A New Structured Knight Tour Algorithm by Four Knights with Heuristic Preset Rules

Tan Chi Wee[1], Nur Hafizah Ghazali[2] and Ghazali Bin Sulong[1,3]
[1]*Faculty of Computing, Universiti Teknologi Malaysia, 81310 Johor Bahru, Johor, Malaysia.*
[2]*School of Computer and Communication Engineering, Universiti Malaysia Perlis, 02600 Arau, Perlis, Malaysia.*
[3]*School of Informatics and Applied Mathematics,*
Universiti Malaysia Terengganu, 21030 Kuala Nerus, Terengganu, Malaysia.
seantan8989@gmail.com

***Abstract*—A Knight Tour problem is an ancient puzzle which remains as a focus of current researcher. The objective of a Knight Tour algorithm is to find out a solution with construct a closed moving path by a knight for visiting each square of a chessboard by only once. This paper proposes a new structural Knight Tour algorithm by four knights moving all together with heuristic preset rules. This new proposed method fulfills any board size of 4n+2 where n>1 with lower execution time and without the needs of any brute force method and backtracking method.**

***Index Terms*—Heuristic; Knight Tour; Preset Rules; Structural.**

I. INTRODUCTION

A Knight Tour problem is an old but interesting issue that still arouse the attention of researcher nowadays. The criteria of a legal Knight Tour move require each of the square checker only allow visited by Knight by only once. Knight Tour, in another word, is sharing common situation with Travelling Salesman problem, where Knight Tour is an un-weighted for each square. Knight Tour problem on a chessboard of size 8×8 first studied by Euler in year of 1759 [1]. Right after, it become a famous topic in mathematic field with many different solutions such as [2]–[6]. Initially, research has been conducted on a small chessboard, as a result, many skills has been proposed such as brute force algorithm, divide-and-conquer algorithm and neural network solution. The advantages of these methods are quick and the easy to implement. Over the decade, Knight Tour getting its own place and involved in many other research fields such as watermarking and steganography. As a result, Knight Tour problem become crucial when it come across a bigger chessboard as previous solution was not able to cater the latter.

This paper suggests a new structural Knight Tour algorithm by four Knight Tour with heuristic rules which able to fit on a large chessboard size. The remaining of the paper is organized as follow: Section II will describe about some significant research of previous researcher. The algorithm and new proposed method and its' implementation will be explained in Section III. Further, discussion and experimental result from example will be given in Section IV for better understanding of the proposed method. Conclusion with future work and reference of this method is provided in Section V and Section VI respectively.

II. EXISTING METHODS

The common method of constructing the closed Knight Tour move is moving the Knight from a single starting point to the ending point on a sequential way. Ball and Coxeter [7] proposed a bisected Knight Tour solution by dividing the chessboard into two rectangular regions and limit the Knight to finish the visit of first region before proceeding to second region. This method revised by Dudeney in his book [8], where the idea of former is extended into four equal parts with each the same size and shape and trespass twice on the other parts. Domoryad [9] use the idea of quadrisect Knight Tour as [8] but without visit the center square checker. Hurd and Trautman [10] on the other hand, proposed a opened Knight Tour move which produces the almost same result as [9] with different algorithm and the unvisited square shifted to one of the corner of the chessboard. Parberry [2] invented a divide-and-conquer algorithm and introduced a concept of "structured" Knight Tour which able to construct a closed Knight Tour moves of various types. By mean of dividing the chessboard, a closed legal move also able to calculate when encounter larger chessboard.

III. IMPLEMENTATION

In this paper, a new Structured Knight Tour Algorithm by four Knights with heuristic preset rules has been proposed. Figure 1 below shows the overall flowchart of the proposed method in this paper. This research is done by using a laptop equipped with Intel Core i7-2670 QM 2.2 GHz processor with 8.0 GB of RAM. The algorithm and programming is implemented by using MATLAB R2013a.

