Communication in Physical Sciences, 2023, 10(3):133-138

Solving towers of Hanoi problem using 2-Consecutive moves Algorithm

Osaghae Edgar O. and Obi Jonathan Chukwuyeni

Received: 16 September 2023/Accepted 23 December 2023/Published 28 December 2023

Abstract: *The problem of Hanoi is a classical one and getting the optimal solution has posed serious challenges to puzzle researchers over the years. This research paper proposed a novel 2-consecutive move algorithm solution to 3-peg towers of the Hanoi problem, which allows a 2-consecutive moves algorithm that moves 2 disks at once in each move instance of the Hanoi algorithm. There are no 2-consecutive moves for the first and last moves; only one disk is moved for these instances. The main purpose of this research work is to derive a 2-consecutive moves algorithm that can be easily implemented in a suitable programming language, and would substantially reduce the computational time to fully compute the Hanoi solution.*

Keywords*: Towers of Hanoi; algorithm; recurrence relations; parallel algorithm; complexity*

Osaghae Edgar O*
Federal University Lokoja, Lokoja, Kogi State, Nigeria.
Email: edgar.osaghae@fulokoja.edu.ng
Orcid id: 0000-0002-2643-323X

Obi Jonathan Chukwuyeni
University of Benin, Benin City, Edo State, Nigeria.
Email: jonathan.obi@uniben.edu
Orcid id: 0009-0007-7560-6639

1.0 Introduction

A conventional Tower of Hanoi problem is basically a 3-peg structure comprised of a source (peg 1), intermediate (peg 2) and destination (peg 3). The disks of Hanoi are in decreasing diameters, and are initially stacked on the source (peg 1). To solve the problem of Hanoi, the disks need to be moved one at a time in a sequential order whereby, only the topmost disk on each peg can be moved (rule 1), and a larger disk may never be placed on top of a smaller one (rule 2). The objective is to move each disk from peg 1 to peg 3 without violating the rules (rule 1 and rule 2). The minimum number of moves to place all disks of Hanoi on peg 3 is $2^n - 1$ (Hysom and Pothen, 2001; German, 2012; Majumdar et. al., 2020; Obandan and Obahiagon, 2015; Leiss and Mackey, 2018).

In this paper, there is an attempt to use 2-consecutive moves to arrange the disks of Hanoi, from peg 1 to peg 3. In the experiment, a maximum of 2 consecutive moves were possible to accurately place the all the disks on peg 3 without violating any of the rules of the arrangement process.

Using parallel algorithms to solve sequential algorithms can make these algorithms run optimally. Some researchers use number theory to either make their algorithms more efficient or derive alternative approaches to solving these algorithms (Hysom and Pothen, 2001; Ikpotokin et al., 2004a; Ikpotokin et al., 2004b; Bhaskar and Shubham, 2015; Chetverushkin et al., 2022; Chiemeke and Osaghae, 2006; Osaghae, 201; Osaghae et al., 2007; Lu and Dillon, 1994; Guillaume and Jean-Charles, 2018; Meng et al., 2016; Miller et al., 2021; Israa and Abdalameer, 2018). Solving towers of Hanoi problem is computationally expensive and there is a need to find ways to introduce parallel algorithms, to speed up the computational process (Wu and Chen, 1992; Kathavatel and Srinath, 2014).

Existing Researchers have been trying to speed up the computational time for solving towers of Hanoi problem, by trying to develop novel parallel algorithms to replace sequential algorithms. Some of the reviewed are discussed as follows.

Ikpotokin *et al*. (2014a) proposed a rule of thumb that dictates how the use of number theory can help determine when to begin the first placement of disk 1 on either P2 or P3. Although, this research work provided an excellent guide in providing the foundation for this paper, especially when a disk needs to be moved to an empty peg. The limitations of this research work are that there is no guide on which disk to place on an empty peg and there was no application of parallel algorithm to speed up the algorithm.

