Knight Quest: A Geometric Heuristic Approach to Knight Pathfinding

Aleksandar Gladović

alsocial5371@gmail.com

July 6, 2025

Contents

1. Introduction

This article presents a novel, geometry-inspired solution to the knight's shortest path problem by exploiting the emerging modularity and rotational symmetry of L-shaped knight moves. Rather than relying solely on traditional search algorithms, this method models the knight's movements as base vectors in the complex plane, specifically using the canonical vectors (2,1) and (1,2) represented as $\overrightarrow{u} = 2 + i$ and $\overrightarrow{v} = 1 + 2i$. These vectors are then rotated and mirrored according to the signed distance vector between the starting point $A_{x,y}$ and the destination $B_{x,y}$, allowing for a modular decomposition of the knight's path. The solution operates by transforming the pathfinding problem into one of rotational alignment and mirrored symmetry, with strategic reference points $R_{x,y}$ used to identify optimal paths across different directional regions. By analyzing the periodic structure of knight moves—especially across diagonals like $y = x$ and lines $y = \frac{x}{2}$ the algorithm leverages the repeating alternation in movement patterns to reduce the problem to a combination of sequence evaluation and rotation-based projection. The remainder of the article will formalize this model and define its evaluation metrics.

For Implementation details in python, log files, testing and comparison against BFS check the Github repository.

2. Problem statement

Find the minimum number of moves for a knight to reach a square on the grid. Optional, construct a path of knight moves from starting to final square on the grid.

3. Preliminaries

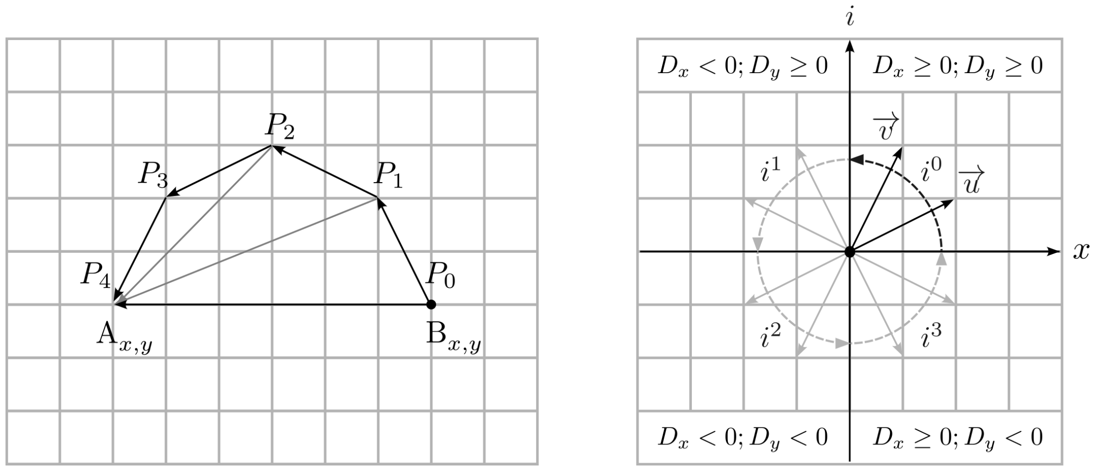

Figure 1: Path construction

Let: $A_{x,y}$ be the starting position and $B_{x,y}$ the final position. A_x, B_x denotes positions on x-axis; A_y, B_y denotes positions on y-axis; where $x, y \in \mathbb{Z}$. $P_{i,j} = (p_0, p_1, ..., p_n)$ is a sequence of path segments where $0 \leq i \leq n, j \in \{x, y\}$ and $p_0 = \{B_x, B_y\}$. Then $f_{seg}(P_i, K)$ is a function that modifies a segment:

$$f_{seg}(P_i, K) = \{P_{i,x} + K_x, P_{i,y} + K_y\}$$

Let: $D_{x,y}$ be the signed distance vector in the x and y directions from the path segment P_i to the starting position A.

$$D_{x,y} = f_{dist}(P_i) = \{A_x - P_{i,x}, A_y - P_{i,y}\}$$

Then $\overrightarrow{u} = 2 + i$, $\overrightarrow{v} = 1 + 2i$ are knight base vectors in the first quadrant, where $u, v \in \mathbb{C}$. $U_{x,y}$ and $V_{x,y}$ are real parts of 90° counterclockwise rotation of those vectors representing knight movement. $\Delta\theta$ is the rotation scalar of the 90° angle respectively of D.