We take the structured concept by Parberry [2] but without the needs of dividing the large board. The idea of a total number of four Knights is being employed is a consequence idea from structural Knight Tour. The structural Knight Tour predefined a set of moves for each four corners with each corner square are interconnected with another two squares and two of the neighborhood squares linked to other squares as shown in Figure *2*.

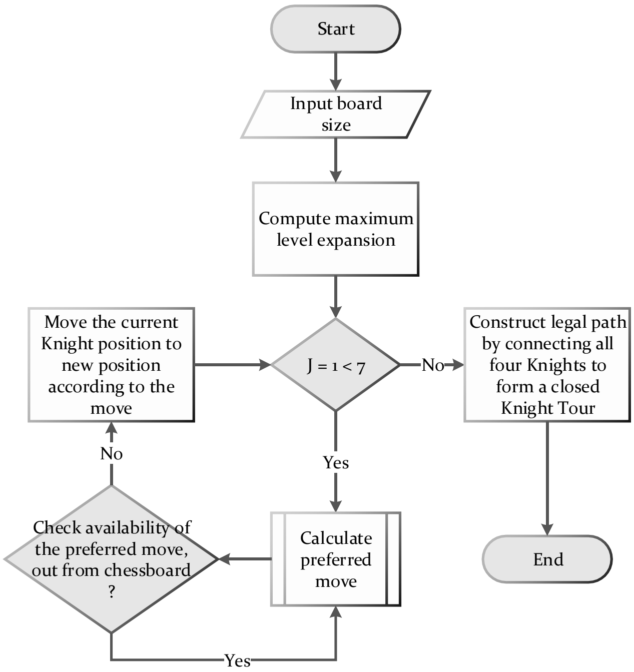


Figure 1: Flowchart of proposed method

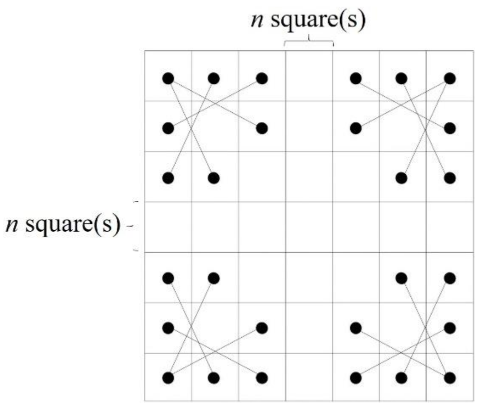


Figure 2: Structural Knight Tour

Instead of convention "Trial and Error" or Backtracking algorithm of guessing the correct move (complete a closed Knight Tour move), three new rules with respect to structural Knight Tour has been invented to optimize the Knight Tour especially in large dimension chessboard. These heuristic preset rules suggest the possible move to four Knights from each corner and those suggested moves always being the best move to complete the Knight Tour sequence.

Preset Rule 1: Knight only allowed to move outward against margin or outer square of the chessboard (if possible).

Preset Rule 2: Knight only allowed to make a swerve at corner of a chessboard without bouncing back to the direction where it comes from.

Preset Rule 3: Knight only allowed to move to another square with respect to structural Knight Tour as illustrated at Figure 2, in other words, only one choice for Knight when it comes to the corner square.

Before the Knight Tour started, each corner is placed a Knight, K_1, K_2, K_3, and K_4. Each of four Knights will connected with two squares at the beginning of the tour (Preset Rule 3). Consequence, Knight 1 is connected to SW_1 and SW_2, Knight 2 is connected to NW_1 and NW_2, Knight 3 is connected to NE_1 and NE_2 whereby Knight 4 is connected to SE_1 and SE_2 (Figure *3*).