Mishra and Vishnoi (2021) proposed an iterative approach to apply a parallel algorithm in solving the Hanoi problem. They used *n* system cores of the computing running time for each system, given the conventional time *T* to successfully run the Hanoi algorithm that would reduce to *T/n* time. They suggested future direction of the research work, to involves using a memorization matrix to further increase the efficiency of the computational time of the algorithm.. Lu and Dillon (1995) also proposed three versions of parallelism for the multi-peg Towers of Hanoi problem. These parallel paradigms do not involve the exchange of disks, which is allowed in conventional paradigms. They noticed three problems associated with this approach, which are: (i) they have not been able to solve the problem for full Towers without certain constraints. (ii) the solution with a minimum number of steps does not necessarily take the minimum number of moves. (iii) an unrestricted case allowing any number of disks. However, Wu and Chen (1992) proposed a variant of the towers of the Hanoi problem allowing parallel moves. In their paper, every top disk may be simultaneously moved from its peg and placed on another peg at a given time. The challenge of this research is that no more than one disk can be placed on the same peg. Therefore, the present study seek to use 2-consecutive disk moves to arrange the disks of Hanoi, from source peg to destination peg.

2.0 Materials and Methods

In this section, we are going to provide a solution for Hanoi using a 2-consecutive moves algorithm to arrange the disks on the peg 3 of the Hanoi.

a) Problem:

You are given three pegs labelled A, B and C.

i) On peg A, there are *n* disks of different sizes, in the order of the largest disk on the bottom to the smallest one on top.

ii) Pegs B and C are empty.

iii) The aim is to move the n-1 disks (n>1) from peg A to peg B by successively moving two 2-consecutive disks from a peg to another peg in a succession of $(2^n) / 2$ moves and eliminate $2^n – ((2^n) / 2)-2$ moves.

b) Inductive Proof of Puzzle Solution with two Consecutive moves

(basis) One disk can be moved from one peg to another peg, if the number of disks=1. When the number of disks > 1, two consecutive disks can be moved in succession, one after the other.

(inductive step) Assume that we can move two-consecutive disks from any given peg to any other peg. Since any of the disks 1 through *n-1* can be placed on top of disk *n*, all *n* disks can be placed as follows:

i) Move *n-1* disks from peg A to peg B in 2-consecutive disk moves, where *n>1*.

ii) Move disk *n* from peg A to peg C, where disk 1 and then disk *n* are moved to the top of disk 2 and peg C respectively.

iii) Move *n-1* disks from peg B to peg C in 2-consecutive disk moves, except disk 1 which moves without 2-consecutive disk moves.

Example 1: The solution for 1 disk takes 1 move:

1						1
A	**B**	**C**		**A**	**B**	**C**

The minimum number of moves is $2^n – 1$; $2^1 – 1 = 1$ move, for $n=1$.

Example 2: The solution for 2 disk takes 2 moves:

```
1                        1
2  _  _    _  1  2   _  _  2
A  B  C    A  B  C   A  B  C
```

The minimum number of moves is supposed to be $2^n - 1$; $2^2 - 1 = 3$ moves, for $n=2$, but rather, we have 2 moves, when two consecutive disks were made in the first move.

Example 3: The solution for 3 disks takes 4 moves:

The solution for 3 disks takes 4 moves:

```
1
2                    1           2
3  _  _    3  2  1   _  2  3   1  _  3
A  B  C    A  B  C   A  B  C   A  B  C

      1
      2
_  _  3
A  B  C
```

The minimum number of moves is supposed to be $2^n - 1$; $2^3 - 1 = 7$ moves, for $n=3$, but rather, we have 4 moves, when two-consecutive disks were made in the first, second and third moves, 3 moves that were supposed to be made serially, were made consecutively.

