**Analysis of Different Approaches in Solving Knight’s Tour Problem  
CS7IS2 Project (2019-2020)**

Aashima Goel, George John Chavady, Githu Jeeva Savy, Piyush Mankad, Soumen Ghosh

[goelaa@tcd.ie](mailto:goelaa@tcd.ie), [chavadyg@tcd.ie](mailto:chavadyg@tcd.ie), [jeevasag@tcd.ie](mailto:jeevasag@tcd.ie), [mankadp@tcd.ie](mailto:mankadp@tcd.ie), [ghoshso@tcd.ie](mailto:ghoshso@tcd.ie)

**Abstract.** The Knight’s tour problem on the chess board is an especial but intriguing tour problem. In this paper we propose an algorithm based on the neural networks and compare its performance against the existing algorithms like depth first search, breadth first search, recursive backtracking and Warnsdorff’s algorithm.

abstract should summarize the contents of the report and should contain at least 70 and at most 150 words. It should be set in 9-point font size and should be inset 1.0 cm from the right and left margins. There should be two blank (10-point) lines before and after the abstract. This document is in the required format. The abstract should give a concise overview of the main points of the report: the motivation behind the work, a very high level description of the problem and how it was solved by the proposed algorithms. The abstract must not include any figures or table.

This document is a guideline for writing the final report for the CS7IS2 module *Artificial Intelligence*. You should follow its general structure as shown below.

You should not change its format (font, size, margin, space, etc.).

Report that not comply to the format or exceed the maximum length will be penalised (-5 marks).

Brevity is desirable in communication, however you should provide all those details necessary for the good understanding of the described methods and algorithms.

The report will be graded on the basis of:

* Originality;
* Technical soundness;
* Organisation;
* Clarity of presentation
* Adequacy of bibliography/Results (this last point strongly depends on the type of report)

**Your report should provide a survey and an experimental comparison of multiple solution approaches to a particular problem. This is a critical review of at least three papers that significantly contributed to advance the state-of-the-art for the problem you are analysing. It should not be a mere summary of the papers. You are expected to conduct an analytical review of the methods under analysis to try to find common aspect and differences, connections between methods, drawbacks and open problems. Unless the faced problem has emerged recently, students should choose their papers by diversifying the range of approaches used to solve the problem. A good guideline could be to choose a paper from a decade or two ago, and a couple of more recent papers. You need to experimentally evaluate approaches in a simulation of a problem, in a range of scenarios, and analyse the pros and cons of each approach.**

**1 Introduction**

In the recent decades, chess board games have become a centre of attraction for many of the efforts involving Artificial Intelligence. The number of chess game engines with the capability of playing chess at a master’s level are over 250. This can be achieved by using certain advanced techniques such as minimax algorithm for best move identification, alpha beta pruning methods for reducing the tree searches etc. [1] Even though these advanced algorithmic developments have boosted the positioning and sorting techniques in chess games, they are no longer near human intelligence.

The Knight’s tour problem is an intriguing old puzzle. It is a special case of NP hard problem of finding a Hamiltonian path on a special graph. It has fascinated chess players, computer scientists and mathematicians for many years. The traversal movements are easy for a king, queen, rook etc. The ‘L’ shaped movement of the Knight offers a unique and difficult puzzle. The main goal of this puzzle is to find an optimum path from the beginning to the end point by traversing all the points on the chess board only once. The sequence of moves is done by following the rules in chess (L shape moves). This problem can also be translated to a graphical problem in which the existence of a Hamiltonian circuit is checked.

There are two types of Knight’s tour problem. The open one and the closed one. If the last square is reachable from the first square by a single knight’s move, it can be called as a closed one. The problem in which every square is visited once and don’t have the ability to return to the origin in one move can be called as an open Knight’s tour. The problem of the closed knight’s tour problem is an instance of the graphical Hamiltonian cycle problem.

There are different approaches in solving the knight’s tour problem using computer-based solutions. We can use either algorithmic solutions or heuristics-based ones. Some of the algorithms which we can use are categorized into Brute force algorithms, divide and conquer methods, Warndorff’s method and solutions based on machine learning and neural networks.

One of the applications of the solution of the Knight’s tour problem is in preserving digital image information security. This can also be used in image encryption schemes for visual cryptography. This encryption is done mainly by dividing the image into 8x8 pixels and then shuffling the pixel values [2].