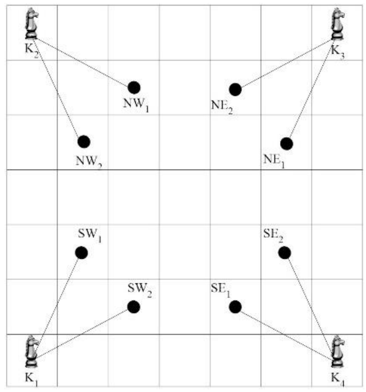


Figure 3: Position of K_1, K_2, K_3 and K_4 with their connect squares

When the tour starts, each of the connected square will move according the priority as shown in Table 1 where Preset Rule 1 and Preset Rule 2 applied. If the Priority move 1 is not granted, then it will move according to Priority move 2 and so on (The lower the value, the higher of priority).

Table 1
Preferred moving direction according the priority for each Knight

Priority	SW_1		SW_2		NW_1		NW_2		SE_1		SE_2		NE_1		NE_2	
	x	y	x	y	x	y	x	y	x	y	x	y	x	y	x	y
1	-1	2	2	-1	2	1	-1	-2	-2	-1	1	2	1	-2	-2	1
2	1	2	2	1	2	-1	1	-2	-2	1	-1	2	-1	-2	-2	-1
3	2	-1	-1	2	-1	-2	2	1	1	2	-2	-1	-2	1	1	-2
4	-2	1	1	-2	1	2	-2	-1	-1	-2	2	1	2	-1	-1	2
5	-2	-1	-1	-2	-1	2	-2	1	1	-2	2	-1	2	1	1	2
6	2	1	1	2	1	-2	2	-1	-1	2	-2	1	-2	-1	-1	-2
7	1	-2	-2	1	-2	-1	1	2	2	1	-1	-2	-1	2	2	-1

From the Table *1*, it is clearly shows that different preferred move is defined for each Knight with different priority. Taking the consideration, as an example, we prefer that SW_1 is always moving against the left part of chessboard with moving direction of [-1,2], but when it comes the upper part of the chessboard, SW_1 is targeted to move its foot step to the right of the chessboard along the chessboard. That is the second priority of the SW_1. The third priority move for SW_1 is moving downward along the right side of the chessboard, and it explains the preferred move of third priority is [2,-1] as noted by negative value of coordinate y because we want the Knight moving down along y axis. Same concept is applied for other Knights and its connected square also.

IV. Discussion and Experimental Results

As the Knight iterate and expand each time, more and more square of the chessboard are being covered. It is good to mention that this structured Knight Tour fulfill the board size, m where $m = 4n + 2$, where $n \subset \mathbb{Z}^+$ for a closed completed Knight Tour path.

Before the Knight Tour start, we already have a total of 12 squares covered (4 corner Knights + 8 interconnect squares). The number of square "stepped" by each Knight through two squares will be increased by 8. This is because each of the square will move into one new square based on the heuristic rules given (Table *2*) whereby maximum level of expansion, E is given by $E = \frac{m^2-12}{8}$ where $m = 4n + 2$. Maximum level of expansion ensures the all square are being covered without revisit of any of the square.

Table 2
Cumulative number of square covered each iteration

Level of Expansion, E	Total cumulative number of square covered, $\sum_n$
1	12
2	20
3	28
4	36
5	44
6	52
⋮	⋮
n	$\sum_{n-1} + 8$

When all the square is covered, in other words, the maximum level of expansion, E is achieved, Knight Tour success in its way to construct its' own path. Next thing, we combine all eight paths into a single path because each of eight paths will meet with each other at the end node of their path respectively. In this context, SW1, NW2, SE2, NE1 after expansion E will be connected to NW1, NE2, SW2 and SE1 accordingly. After contrasting the path, a closed perfectly structural Knight Tour is formed and the starting point is now versatile with any point of along the path.

Figure 4 and Figure 5 show the example of proposed method on chessboard of board size, m=6 and m=10 respectively. Noted that, the sequence number of the Knight Tour is also displayed along the path. (Take note that the starting point in our example is [1,1] for better understanding).

