

Article

## Solving the $(n^2 - 1)$ -Puzzle with $\frac{8}{3}n^3$ Expected Moves

Ian Parberry

Department of Computer Science & Engineering, University of North Texas, Denton, TX 76203–5017, USA; E-Mail: ian@unt.edu; Tel.: +1-940-565-2845

Academic Editor: Dimitris Fotakis

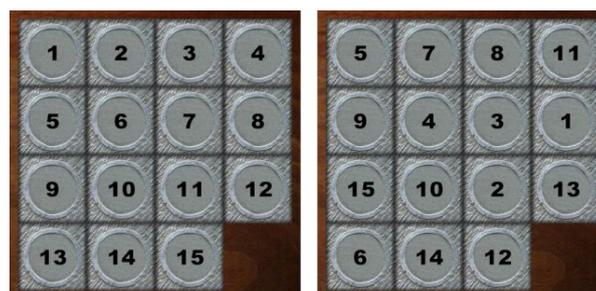
Received: 13 January 2015 / Accepted: 30 June 2015 / Published: 10 July 2015

**Abstract:** It is shown that the greedy algorithm for the  $(n^2 - 1)$ -puzzle makes  $\frac{8}{3}n^3 + O(n^2)$  expected moves. This analysis is verified experimentally on 10,000 random instances each of the  $(n^2 - 1)$ -puzzle for  $4 \leq n \leq 200$ .

**Keywords:** 15-puzzle; 8-puzzle; analysis of algorithms; average case analysis; greedy algorithm;  $(n^2 - 1)$ -puzzle

### 1. Introduction

The  $(n \times m)$ -puzzle is defined as follows. Given  $nm - 1$  numbered tiles arranged in row-major order in an  $n \times m$  grid leaving a blank space in the last position where one tile is missing, the aim is to scramble the puzzle and return it to the initial configuration by repeatedly sliding an adjacent tile into the blank location. When  $n = m$  the puzzle is also called the  $(n^2 - 1)$ -puzzle. For example, Figure 1 shows an example for the 15-puzzle of the solved configuration (left) and a random configuration (right).



**Figure 1.** The solved 15-puzzle (left), and a random configuration (right).

The 15-puzzle has a long and interesting history (see, for example, Hordern [1]) that is said to date back to the 1870s. More recently, the 15-puzzle has appeared in the form of various apps on mobile devices and as minigames inside larger games. For example, the 15-puzzle can be found in the original *Final Fantasy* (Square Enix, 1987) and *The Legend of Zelda: The Windwaker* (Nintendo 2003), and the 8-puzzle can be found in *Machinarium* (Amanita Design, 2009).

Ratner and Warmuth [2] have proved that the problem of finding the minimum number of moves for the  $(n^2 - 1)$ -puzzle is NP-hard, and they demonstrate that a polynomial time approximation algorithm exists. Kornhauser, Miller, and Spirakis [3] show an  $O(n^3)$  time algorithm for the  $(n^2 - 1)$ -puzzle, which therefore uses  $O(n^3)$  moves in the worst case.

Parberry [4] gave worst case upper and lower bounds of  $5n^3$  and  $n^3$ , respectively, on the number of moves required to solve the  $(n^2 - 1)$ -puzzle using a greedy algorithm. Whilst upper bounds are certainly interesting, a hypothetical player faced with solving a random configuration of the puzzle is likely to be more concerned about the expected number of moves than the worst case. Parberry [4] also gave lower bounds of at least  $2n^3/3$  for the expected number of moves, and at least  $0.264n^3$  moves for a random configuration with probability one.

We extend this work by showing both theoretically and experimentally that the greedy algorithm solves the  $(n^2 - 1)$ -puzzle in expected number of moves  $\frac{8}{3}n^3 + O(n^2)$ . The main body of this paper is divided into three sections. Section 2 contains a brief description of the greedy algorithm. Section 3 contains the average-case analysis of the number of moves required. Section 4 contains an experimental verification of this analysis.