The organization of this paper is as follows. In section 2, some of the previous works and algorithmic approaches are discussed. The approaches used by us are discussed in section 3. The comparisons of performance evaluations for different algorithms are discussed in Section 4. A conclusion section is added at last.

**2 Related Work**

There are many existing possible solutions for solving the Knight’s tour problem. There are different techniques and approaches used for solving the Knight’s tour. One of the earliest systematic solutions for this problem was contributed by Euler in 1759 [2]. The method proposed by Euler was aimed at the movement of Knight around a chess board till only a few cells are left. To incorporate those cells Euler introduced a set of rules and these are extremely tedious. So it is not easily applicable as a computer algorithm.

German mathematician H C Warnsdorff [6] came up with a solution which uses a simple greedy heuristic approach. He suggested that to always move to an adjacent unvisited square with a minimal degree. So the heuristics determines the next square to be selected as the one with the fewest number of moves rather than using a random selection. The methodology proposed by Warnsdorff doesn’t produce the desired results every time. Also, the rule doesn’t provide a tie breaking rule. Genetic algorithm [7] provides a modern way of implementing the knight’s tour problem. There are also other algorithms like intelligent ant colony optimizations algorithms based on this problem. The most modern technique is the usage of a neural network based solution for the problem. Each possible Knight movement can be represented as a neuron.

In this paper we are comparing the performance of various algorithms like depth first search, breadth first search, recursive backtracking, Warnsdorff’s algorithm and also a novel neural network based solution. The performance evaluation metrics for this problem generated using different algorithms are compared to find the pros and cons of each algorithm and also to find the optimum one.

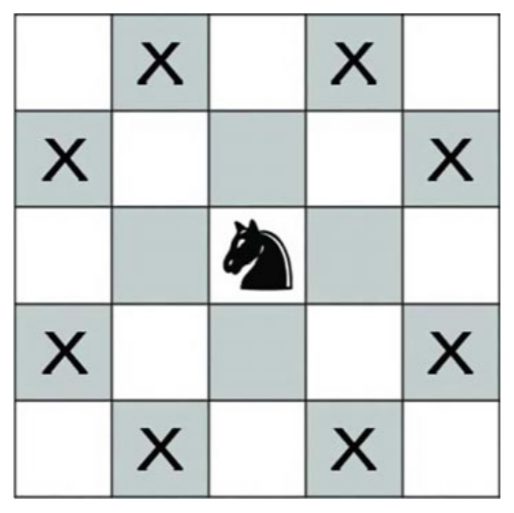
In this section you will discuss possible approaches to solve the problem you are addressing, justifying your choice of the 3 you have selected to evaluate. Also, briefly introduce the approaches you are evaluating with a specific emphasis on differences and similarities to the proposed approach(es).

**3 Problem Definition and Algorithms**

This section formalises the problem you are addressing, and the models used to solve it. This section should provide a technical discussion of the chosen/implemented algorithms. A pseudocode description of the algorithm(s) can also be beneficial to a clear explanation. It is also possible to provide one example that clarifies the way an algorithm works. It is important to highlight in this section the possible parameters involved in the model and their impact, as well as all the implementation choices that can impact the algorithm.

The Knight in a chess board can move only in one of the eight possible directions which is shown in figure 1 [1]. As discussed earlier, the closed Knight’s tour is an example of a Hamiltonian circuit problem. Every square on the chess board will be visited by the Knight only once until it has finished with the 64 squares. After traversing all the squares, we are returning to the start position. Since we arrive at the position where the Knight’s movement started, such a tour can be called a closed Knight’s tour problem.

An open Knight’s tour will be using a Hamiltonian path instead of using a closed Hamiltonian circuit. The main difference between the closed and open Knight’s tour problem is that in an open tour we are not required to get back to the starting square.



**Figure 1: The Eight Possible Knight Moves**

Our focus was to concentrate on the knight’s tour problem on an m x m sized board. Then start from an initial position and then perform the Knight tour traversal on until all the squares are visited.

There are certain conditions to be met for the existence of a Knight’s tour [3]. For a Knight’s tour to exist on an m x n chessboard with m less than or equal to n, following conditions need to be met:

1. m and n both numbers are odd
2. m = 1, 2, or 4
3. m = 3 and n = 4, 6, or 8.