Example 4: The solution for 4 disks takes 8 moves:

```
1
2
3        3              1  1  2
4  _  _  4  1  2   4  3  2  4  3  _
A  B  C  A  B  C   A  B  C  A  B  C

   1                                2
   2           1  1     3           3
_  3  4  2  3  4  2  _  4   _  1  4
A  B  C  A  B  C  A  B  C   A  B  C
      1
      2
      3
_  _  4
A  B  C
```

The minimum number of sequential moves is supposed to be $2^n - 1$; $2^4 - 1 = 15$ moves, for $n=4$, but rather, we have 8 moves, when two consecutive disks were made in the first to seventh moves, 7 serial moves were eliminated.

Example 5: The solution for n-1 odd-numbered disks takes $(2^n) / 2$ moves:

```
1              3              4                1
2              4              5                4
…              …              …                …
n-2            n-2            n-2  1           n-2      2
n-1   _   _    n-1  1   2     n-1  2   3       n-1  _   3
A     B   C    A    B   C     A    B   C       A    B   C
            3              2              1
            4              3              2
…           …              …              …
       1    n-2            n-2            n-2
   _   2    n-1   1    _   n-1      _  _  n-1
   A   B    C     A    B   C        A  B  C
```

The minimum number of sequential moves is supposed to be $2^n - 1$ moves, for $n-1$ odd numbered disks, but rather, we have $(2^n) / 2$ moves, when two-consecutive disks moves were made from move 1 to move $((2^n) / 2)-1$ moves, $2^n - ((2^n) / 2)-2$ serial moves were eliminated.

Example 6: The solution for n even numbered disks takes $(2^n) / 2$ moves:

```
1              3              4                1
2              4              5                4
…              …              …                …
n-1            n-2            n-2      1       n-2  2
n     _   _    n-1  2   1     n-1  3   2       n-1  3   _
A     B   C    A    B   C     A    B   C       A    B   C

               3              2              1
               4              3              2
…              …              …              …
      1        n-1            n-1            n-1
      2   _    n        _  1  n        _  _  n
      A   B    C        A  B  C        A  B  C
```

The minimum number of sequential moves is supposed to be $2^n - 1$ moves, for $n-1$ odd numbered disks, but rather, we have $(2^n) / 2$ moves, when two-consecutive disks moves were made from move 1 to move $((2^n) / 2)-1$

moves, $2^n - ((2^n) / 2)-2$ serial moves were eliminated.

c) Recursive Algorithm: The proof by induction is constructive because it tells you how to solve a problem with n disks in terms of solutions to the problems with n-1 disks; it corresponds directly to a recursive program. The procedure HANOI(n, a, b, c, c_2, num), implements the 2-consecutive algorithm of 3-Peg of Hanoi. The parameter n is the number of disks of the Hanoi. The parameters a, b and c are the Pegs a (source), b(intermediate) and c (destination) respectively. The parameters c_2 and num are the 2-consecutive value and even/odd numbered of n disks. The inner procedures HANOI(n-1, a, c, b, c_2, num) implements the segment of the algorithm that moves the set of disks from Peg a to b using the support of Peg c. To ensure that the Hanoi is properly arranged, the num argument, which is either an even or odd number, would determine which Peg (either b or c) to place the first disk (disk 1). The second inner procedure HANOI(n-1, b, c, a, c_2, num) implements the segment of the algorithm that moves the set of n-1 disks from Peg b to c using the support of Peg a. When the procedure HANOI(n-1, a, c, b, c_2, num) and HANOI(n-1, b, c, a, c_2, num) have moved the set of disks 2 to n-1 to Peg c, the disk 1 is move from Peg a to Peg c, if n is odd or from Peg b to Peg c is n is even.

```
procedure HANOI(n, a, b, c, c_2, num)
    if n==1 then begin
        print ("move disk 1 from peg a to peg c")
    end
    if n>1 then begin
        HANOI(n-1, a, c, b, c_2, num)
        print ("move disk n from peg a to peg c")
        HANOI(n-1, b, c, a, c_2, num)
        If num==even_number then begin
            print ("move disk 1 from peg b to peg c")
        end
        If num==odd_number then begin
            print ("move disk 1 from peg a to peg c")
        end
    end
end
```