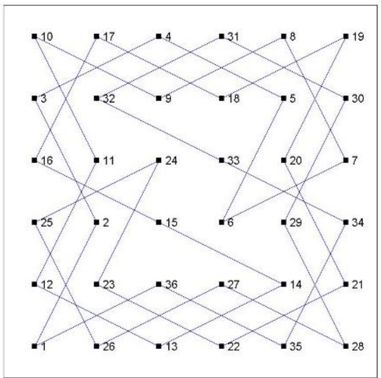


Figure 4: Proposed solution on a 6×6 *chessboard*

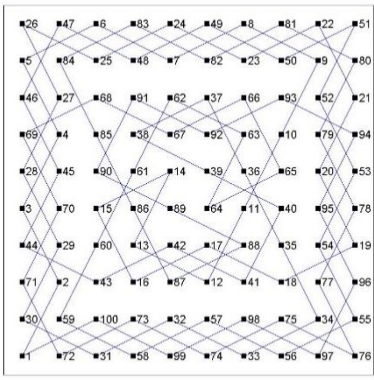


Figure 5: Proposed solution on a 10×10 *chessboard*

In order to make sure all of the square in the chessboard is covered, this proposed algorithm also able to foresee the maximum level of expansion needed for each Knight. The maximum level of expansion with execution time for some tested board size, m is showing at Table 3 below.

Table 3
Experimental data with sample board size with execution time

Board Size, M	Total Square in Chessboard	Maximum Level of Expansion,	Execution Time, T Seconds
6	36	3	0.00997
10	100	11	0.02735
14	196	23	0.05156
18	324	39	0.08133
22	484	59	0.12350
26	676	83	0.16182
30	900	111	0.20765
102	10404	1299	2.4013
210	44100	5511	15.1955
502	252004	31499	472.1605

As a comparison, Table 4 below shows the execution time that found from previous significant study with time provided. The proposed method outperforms all of the previous study on term of execution time with a very outstanding result. The time needed for proposed method generally is about less than 1 second to form a closed Knight Tour path.

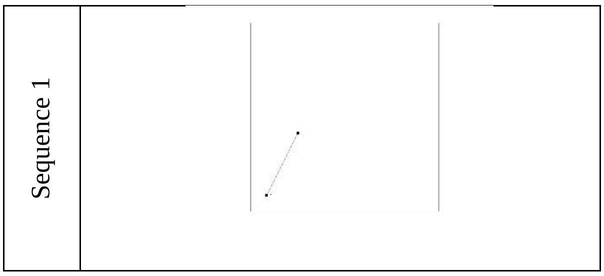
Table 4
Comparison of execution time with some significant study with time provided

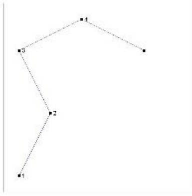
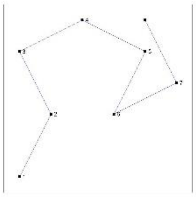
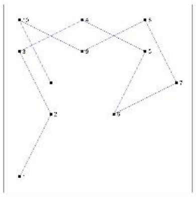
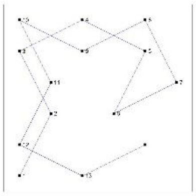
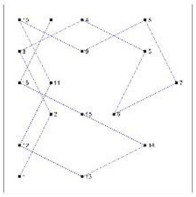
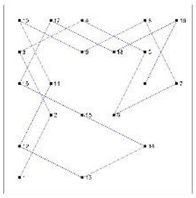
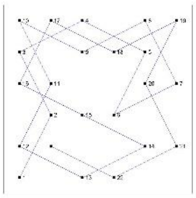
Year	Method	Platform	Execution Time
1995	UNSW [11] Prolog	SCO UNIX 33 MHz	45 min
1995	UNSW Prolog	Sparc Center 2000 66 MHz	22 min
1996	Binary Decision Diagrams	SUN 670/140	6.5 min
2008	Brute Force using Prolog [5]	Intel PC 3.0 GHz	35 s
2017	Proposed method	Intel Laptop 2.2 GHz	< 1 s