## 2. The Greedy Algorithm

The greedy algorithm for the  $(n^2 - 1)$ -puzzle works as follows (for more details see, for example, Parberry [4]). There are sequences of five moves that bring a tile one place horizontally or vertically, and a sequence of six moves that brings a tile one place diagonally. Move the blank to the position immediately above the first tile, then use a sequence of these moves to bring that tile to the top left corner. Repeat this for the remaining tiles in the first row, taking care not to disturb the work that has already been done. The last two tiles in the row require a few extra moves to flip into place. Once the first row has been completed, do likewise for the first column. Once the first row and column are in place, recurse on the remaining  $((n - 1)^2 - 1)$ -puzzle. The base of the recursion can be solved by brute force, for example, the 3-puzzle can be solved in six moves, the 8-puzzle can be solved in 31 moves (Reinefeld [5]) and the 15-puzzle can be solved in 80 moves (Brünger *et al.* [6]).

## 3. Theoretical Analysis

Suppose  $n$  is even (the case where  $k$  is odd is similar). Consider the expected number of moves required to solve the first half of the first row of the puzzle. For each of those  $n/2$  tiles  $t \in \{(0, k) \mid 0 \leq k < n/2\}$ , the expected number of moves required to move tile  $t = (i, k)$  to position  $(i, k)$  will be equal to the sum over all positions  $p$  of the number of moves required to move  $t$  from position  $p$  to position  $(i, k)$ , divided by  $n^2$ . We will do this in three parts, the number of moves required to move the blank into

place above tile  $t$  (Section 3.1), and the number of moves required to move  $t$  to position  $(i, k)$  once the blank is in place (Section 3.2), and the total number of moves (Section 3.3).

### 3.1. The Blank

The sum of the number of moves required to move the blank from position  $(0, 0)$  to each of the  $n^2$  possible destinations is:

$$\sum_{i=0}^{n-1} \sum_{j=i}^{i+n-1} j = n^3 - n^2 \tag{1}$$

For example, with  $n = 8$ , looking at Figure 2 (left) and numbering the rows from 0 to  $n - 1$  top-to-bottom, row  $i$  has sum  $\sum_{j=i}^{i+n-1} j$ . For position  $(0, k)$ ,  $0 \leq k < n/2$ , the sum is:

$$(n^3 - n^2) + nk(k - (n - 1)) = (n^3 - n^2) - n(n - 1)k + nk^2 \tag{2}$$

See, for example, Figure 2 with  $n = 8$  and  $k = 0, 1, 2, 3$  from left to right. The difference between the leftmost table and the other tables is that  $k$  rightmost columns are replaced with  $k$  columns with lower values (which are the same as columns 1 through  $k$  in the leftmost table). However, the difference in value of each replaced cell is constant  $k - (n - 1)$ . Since there are  $k$  replaced columns with  $n$  rows, Equation (2) follows.



**Figure 2.** Number of moves required to move the blank to each position from the first half of the first row of the 63-puzzle.

Therefore, summing Equation (2) for  $k = 1$  to  $n/2 - 1$ , the total number of moves for positioning the blank above tiles  $\{(0, k) | 0 \leq k < n/2\}$  is:

$$\frac{n}{2} (n^3 - n^2) - n(n - 1) \sum_{k=1}^{n/2-1} k + n \sum_{k=1}^{n/2-1} k^2 = \frac{5}{12}n^4 - \frac{1}{4}n^3 - \frac{1}{6}n^2 \tag{3}$$

Hence, the total number of moves for moving the blank into place while solving the first row and the first column is less than four times Equation (3) minus Equation (1) (the latter to avoid counting cell  $(0, 0)$  twice), that is,

$$4 \left( \frac{5}{12}n^4 - \frac{1}{4}n^3 - \frac{1}{6}n^2 \right) - (n^3 - n^2) = \frac{1}{3} (5n^4 - 6n^3 + n^2) \tag{4}$$

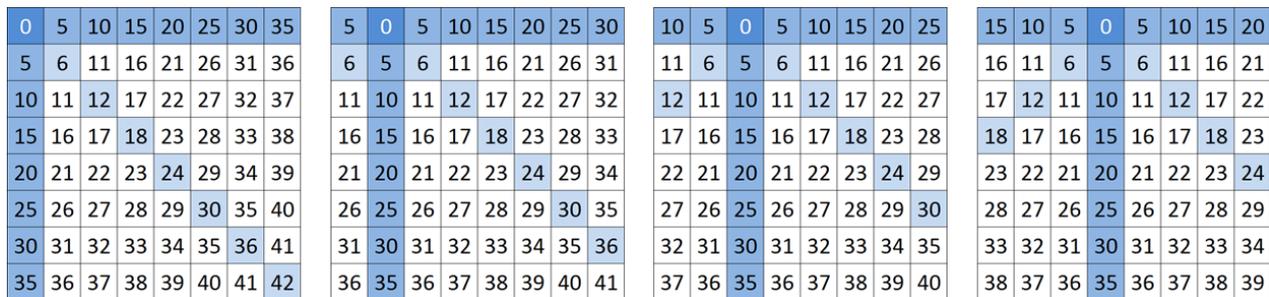
3.2. The Tiles

The sum of the number of moves required to move tile (0, 0) to position (0, 0) from all of the  $n^2$  possible sources is (see the leftmost entry of Figure 3, and Figure 4):