d) Complexity of HANOI

Recurrence relation for the number of moves: when $n = 1$, no 2-consecutive moves are necessary, it just to move disk 1 from Peg a to Peg c. Using 2-consecutive movement of disks, when n >1, the first call of the procedure HANOI, recursively moves the set of disks range from disk 1 to disk n-1. The print statement prints move disk n from Peg a to Peg b. Then the HANOI procedure is called again to implement the moves of the range of disk 2 to disk n-1, from Peg b to Peg c. Lastly, disk 1 is moved from Peg A to Peg c, if n is an odd number and from Peg b to Peg c, if n is an even number. Hence, the initial number of moves made by HANOI on n is:

$$T(n) = 2^n - 1 \text{ moves} \quad (1)$$

When 2-consecutive moves are made in the movement of the disks from Peg a to Peg c, $T(n) = (2^n) / 2$ moves. The number of eliminated moves is $2^n - ((2^n) / 2)-2$.

Theorem: for $n \geq 1$, HANOI(n, a, b, c, c_2, num) makes $(2^n) / 2$ moves.

Proof: For $n = 1$:

$T(1) = (2^1) / 2 = 1.$

For $n = 2$:

$T(2) = (2^2) / 2 = 2.$

For $n = 3$:

$T(3) = (2^3) / 2 = 4.$

For $n = 4$:

$T(4) = (2^4) / 2 = 8.$

…

For $n = n$:

$T(n) = (2^n) / 2.$

3.0 Results and Discussion

In this section, we are going to discuss the advantages the 2-consecutive moves of Hanoi could bring to the computational process of computing the Hanoi puzzle. Solving the 3-Pegs of Hanoi problem requires the formula $2^n - 1$, which would aid the computation of the number of moves of Hanoi. The experiment shown in this paper led to the derivation of a novel algorithm, which provides 2-consecutive moves in the Hanoi algorithm, that assist a researcher providing a prelude for the introduction of 2-parallel moves to the solution of Hanoi. In moving the disks from Peg A to Peg C, is aided by making 2 consecutive moves, at each movement of the disks, except the movement of the final disk (disk 1 to Peg C). The final movement of the last disk (disk 1) to Peg C does not use 2-consecutive moves. Since the 2-consecutive moves algorithm is a prelude to the introduction of a 2-parallel algorithm to the Hanoi problem, it has been shown how this research has reduced the number of moves of Hanoi. The number of moves of the 2-consecutive method of Hanoi is $(2^n) / 2$ and the number of moves eliminated is $2^n - ((2^n) / 2)) - 1$. The main reason for this research work is to provide a prelude for the introduction of a 2-parallel algorithm, to reduce the number of moves of Hanoi.

4.0 CONCLUSION

In this paper, research work was conducted to use 2-consecutive moves in the movement Hanoi disks from one Peg to another. The results of the derivation of the 2-consecutive moves algorithm, it shows that the new algorithm has drastically reduce the number of moves of Hanoi from $2^n - 1$ moves to $(2^n) / 2$ moves. In future research work, there would be an attempt to implement a 2-parallel algorithm, that would use the result of this research work.

5.0 REFERENCE

Bhaskar K. M. & Shubham V. (2015). Tower of Hanoi-An Iterative Solution For Parallel Computation, *International Journal of Research Science and Management*, 2, 7, pp. 1-7.

Chetverushkin B., Chechina A., Churbanova N. & Trapeznikova M. (2022). Development of Parallel Algorithms for Intelligent Transportation Systems, *Mathematics*, 10, 643, pp. 2-18.

Chiemeke S. C. & Osaghae E. O. (2006). The turning points in the solution of n-queens problem using backtracking method, *Journal of Nigerian Association of Mathematical Physics*, 2, 10, pp. 107-114.

German V. (2012). Annotation of logic programs for independent AND-Parallelism by partial evaluation, *Cambridge University Press*, pp. 583-600.

Guillaume P. & Jean-Charles R. (2018). Parallel Algorithms for Operations on Multi-Valued Decision Diagram, *Association for the Advancement of Artificial Intelligence*, 6625-6632.