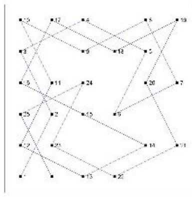
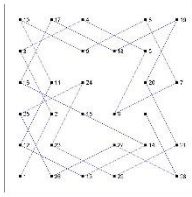
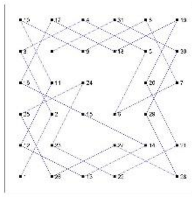
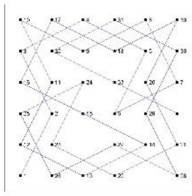
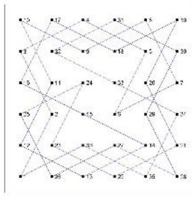
V. Conclusion

This proposed method suggests an algorithm that able to cater Knight Tour problem with shorter execution time by using three heuristic preset rules and the concept of structural Knight Tour. With the present of preset rules, the execution time for a completely closed Knight Tour able to be generated in a very abbreviated time. Further, this algorithm able to solve Knight Tour in a huge board of board size larger than 500 which is not achieved by most of the researcher.

Appendix A



Sequence 4	
Sequence 7	
Sequence 10	
Sequence 13	
Sequence 16	
Sequence 19	
Sequence 22	

Sequence 25	
Sequence 28	
Sequence 31	
Sequence 33	
Sequence 36	

REFERENCES

[1] L. Euler, “Solution d’une question curieuse que ne paroit soumise a aucune analyse,” in *Mémoires de l’Academie Royale des Sciences et Belles Lettres*, 1759, pp. 310–337.

[2] I. Parberry, “An efficient algorithm for the Knight’s tour problem,” *Discret. Appl. Math.*, vol. 73, no. 3, pp. 251–260, 1997.

[3] S. S. Lin and C. L. Wei, “Optimal algorithms for constructing knight’s tours on arbitrary n×m chessboards,” *Discret. Appl. Math.*, vol. 146, no. 3, pp. 219–232, 2005.

[4] L. Paris, “Heuristic Strategies for the Knight Tour Problem,” in *Proc. Int. Conf. Artif. Intell. IC-AI ’04, Vol. 2 Proc. Int. Conf. Mach. Learn. Model. Technol. Appl. MLMTA ’04*, June 21-24, 2004, Las Vegas, Nevada, USA, pp. 4–8, 2004.

[5] R. Borrell, “A brute force approach to solving the knight’s tour problem using Prolog,” in *Proc. 2009 Int. Conf. Artif. Intell. ICAI* 2009, vol. 2, 2009.

[6] M. M. Ismail, A.F. Zainal Abidin, S. Widiyanto, M.H. Misran, M. Alice, N.A. Nordin, E.F. Shair, S.M. Mustaza, M.N. Shah Zainudin, “Solving Knight’s tour problem using firefly algorithm,” in the P*roc. 3rd Int. Conf. Eng. ICT*, no. April, pp. 5–8, 2012.

[7] W. W. R. Ball, *Mathematical Recreations and Essays*, 12th ed. New York, USA: MacMillan, 1939.

[8] H. E. Dudeney, *Amusements in Mathematics*, Thomson Nelson and Sons Ltd. London, 1917.
[9] A. P. Domoryad, *Mathematical Games and Pastimes*. Oxford, 1964.
[10] S. P. Hurd and D. A. Trautman, "The Knight's Tour on the 15-puzzle," *Math. Mag.*, vol. 66, no. 3, p. 159, Jun. 1993.
[11] K. Lim, "The Knight's tour," *Parabola*, no. 1, UNSW Australia, Sydney, 1981.