Efficient State Enumeration in a Torus Grid World: Algorithms and Analysis

Vasil Dakov

Abstract

We address the problem of state enumeration in a torus grid world with unknown dimensions and starting state. The goal is to ensure that every state is visited while maintaining an algorithmic complexity of $O(35S)$, where $S = w \cdot h$ is the area of the torus. We present and analyze a baseline spiral algorithm and an improved approach leveraging irrational slopes to ensure termination and efficiency. Theoretical guarantees and empirical observations are provided to validate the proposed solutions.

1 Problem Overview

We define the problem as a search task in a $w \times h$ torus grid world. The state space is defined as:

$$\mathcal{S} = \{(i, j) \mid \forall i, j : 0 \leq i < w, 0 \leq j < h\}.$$

The agent has no prior knowledge of the world's dimensions or its initial state. The action space is given by:

$$\mathcal{A} = \{(i, (j+1) \mod h), (i, (j-1) \mod h), ((i+1) \mod w, j), ((i-1) \mod w, j)\}.$$

The objective is to reach a goal state, which can be any state in $\mathcal{S}$. Without loss of generality, the problem can be reformulated as visiting all states in $\mathcal{S}$, leveraging the torus's wrap-around property.

Several mathematical properties of our environment and solution must be defined to proceed.

We can reformulate the problem as visiting all possible spaces on the torus. This is due to the fact that we have no knowledge of the environment and the goal can be on any single state. This reformulation gives the freedom to start from an arbitrary location- starting location is irrelevant for the torus' enumeration due to its wrap-around.

2 Solution

To be sure all spots on the torus are visited, we must ensure our algorithm never enters a periodic cycle prior to enumerating the whole torus. Otherwise, that would leave edge cases with some unenumberated state(s). For example, a simple diagonal policy ($right \rightarrow down \rightarrow right \rightarrow down$) would enter a cycle on 4×4 grid.

Baseline solution

The simplest solution with a guarantee of enumerating all states is a spiral outward. It only requires knowledge of the initial state. In all cases, it would steadily grow toward unvisited states, assuring eventually all states are visited (and consequently the goal state). Pseudocode provided in 1 along with a visualization in Figure 1. Unfortunately its runtime comlexity is quadratic with respect to the larger side of the grid: $\mathbf{O}(\max(w,h)^2)$, as the spiral will always enumerate a square with sides equal to the larger side. In many cases this violates our $\mathbf{O}(35 \cdot S)$ restriction. For example, for $w = 1, h = 10^6$, the algorithm would do 10^{12} steps or $S \cdot 10^6 S$

Algorithm 1 Spiral Movement Policy

Require: Grid dimensions, current position (nx, ny), and global variables: `counter`, `step_size`, `direction`.
Ensure: Next position (nx, ny).

Extract nx, ny from the current position.
if `direction` $= 0$ **then** ▷ Move right
    $nx \leftarrow (nx + 1)$ mod `width`
else if `direction` $= 1$ **then** ▷ Move down
    $ny \leftarrow (ny - 1)$ mod `height`
else if `direction` $= 2$ **then** ▷ Move left
    $nx \leftarrow (nx - 1)$ mod `width`
else if `direction` $= 3$ **then** ▷ Move up
    $ny \leftarrow (ny + 1)$ mod `height`
end if
`counter` $\leftarrow$ `counter` $+1$
if `counter` $=$ `step_size` **then**
    `counter` $\leftarrow 0$
    `direction` $\leftarrow$ (`direction` $+ 1$) mod 4
    if `direction` is even **then** ▷ Increase step size after right/down or left/up
        `step_size` $\leftarrow$ `step_size` $+ 1$
    end if
end if
return (nx, ny)

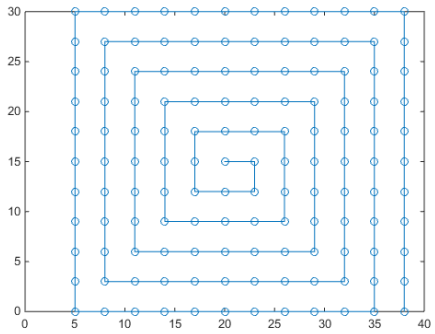

Figure 1: How the spiral algorithm looks like. It is visible how it grows outward and would eventually cover the entire grid. Image credit: https://stackoverflow.com/questions/29466058/spiral-loop-on-a-matrix-from-a-point

Improved Solution

My solution greatly improves on edge cases like the one in the baseline solution, while staying linear. It is a diagonal algorithm, iterating in the $x-$ and $y-$ axes by taking steps along an irrational slope with steps sizes $\Delta x = (\sqrt{5} - 1)/2$ and $\Delta y = 1$. It guarantees that it enumerates the grid due to the mathematical properties of irrational numbers, low-discrepancy sequences and the torus itself. Pseudocode provided in 2.

Why this solution works

Here is a brief overview of the properties that guarantee the effectiveness and termination of my solution.