$$U = f_{rot}(u, \Delta\theta) = \left\{\Re(u \cdot i^{\Delta\theta}), \Im(u \cdot i^{\Delta\theta})\right\}$$
$$V = f_{rot}(v, \Delta\theta) = \left\{\Re(v \cdot i^{\Delta\theta}), \Im(v \cdot i^{\Delta\theta})\right\}$$

Quadrant rotations and adjacent square adjustment

$$\Delta\theta = \left| \begin{array}{lll} 0 & \text{if } D_x \geq 0 \text{ and} & D_y \geq 0 \\ 3 & \text{if } D_x \geq 0 \text{ and} & D_y < 0 \\ 1 & \text{if } D_x < 0 \text{ and} & D_y \geq 0 \\ 2 & \text{if } D_x < 0 \text{ and} & D_y < 0 \end{array} \right| \qquad |D_{x,y}| = 1 \rightarrow \Delta\theta = \Delta\theta + 1$$

4. Reference point

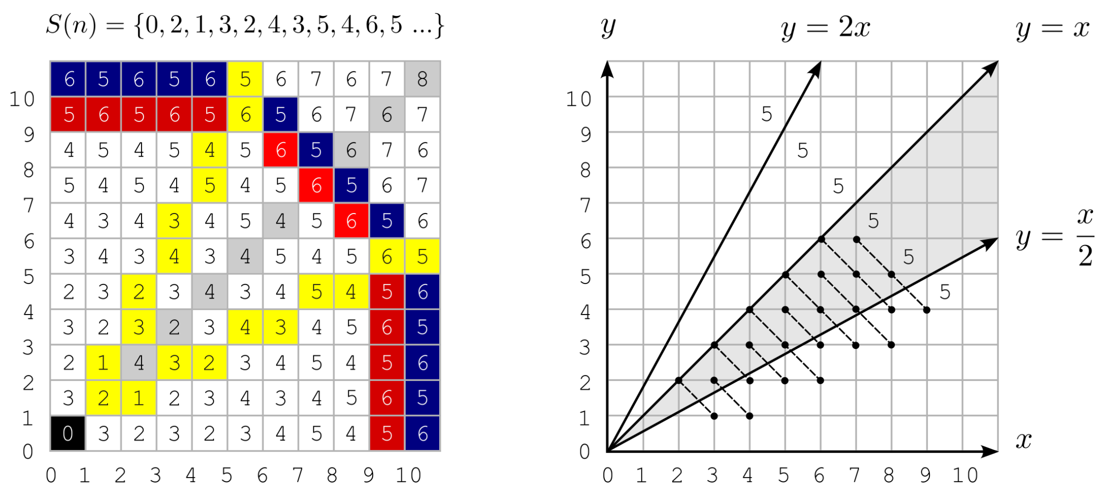

Figure 2: Reference point projection

Knight moves are mirrored across the main diagonals: $y = x, y = -x, -y = x$ and $-y = -x$. They exhibit repeating patterns along perpendicular lines, such as from $y = x$ to $y = \frac{x}{2}$ or $y = 2x$, where the movement alternates either vertically (x to $y = \frac{x}{2}$) or horizontally (y to $y = 2x$). By analyzing sequences of moves along the $y = \frac{x}{2}$ line, we can determine the minimum number of moves required to reach a target square.

Let $R_{x,y}$ be a reference point between P_i and A, independent of orientation or direction, due to the mirroring property of knight movement.

$$R_{x,y} = f_{ref}(D) = \{\max(|D_x|, |D_y|), \min(|D_x|, |D_y|)\}$$

To determine the minimum number of moves to reach this point, if it lies within the region between $y = x$ and $y = \frac{x}{2}$, we project it into the region below $y = \frac{x}{2}$ by determining how far it must be moved diagonally. Such that:

$$y - \Delta \leq \left\lfloor \frac{x + \Delta}{2} \right\rfloor \quad \text{for } \ \Delta \in \mathbb{N}$$

$$y - \Delta \leq \frac{x + \Delta}{2} \Rightarrow 2y - 2\Delta \leq x + \Delta \Rightarrow 2y - x \leq 3\Delta \Rightarrow \Delta \geq \frac{2y - x}{3}$$

$$\text{For first integer this is true: } \Delta = \left\lceil \frac{2y - x}{3} \right\rceil$$