$$2 \left( 5 \sum_{i=1}^{n-1} i^2 + \sum_{i=1}^{n-2} \sum_{j=1}^i j \right) + 6 \sum_{i=1}^{n-1} i = \frac{11}{3}n^3 - 3n^2 - \frac{2}{3}n \tag{5}$$

Similarly, for position (0, k),  $0 \leq k < n/2$ , the sum is (see Figure 3):

$$\left(\frac{11}{3}n^3 - 3n^2 - \frac{2}{3}n\right) - (3n^2 - 4n - 4)k + (6n + 4) \sum_{i=1}^{k-1} i = \left(\frac{11}{3}n^3 - 3n^2 - \frac{2}{3}n\right) - (3n^2 - 7n - 6)k + (3n + 2)k^2 \tag{6}$$



**Figure 3.** Number of moves required to move a tile from each position to the first half of the first row of the 63-puzzle.

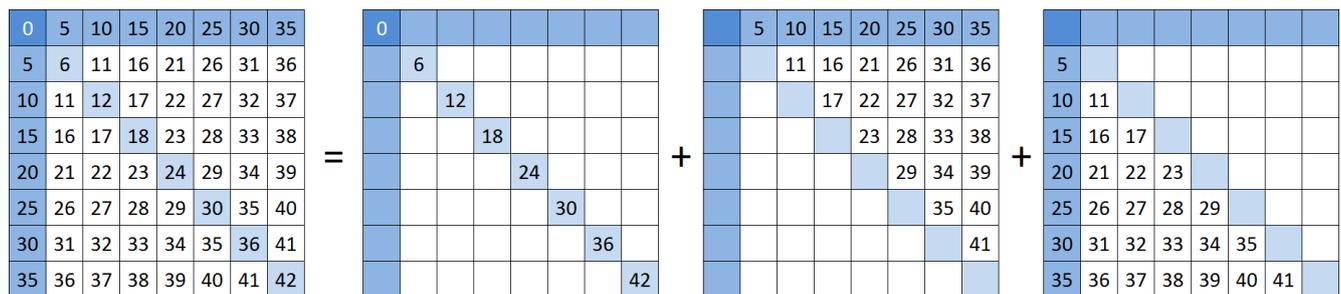
Therefore, the total number of moves for moving tiles  $\{(0, k) | 0 \leq k < n/2\}$  into place in the first half of the first row is:

$$\frac{n}{2} \left(\frac{11}{3}n^3 - 3n^2 - \frac{2}{3}n\right) - (3n^2 - n - 2) \sum_{k=1}^{n/2-1} k + (3n + 2) \sum_{k=1}^{n/2-1} k^2 = \frac{19}{12}n^4 - \frac{11}{12}n^3 - \frac{1}{3}n^2 - \frac{1}{3}n \tag{7}$$

and the total number of moves for moving the first row and column tiles into place is, by Equations (5) and (7), at most:

$$4 \left(\frac{19}{12}n^4 - \frac{11}{12}n^3 - \frac{1}{3}n^2 - \frac{1}{3}n\right) - \left(\frac{11}{3}n^3 - 3n^2 - \frac{2}{3}n\right) = \frac{1}{3} (19n^4 - 22n^3 + 5n^2 - 2n) \tag{8}$$

The astute reader will have noticed that we have under-counted by  $O(n)$  to bring the blank into position at the start, and  $O(1)$  for the last tile in the row and column. This is more than compensated for by the fact that we have over-counted by  $O(n^3)$  when moving the blank in Section 3.1 since there is never any need to move the blank to the last row.