Guarantee of termination: We can use the irrational winding of the torus theorem to guarantee that all spots on the torus are covered. It states that, for an n-dimensional torus, a line with an irrational slope is dense on the torus (or covers all points on it when traced for long enough)[1]. Conversely, a rational slope forms a closed line.

Our environment torus is slightly different as it is discrete. As such, we are doing an approximation of the irrational winding of the torus, by iterating in both directions and rounding up to the nearest index. The slope is assured to be irrational, as $\Delta x = (\sqrt{5}-1)/2$ is irrational - Δy does not need to be irrational to satisfy this and empirical experiements showed $\Delta y = 1$ provides faster termination. Intuitively, we have an irrational slope, and irrational numbers have the property of being aperiodic, meaning with enough iterations a solution will always be found.

Efficiency:

The algorithm empirically has shown to perform much better than the baseline solution, and also has the theoretical properties to back it up. Iterating irrational steps additively gives what is called a *low-discrepancy sequence* [3]. It is an approximation of how well all sizes of that subsequence fall into an equidistributed (uniformly distributed) sequence. In other words, it is a measure of the dissimilarity with a uniform distribution. For irrational numbers, and large sequences, it is proven that the discrepancy D tends to $\frac{1}{N}$ where N is the number of elements in the sequence. For our problem, this is limited to $\frac{1}{35 \cdot S}$. If we assume to have achieved this low discrepancy, we can restate the problem as a covering of the unit circle via a uniform distribution. This is equivalent to the coupon collector's problem from probability theory [2]. It gives us an upper bound of the number of steps we may need to take to enumerate the unit circle.

This upper bound is $E(T) = S \cdot H_S = S \log S + \gamma S + \frac{1}{2} + O(1/S)$ where T is the number of steps expected to cover the square, S is the area of the grid, H_S is the Sth harmonic number and $\gamma \approx 0.577215...$ or the Euler-Mascheroni constant [2].

Naturally, there is clumsiness here due to our assumptions and the the discrete nature of our torus, and for how large the N needs to be to ensure low discrepancy. Empirically, however, the approximation seems to be good enough, results ranged between $[1 \cdot S, 18 \cdot S]$, well in the limits. . Some edge cases might be possible, however, violating the $O(35 \cdot S)$ constraint. This is also dependent on the step-size, where the golden ration I used has better theoretical properties.

Algorithm 2 Irrational Movement Policy

Require: Grid dimensions, current position (nx, ny), goal position, and global variable: `counter`.

Ensure: Next position (nx, ny).

Extract nx, ny from the current position.
$\Delta_x \leftarrow (\sqrt{5} - 1)/2$ ▷ Golden ratio adjustment
$\Delta_y \leftarrow 1$
if `counter` $= 0$ **then** ▷ Adjust x coordinate
    $nx \leftarrow (nx + \Delta_x)$ mod `width`
else if `counter` $= 1$ **then** ▷ Adjust y coordinate
    $ny \leftarrow (ny + \Delta_y)$ mod `height`
end if
`counter` $\leftarrow$ (`counter` $+ 1$) mod 2
return (nx, ny)

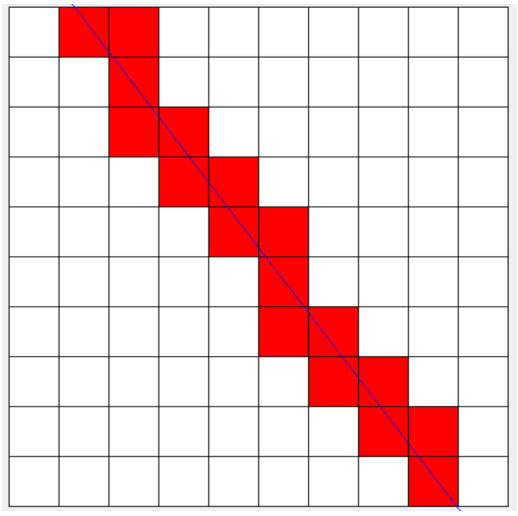

Figure 2: Approximate vissualization of the irrational grid policy. Image credit: https://i.sstatic.net/MIt2B.png

Various grid sizes were tested, symmetric, non symmetric, heavily skewed, and such to ensure all properties were tested. Some examples are: (3, 3), (4, 5), (9, 9), (100, 100), (1000, 1000), (1, 10), (1, 1000), (1000, 1), (1, 1000000), (500, 500), (250, 400), (50, 50), (25, 40), (2, 5000).

3 Alternatives considered

Creating heuristics was an alternative considered for this task. Most predominantly, Euclidean distance was tested modularly, to ensure admissibility and consistency on the sequence created. As expected, it performs optimally and finds the shortest part of the goal each time. In general, all heuristics seemed to give away some spatial information. This was deemed "cheating" and violating the problem conditions, however, as we were giving the agent information on the location of the goal.

References

[1] Patrick Iglesias-Zemmour. The irrational toruses.

[2] Wikipedia. Coupon collector's problem.

[3] Wikipedia. Low-discrepancy sequence.