Out of the several approaches used for solving the problem we are comparing the performance of certain algorithms in performing the Knight’s tour on the board.

**3.1 Brute Force Algorithms**

Here we are considering the performance of two off the shelves approaches in solving the problem. We are using both breadth first search and depth first search for the Knight’s tour. In the breadth first search approach the neighbouring nodes at the present depth are explored first prior to the nodes present at the next depth level [4]. In the depth first search the traversal starts from a root node and then proceeds along the same branch as far as possible before back-tracking and switching the branches.

**3.2 Warnsdorff’s Algorithm**

As previously suggested, Warnsdorff gave a simple heuristic to solve the Knight’s tour problem on an 8x8 chessboard. He stated that, in order to select the next move for the Knight, the available moves should satisfy three conditions declared below [6];

1. the position should be adjacent to the current position.
2. the position be unvisited by the Knight
3. the position should have the minimal number of adjacent, unvisited squares

This led to a number of problems when the proposed heuristic ended up in a tie. Warnsdorff suggested that any random move may be chosen to break the tie however experiments [6] show that this rule (breaking ties at random) decreases in effectiveness as the size of the board grows larger.

This problem could be solved by specifying some additional rule to the tie breaking problem rather than solving it randomly. An example of such a solution was given by Paul [8] in his paper, where he solves the problem by fixing the move order (by giving every move an index from 1 to 8) and selecting the first move from the move order shall a tie breaking situation arise. His method produced far greater results than breaking ties at random.

**3.3 Backtracking Algorithm**

Backtracking is a recursive algorithmic approach [5]. This algorithm proceeds to find the solution of a problem by using an incremental approach. Whenever there is a need for a different approach to a problem at one stage, this algorithm tries all the possible combinations of options recursively. By the nature of the Knight’s tour problem, a backtracking algorithm can provide an efficient traversal. But the naive backtracking algorithm can be very slow even in powerful computers due to the number of traversals required. This is where the advantage of Warnsdorff’s algorithm comes into picture as it uses simple heuristics and thereby reduces the complexity involved. A Knight’s tour on a chessboard has different options at different stages during the tour. Sometimes, some options which we choose might lead us to a dead end and then we have to start again by recursive backtracking and choose some alternate options.

**3.4 Neural Networks**

**4 Experimental Results**

This section should provide the details of the evaluation. Specifically:

* Methodology: describe the evaluation criteria, the data used during the evaluation, and the methodology followed to perform the evaluation.
* Results: present the results of the experimental evaluation. Graphical data and tables are two common ways to present the results. Also, a comparison with a baseline should be provided.
* Discussion: discuss the implication of the results of the proposed algorithms/models. What are the weakness/strengths of the method(s) compared with the other methods/baseline?

**5 Conclusions**

Provide a final discussion of the main results and conclusions of the report. Comment on the lesson learnt and possible improvements.

A standard and well formatted bibliography of papers cited in the report. For example:

**References**

1. Escalante, R.G. and Malki, H.A., 2006, July. Comparison of artificial neural network architectures and training algorithms for solving the knight's tours. In The 2006 IEEE International Joint Conference on Neural Network Proceedings (pp. 1447-1450). IEEE.
2. Singh, M., Kakkar, A. and Singh, M., 2015. Image Encryption Scheme Based on Knight's Tour Problem. Procedia Computer Science, 70, pp.245-250.
3. Schwenk, A.J., 1991. Which rectangular chessboards have a knight's tour?. Mathematics Magazine, 64(5), pp.325-332.
4. Cormen, T.H., 1990. Leiserson, Charles E.-Rivest, Ronald L. Introduction to algorithms.
5. Ghosh, D. and Bhaduri, U., 2017, September. A simple recursive backtracking algorithm for knight's tours puzzle on standard 8× 8 chessboard. In *2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI)* (pp. 1195-1200). IEEE.
6. Squirrel, D. and Çull, P., 1996. A Warnsdorff-Rule Algorithm for Knight's Tours.
7. Al-Gharaibeh, J., Qawagneh, Z. and Al-Zahawi, H., 2007. Genetic Algorithms with Heuristic-Knight's Tour Problem. In *GEM* (pp. 177-181).
8. Pohl, Ira. (1967). A method for finding Hamilton paths and Knight's tours. Commun. ACM. 10. 446-449. 10.1145/363427.363463.