Then to project the reference point R into the region of space bellow $y = \frac{x}{2}$:

$$R'_{x,y} = \{R_x + \Delta, R_y - \Delta\}$$

5. Evaluation function

Let n be the index of the sequence of moves along the line $y = \frac{x}{2}$, and let m be the index of the sequence of moves along the y-axis. Then, we can define n and m in terms of the reference point R and its projection:

$$n, m = f_{term}(R) = \left\{ \begin{array}{lll} n = R_x; & m = \left\lceil \frac{n}{2} \right\rceil - R_y; & R_y < \frac{R_x}{2} \\ n = R'_x; & m = \left\lceil \frac{n}{2} \right\rceil - R'_y; & R_y \geq \frac{R_x}{2} \\ n = 5; & m = 0; & R_x = R_y = 2 \end{array} \right\}$$

Let $S_1(n) = (s_0, s_1, \ldots, s_n)$ be a sequence of moves along the line $y = \frac{x}{2}$. Let $S_2(n, m) = (s_0, s_1, \ldots, s_n)$ be the alternating sequence derived from the values of $S_1(n)$, where $s_n \in \mathbb{W}$. Let $p(a)$ be a parity function. Then, we define the sequence values as follows:

$$p(a) = a \bmod 2 \qquad S_1(n) = \frac{n + 3p(n)}{2}$$

$$S_2(n, m) = S(n) - [\ p(n) \cdot p(m)\ ] + [\ p(n-1) \cdot p(m)\ ]$$

Therefore, to find the minimum number of moves for a knight to reach a square on the board based on the n, m terms we define the evaluation function as:

$$f_{eval}(n, m) = \left\{ \begin{array}{cc} S_2(n, m) & n, m > 0 \\ 3 & n = m = 1 \end{array} \right\}$$

6. Solution

To evaluate the moves and construct a path calculate the initial D, R and n, m:

$$D = f_{dist}(P_0) \quad R = f_{ref}(D) \quad n, m = f_{term}(R)$$

$f_{eval}(n, m)$ finds the minimum number of moves to reach a square on the board:

$$f_{eval}(n, m) = \left\{ \begin{array}{cc} S_2(n, m) & n, m > 0 \\ 3 & n = m = 1 \end{array} \right\}$$

$f_{move}(D)$ applies a move based on the distance and direction to the target square:

$$f_{move}(D) = \left\{ \begin{array}{ll} D; & \max(D_x, D_y) \; / \; \min(D_x, D_y) = 2 \\ U; & f_{eval}(n, m) - f_{eval}(f_{term}(f_{ref}(R_x - 2, R_y - 1))) = 1 \\ V; & f_{eval}(n, m) - f_{eval}(f_{term}(f_{ref}(R_x - 1, R_y - 2))) = 1 \end{array} \right\}$$

$f_{path}(P)$ recursively constructs a path from a sequence of valid path segments:

$$f_{path}(P) = f_{path}(P \cup f_{seg}(P_{last}, f_{move}(f_{dist}(P_{last})))) \text{ while } f_{eval}(n, m) > 0$$

7. Empirical results

To benchmark performance, tested both the KnightQuest algorithm and a classical brute-force BFS approach on $1,000$ randomly generated knight pathfinding cases within the coordinate range $(-500, 500)$. On average, BFS required 26.6 seconds per case, totaling nearly 7.5 hours of computation. In contrast, KnightQuest averaged just 4.7 milliseconds per case and completed the entire batch in under 5 seconds, achieving an average speedup of $4,600$. Additionally, the $f_{eval}()$ heuristic function yielded instant move estimations with a 4,700,000$\times$ speedup over BFS. No fallbacks were triggered, and KnightQuest succeeded in 100% of cases.

Test Summary	
Tested over the range:	(-500, 500)
Seed:	42
Total test runs:	1000
Failed tests:	0
Total fallback moves used:	0
Average KQ path time:	0.00478579s
Average BFS path time:	26.76225433s
Average feval time:	0.00000585s
Average KQ speedup:	4616.54x
Average feval speedup:	4781206.19x