**Figure 4.** Decomposing the leftmost entry of Figure 3 to show the structure of Equation (5).

### 3.3. Tiles and Blank Together

The total number of moves used to solve the first row and column is at most the sum of the results of Equations (4) and (8). That is,

$$\frac{1}{3} (5n^4 - 6n^3 + n^2) + \frac{1}{3} (19n^4 - 22n^3 + 5n^2 - 2n) = 8n^4 - \frac{28}{3}n^3 + 2n^2 - \frac{2}{3}n \quad (9)$$

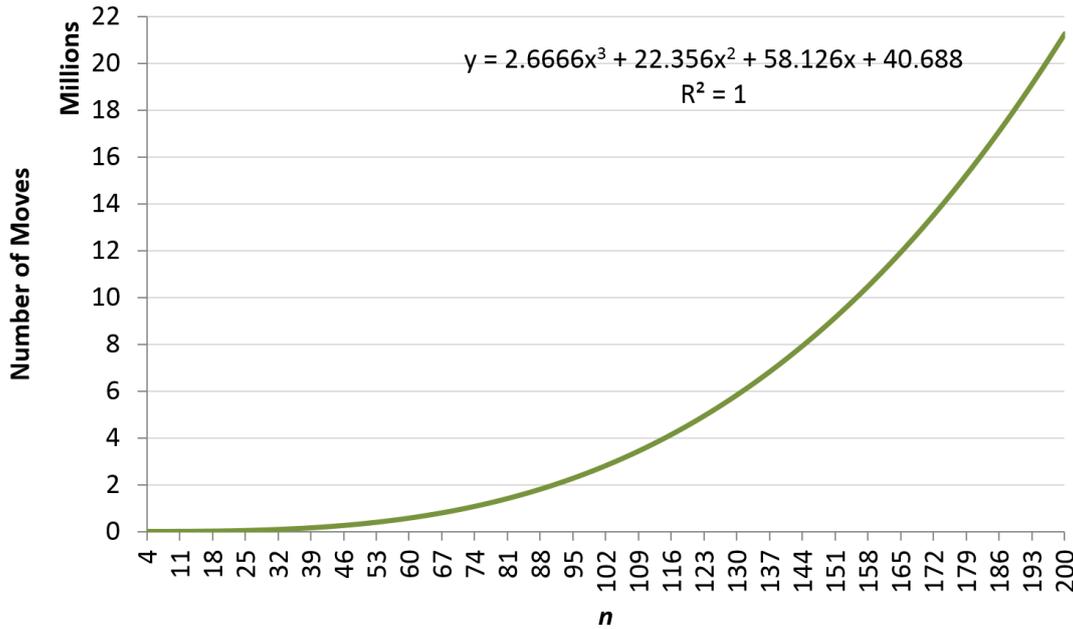
The expected number of moves to solve the first row and column is the total number of moves from Equation (9) divided by the number of tiles (which is  $n^2$ ), that is,  $8n^2 + O(n)$ . The argument so far has assumed that  $n$  is even. One can prove similarly the expected number of moves for odd  $n$  is also  $8n^2 + O(n)$ .

Having put the first row and column of the  $n \times n$  puzzle into place, the remaining  $(n - 1) \times (n - 1)$  sub-puzzle is then solved recursively. Note that if every even permutation of the  $n^2 - 1$  tiles in the  $n \times n$  puzzle is equally likely, then since the moves made to put the first row and column into place depend only on the position of those tiles in the permutation (and not, for example, on the values of any of the tiles), the resulting even permutation of the  $(n - 1)^2 - 1$  tiles in the  $(n - 1) \times (n - 1)$  sub-puzzle on which we recurse is equally likely to be any even permutation of the remaining  $n^2 - 2n$  tiles. Therefore, the expected number of moves to solve the whole puzzle is bounded above by:

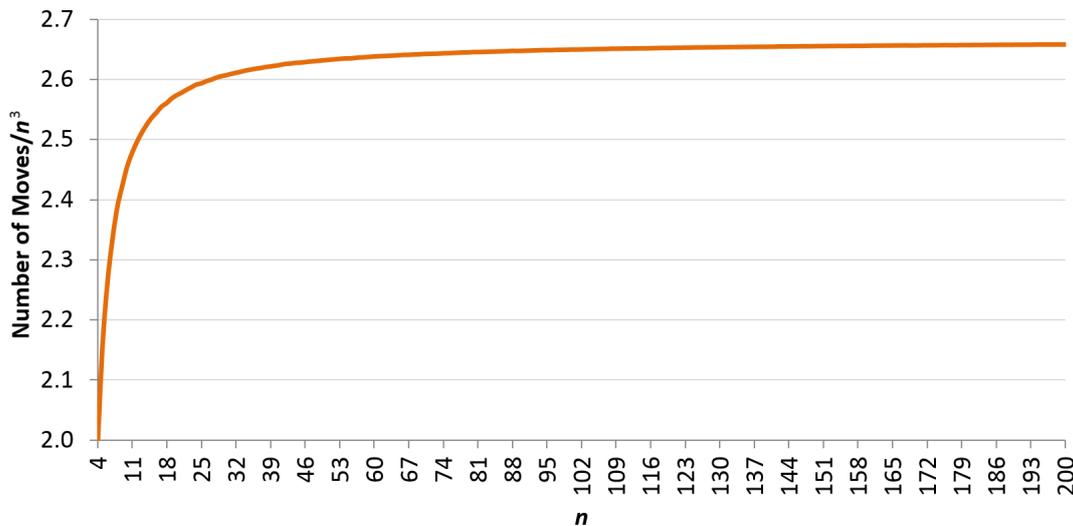
$$8 \sum_{i=2}^n i^2 + O(n^2) = \frac{8}{3}n^3 + O(n^2).$$

## 4. Experimental Analysis

We generated 10,000 random instances of the  $(n^2 - 1)$ -puzzle for all  $n$  such that  $4 \leq n \leq 200$  using the standard algorithm for generating a random even permutation based on the Mersenne Twister (Matsumoto and Nishimura [7]). We then solved each instance using the greedy algorithm and measured the average number of moves required to solve each size, which should approximate the expected value if the sample size is large enough. We found that the average number of moves tends to  $2.6666n^3 + O(n)$  with an  $R^2$  value of 1.000 (see Figure 5). In fact, the number of moves divided by  $n^3$  is less than 2.66 for  $4 \leq n \leq 200$  (see Figure 6). Consider that the theoretical bound is the expected number of moves, while the experimental bound is the average number of moves over a relatively small (compared to the solution space) random sample. The fact that the theoretical and experimental results agree so closely in this case is quite remarkable.



**Figure 5.** The average number of moves required to solve 10,000 random instances of the  $(n^2 - 1)$ -puzzle for  $4 \leq n \leq 200$ .



**Figure 6.** The average number of moves required to solve 10,000 random instances of the  $(n^2 - 1)$ -puzzle divided by  $n^3$  for  $4 \leq n \leq 200$ .

### 5. Conclusion and Open Problems

We have shown both theoretically and experimentally that the real-time algorithm from [4] solves the  $(n^2 - 1)$ -puzzle in expected number of moves  $\frac{8}{3}n^3 + O(n^2)$ . However, the best known lower bound for the expected number of moves is  $2n^3/3$  from Parberry [4]. We conjecture that there is almost certainly an algorithm with a smaller expected number of moves, and that the lower bound is also almost certainly not tight.

## Supplementary Materials

Supplementary materials can be accessed at: <http://www.mdpi.com/1999-4893/8/3/459/s1>.

## Conflicts of Interest

The author declares no conflict of interest.

## References

1. Hordern, L.E. *Sliding Piece Puzzles*; Oxford University Press: Oxford, UK, 1986.
2. Ratner, D.; Warmuth, M. The  $(n^2 - 1)$ -puzzle and Related Relocation Problems. *J. Symb. Comput.* **1990**, *10*, 111–137.
3. Kornhauser, D.M.; Miller, G.; Spirakis, P. Coordinating Pebble Motion on Graphs, the Diameter of Permutation Groups, and Applications. In Proceedings of the 25th Annual Symposium on Foundations of Computer Science, Singer Island, FL, USA, 24–26 October 1984; pp. 241–250.
4. Parberry, I. A Real-Time Algorithm for the  $(n^2 - 1)$ -Puzzle. *Inf. Proc. Lett.* **1995**, *56*, 23–28.
5. Reinefeld, A. Complete Solution of the Eight-Puzzle and the Benefit of Node Ordering in IDA\*. In Proceedings of the 13th International Joint Conference on Artificial Intelligence, Chambéry, France, 28 August–3 September 1993; pp. 248–253.
6. Brünger, A.; Marzetta, A.; Fukuda, K.; Nievergelt, J. The parallel search bench ZRAM and its applications. *Ann. Oper. Res.* **1999**, *90*, 45–63.
7. Matsumoto, M.; Nishimura, T. Mersenne Twister: A 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simulat.* **1998**, *8*, 3–30.

© 2015 by the authors; licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution license (<http://creativecommons.org/licenses/by/4.0/>).