modify S by adding an extra sequence of moves E_i immediately before this move. The overall effect of move sequence E_i will be to cycle the squares in cells (cx, cy), $(cx + 1, cy)$, and $(cx, cy + 1)$ with $(x, y) = p_i$ and to not move any other squares (or the hole). Thus, when resuming S after E_i, the behavior of S will be the same as before (except that those 3 squares will have been permute). Because the effect of S on its own is the identity transformation, the overall effect of the modified move sequence S' is to cycle the squares in cells (cx, cy), $(cx + 1, cy)$, and $(cx, cy + 1)$ for each $(x, y) = p_i \in P \setminus \{p_1\}$. In other words, S' will accomplish exactly the transformation needed to transform configuration s into configuration t.

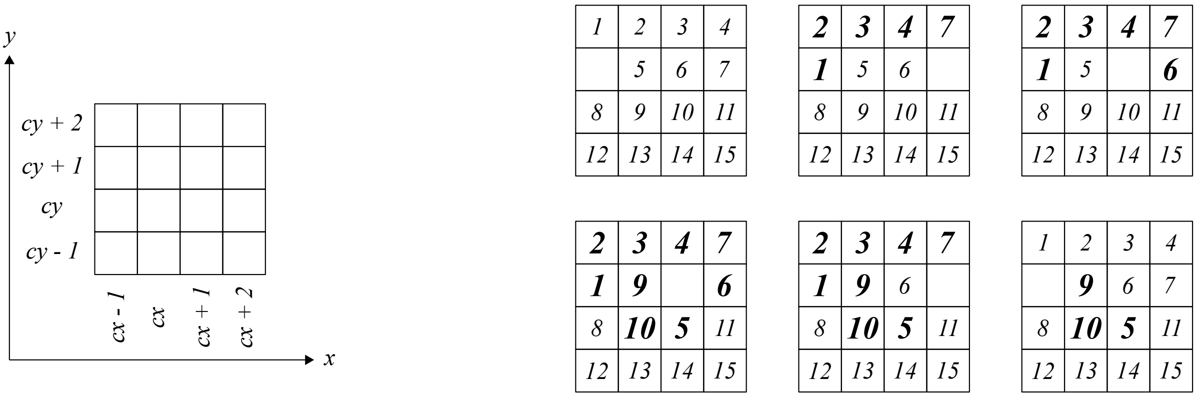

Figure 3: An example move sequence E_i within 4×4 window w_i (left) using numbers 1 through 15 to represent the labels of the squares in the window. In each board (right), the bold labels indicate squares that are displaced from their original position in the first board. In the first board, we show an example of what the window might look like immediately before E_i in the case that the hole starts in cell $(cx-1, cy+1)$. In the second board, we show the window after the hole has moved to cell $(cx+2, cy+1)$ along the boundary of the board. In the third board, we show the window after one more step (with the hole moving to $(cx+1, cy+1)$). The fourth board shows the window after the left, down, right, up movement of the hole. The fifth board shows the window after the hole moves back to cell $(cx+2, cy+1)$. The final board shows the window after the hole has moved back to its initial position. This is the final state of the window after all of E_i.

Immediately before E_i is the last moment when the hole is in window w_i. Thus, the hole must be somewhere among the 12 boundary cells of the window; see Figure 3 for an example. We begin E_i by moving the hole along this 12 cell boundary until it is in cell $(cx+2, cy+1)$. This takes at most 6 moves. Next, we move the hole left one cell into position $(cx+1, cy+1)$. Next, the hole moves in a square pattern: left, down, right, and up. This puts the hole back in the exact same position, but cycles the three squares in the positions it passes through. In particular, the squares in cells (cx, cy), $(cx+1, cy)$, and $(cx, cy+1)$ are cycled clockwise. After that, we undo the initial part of E_i: the hole moves right to $(cx+2, cy+1)$ and then retraces its steps to wherever it started before E_i. Note that the only part of E_i affecting the squares in cells (cx, cy), $(cx+1, cy)$, and $(cx, cy+1)$ is the left-down-right-up motion of the hole. Thus, the overall effect of E_i is to cycle the squares in cells (cx, cy), $(cx+1, cy)$, and $(cx, cy+1)$ clockwise and to leave the other squares unaffected. In the example of Figure 3, the desired behavior is to cycle the squares 5, 9, and 10 (the squares in cells (cx, cy), $(cx+1, cy)$, and $(cx, cy+1)$) clockwise, which is exactly what happens by the end of the example.