Hyedeh A., Comfar B. & Abbass N. (2011). On the Solution of the Towers of Hanoi Problem, *World Academy of Science, Engineering and Technology, International Journal of Mathematical and Computational Sciences*, 5, 1, pp. 59-62.

Hysom D. and Pothen A. (2001). A scalable parallel algorithm for incomplete factor preconditioning, *Society for Industrial and Applied Mathematics*, 22, 6, pp. 2194-2215.

Ikpotokin F. O., Chiemeke S.C. & Osaghae E.O. (2004a). Alternative Approach to the Optimal Solution of Tower of Hanoi, *Journal of Institute of Mathematics and Computer Sciences,* 15, 2, pp. 229-244.

Ikpotokin F. O., Chiemeke S. C. & Osaghae E. O. (2004b). Generalized Recursive Optimal Solution for the Multi-Peg Towers of Hanoi, *Journal of Institute of Mathematics and Computer Sciences,* 15, 2, pp. 297-309.

Israa M. & Abdalameer A. (2018). Designing Parallel Algorithms for Parallel System, *Iraqi Journal of Information Technology*, 9, 2, pp. 33-47.

Kathavatel S. & Srinath N. K. (2014). Efficiency of parallel algorithms on multi-core systems using OpenMP, *International Journal of Advanced Research in Computer and*

Communication Engineering, 3, 10, pp. 8237-8241.

Leiss E. & Mackey I. (2018). The worst Parallel Hanoi graphs, *Theoretical Computer Science*, 705, pp. 1-8.

Lu V. M. & Dillon T. S. (1995). Parallelism for Multipeg Towers of Hanoi, *Mathematical Computational Modelling*, 21, 3, pp. 3-17.

Lu V. M. & Dillon T. S. (1994). A note on parallelism for the towers of Hanoi, *Mathematical Computational Modelling*, 20, 3, pp. 1-6.

Majumdar A. A. K. & Islam S. M. S. (2020). On the Sinner's Tower of Hanoi, *Journal of Science and Technology,* 18, pp. 85-92.

Meng Q., Ke G., Wang T., Chen W., Ye, Q., Ma Z. & Liu T (2016). A communication efficient parallell algorithm for decision tree,. *30th Conference on Neural Information Processing Systems, Barcelona, Spain*, 2016.

Miller J., Trümper L., Terboven C. & Müller M. S. (2021). A Theoretical Model for Global Optimization of Parallel Algorithms, *Mathematics*, 9, pp. 2-14.

Obandan S. & Obahiagon K (2015). Rule based expert system approach towards the resolution of the tower of hanoi, *Journal of Resourcefulness and Distinction*, 10, pp. 6-18.

Osaghae E. O. (2021). Solution to n-Queens Problem: Heuristic Approach, *Transactions On Machine Learning and Artificial Intelligence*, 9, 2, pp. 26-35

Osaghae E. O., Akanmu A. A. G. & Akhahowa A. E. (2007). Solution to A NON-STANDARD INITIAL STATE of The Tower of Hanoi Problem, *International Journal of Physical Sciences*, 2, 1, pp. 53-61.

Wu J. S. & Chen J. R. (1992), The towers of Hanoi problem with parallel moves, *Information Processing Letters, 44, pp.* 241-243.

Compliance with Ethical Standards Declarations

The authors declare that they have no conflict of interest.

Data availability

All data used in this study will be readily available to the public.

Consent for publication

Not Applicable

Availability of data and materials

The publisher has the right to make the data Public.

Competing interests

The authors declared no conflict of interest.

Funding

There is no source of external funding

Authors' contributions

Osaghae Edgar O. main contribution was in deriving the rule of thumb regarding how to move 2-consecutive disks of Hanoi, to reduce the time of arranging the disk from source peg to destination peg.Obi Jonathan Chukwuyeni main contribution was to developed the algorithm for implementing the 2-consecutive moves of the Hanoi.