Because E_i cycles the squares in cells (cx, cy), $(cx+1, cy)$, and $(cx, cy+1)$ without affecting anything else, move sequence S' converts configuration s into configuration t as argues above. Furthermore, the number of moves in S' is equal to the number of moves in S plus the number of moves in each E_i for $i = 2, 3, \ldots, |P|$. The number of moves in S is $2c$ times the number of edges in T. Thus the number of moves in S is at most $2cl$. The number of moves in E_i is at most 18. Thus the number of moves in all of the E_is together is at most $18(|P|-1) < c$. In other words, the total number of moves in S' is less than $2cl + c = (2l+1) \cdot c = k$. Thus, S' solves the $n^2 - 1$ puzzle instance.

5. (n^2-1)-puzzle instance solution → rectilinear Steiner tree instance solution

Suppose there exists a solution to the (n^2-1)-puzzle instance. Consider a solution S using the minimum possible number of moves (where this quantity is less than or equal to k). Let R be the set of cells visited by the hole over the course of solution S.

Define a cell of the board to be (s,t)-*similar* if it contains a square in both s and t and if this square is the same square in both configurations. Notice that every non-(s,t)-similar cell a is in R. This is because there are only two ways for cell a to be non-(s,t)-similar: either cell a contains the hole in s or t or cell a contains two different squares in s and t. If cell a contains a hole in s or t, then the hole visits that cell over the course of solution S, and so by definition, $a \in R$. If cell a contains two different squares in s and t then the hole must pass through a in order to move the square placed there in s. In this case too we see that $a \in R$.

Let V be the set of coordinate pairs (x, y) of cells in R and let G be the grid graph induced by vertex set V (with edges between vertex pairs at unit distance). Because the region visited by the hole over the course of solution S (the region of R) must be a connected region, we know that G must be a connected graph. Consider any spanning tree T of this graph. The spanning tree passes through all $|R|$ vertices of G, and has $|R| - 1$ edges. Each edge of this tree has length 1, so we see that T is a tree in the plane of total length $|R| - 1$ which passes through every point (x, y) such that (x, y) are the coordinates of a cell in R. Notice that the cell in position (cx, cy) where $(x, y) \in P$ is non-(s,t)-similar, and therefore in R. In other words, the tree T of total length $|R| - 1$ passes through every point of the form (cx, cy) where $(x, y) \in P$.

Scaling T down by a factor of c, we obtain a new tree T' which passes through every point of the form (x, y) where $(x, y) \in P$, whose edges are all vertical or horizontal, and whose total length is strictly less than $\frac{|R|}{c}$. In other words, we have a rectilinear Steiner tree passing through each point of P of total length less than $\frac{|R|}{c}$. The optimal rectilinear Steiner tree through the points in P has integer length, so to solve the rectilinear Steiner tree instance (using length at most l) it suffices to show that $\frac{|R|}{c} \le l + 1$.

In order to show that $\frac{|R|}{c} \le l + 1$, we need to bound $|R|$, which we will do by bounding the number of (s,t)-similar and non-(s,t)-similar cells in R. Let $R' \subseteq R$ be the set of (s,t)-similar cells in R, so $R \setminus R'$ is the set of non-(s,t)-similar cells (in R).

First, we compute an upper bound on $|R \setminus R'|$. Every cell $a \in R \setminus R'$ is by definition non-(s,t)-similar. Thus, $|R \setminus R'|$ is at most the total number of non-(s,t)-similar cells in the whole board. During the construction of t from s, each point in $P \setminus \{p_1\}$ contributes three non-(s,t)-similar cells (namely, the three cells whose squares are permuted). In addition, the cell containing the hole in both s and t is also non-(s,t)-similar. There are no other non-(s,t)-similar cells in the whole board. Thus, there are exactly $3(|P| - 1) + 1 = 3|P| - 2$ cells that are non-(s,t)-similar. As a result, $|R \setminus R'| \le 3|P| - 2 \le \frac{c}{2}$, because $c = 18|P|$.

Next we compute an upper bound for $|R'|$. Consider any cell $a \in R'$. Because $a \in R' \subseteq R$, the hole must enter this cell at least once over the course of solution S. We wish to show that the hole must actually enter the cell at least twice. Because there are only k moves, we will conclude that $|R'| \le \frac{k}{2}$.

Suppose for the sake of contradiction that the hole enters cell a exactly once. Then at some point in S, the hole enters cell a from a neighboring cell a_{before}, and then exits into cell a_{after}. If $a_{\text{before}} = a_{\text{after}}$, then omitting these two moves leaves the overall behavior of the solution the same. This cannot be the case because S is a minimal move solution. Thus $a_{\text{before}} \neq a_{\text{after}}$. When the hole moves from a_{before} into a, the square initially in a is moved to a_{before}. Then in the next move, the hole moves from a to a_{after} and the square in a_{after} moves into a. Because $a_{\text{before}} \neq a_{\text{after}}$, this means that the square which moves into a due to these two moves is not the square that started there. But since the hole never again enters a, the square now in a never exits it. This means that the square in cell a is different in the initial configuration s and the final configuration t. That, however, contradicts the fact that a must be (s,t)-similar. As desired, this means that the hole must enter cell a at least twice.

Because the hole enters every $a \in R'$ at least twice and there are only k moves, we conclude that $|R'| \le \frac{k}{2}$.

We showed above that $|R \setminus R'| \le \frac{c}{2}$ and $|R'| \le \frac{k}{2}$. Putting this together, we see that $|R| = |R'| + |R \setminus R'| \le \frac{k}{2} + \frac{c}{2} = \frac{(2l+1)\cdot c}{2} + \frac{c}{2} = (l + 1) \cdot c$. As a result, we have that $\frac{|R|}{c} \le l + 1$, which is exactly what we wished to prove.

6. Conclusion

We have shown that the given reduction is answer preserving and polynomial-time computable. Therefore, the reduction from rectilinear Steiner tree is sufficient to prove that the $(n^2 - 1)$-puzzle problem is NP-hard.

A natural open question is to ask what number of moves is required to solve an (n^2-1)-puzzle in the worst case as a function of n. For $n = 4$, this number was shown to be 80 in [11]. This function is known to be at most $O(n^3)$ by the result from [12] and can be shown to be $\Omega(n^2)$ by a simple potential argument.

References

[1] D. Ratner, M. Warmuth, The (n^2-1)-puzzle and related relocation problems, Journal of Symbolic Computation 10 (2) (1990) 111–137.

[2] A. Brüngger, A. Marzetta, K. Fukuda, J. Nievergelt, The parallel search bench ZRAM and its applications, Annals of Operations Research 90 (1999) 45–63.

[3] A. F. Archer, A modern treatment of the 15 puzzle, American Mathematical Monthly 106 (1999) 793–799.

[4] E. R. Berlekamp, J. H. Conway, R. K. Guy, Winning Ways for Your Mathematical Plays, 2nd Edition, A K Peters, Wellesley, MA, 2001–2004.

[5] D. Kornhauser, G. Miller, P. Spirakis, Coordinating pebble motion on graphs, the diameter of permutation groups, and applications, in: Proceedings of the 25th Annual Symposium on Foundations of Computer Science, 1984, pp. 241–250.

[6] K. Yamanaka, E. D. Demaine, T. Ito, J. Kawahara, M. Kiyomi, Y. Okamoto, T. Saitoh, A. Suzuki, K. Uchizawa, T. Uno, Swapping labeled tokens on graphs, Theoretical Computer Science 586 (2015) 81–94.

[7] E. D. Demaine, R. A. Hearn, Playing games with algorithms: Algorithmic combinatorial game theory, in: Games of No Chance 3, Cambridge University Press, 2009, pp. 3–56, arXiv:cc.CC/0106019.

[8] R. A. Hearn, E. D. Demaine, Games, Puzzles, and Computation, A K Peters, 2009.

[9] M. R. Garey, D. S. Johnson, The rectilinear Steiner tree problem is NP-complete, SIAM Journal on Applied Mathematics 32 (4) (1977) 826–834.

[10] M. Hanan, On Steiner's problem with rectilinear distance, SIAM Journal on Applied Mathematics 14 (2) (1966) 255–265.

[11] A. Brüngger, A. Marzetta, K. Fukuda, J. Nievergelt, The parallel search bench ZRAM and its applications, Annals of Operations Research 90 (0) (1999) 45–63.

[12] I. Parberry, A real-time algorithm for the (n^2-1)-puzzle, Information Processing Letters 56 (1) (1995) 23–28.