

Quantum Algorithm to Solve a Maze: Converting Maze Problem into Search Problem

Niraj Kumar and Debabrata Goswami*

Indian Institute of Technology Kanpur, Kanpur – 208016, UP, India

*Email: dgoswami@iitk.ac.in

Abstract

We propose a quantum methodology to approach a Maze problem, which is otherwise a 'NP complete' problem. We convert the problem of solving a classical maze into a quantum search algorithm in order to apply the Grover search algorithm. We show that it is possible to construct an algorithm that can transform a two dimensional perfect maze into the quantum framework. Using the power of Quantum Computation, i.e., Superposition and Entanglement, we show that it is possible to encode the entire combination of solutions into one register. An Oracle can be designed that would also mark specific entities from the population while finding the solution of the maze problem.

Introduction

A maze is a complex system of passages or paths separated by walls or hedges. It is a common tree-branching shape that is a physical situation in which it is easy to get lost. Each line of the maze representation has one input, and two outputs, until we reach the end of the “maze” in which there is no further output. If this rule is followed without any exceptions we get a perfect maze, as any violations of it would cause utter chaos in a world where chaos is enough of a problem already. This action of one input, and two outputs, causes a huge problem for classical computers. Because

it doubles in size each time, the problem takes exponentially more time to solve as the number of decisions increases. For example, a 100 step binary maze would have 2^{100} decisions at the end of the maze, a number too large to be easily dealt with. Thus a typical maze problem is a NP complete problem, which could utilize the ‘quantum parallelism’ of quantum computers that allows the computer to be in multiple states at the same time. One of the problems in the application of quantum computers have been the design of quantum algorithm.

We propose that by devising an appropriate quantum algorithm, a quantum computer can try all the paths of a binary maze at one time, and dramatically reduce the time needed to solve the maze problem.

The Approach

In our particular approach of solving for the possible path of a maze lies in the aspect of the aspect of maze design and its mapping into the quantum domain. The maze is a $n \times m$ two dimensional pathway where n is the number of rows and m being the number of columns. The maze creation algorithm uses a recursive backtracker algorithm [1]. Analogous to the earlier line definition, for a 2D maze, each grid in the maze is called a Room and a square $m \times m$ maze is constructed, starting with an initial empty stack. In our maze, all of the inputs except the last Room must have two connections to the other Rooms.

The first step in the maze is that with a maze of N -decisions, there will be a need to have $(\log_2 N + N)$ qubits in the computer. The reason $\log_2 N$ qubits are used for the number of counting qubits is that because with N decisions, $\log_2 N$ binary digits are needed to keep room for every possible number [2]. Therefore, second step of the algorithm, reads the first $\log_2 N$ qubits (the counting qubits), and prepares to move to that position. After reading the counting qubits and determining our position, we move to that qubit and perform our Hadamard transformation on it. At each

decision, we have a choice of going up or going down, and that is represented with a 1 or a 0. The advantage of the quantum computer is that we can simultaneously travel up and down at the same time [3]. Along those lines, the 1s and 0s of the qubit represent each possible decision. The third step puts the qubit into a superposition of up and down at the point that the second step declared, which is the Hadamard transformation. The fourth step is the simplest of all the steps, which changes the counting qubits by one, so that the next gate knows to move along in the maze. That is the small clause that the extra $\log_2 N$ qubits allows us to do. Otherwise, it would be a guessing game as to what qubits to modify. The fifth step tells us to repeat the process described in the second, third and fourth steps until we reach the “end” of the maze.

The end of the maze is the last row in the maze, in which there are no decision to be made. We have just traversed through N decisions, and being composed of binary decisions, there are 2^N possibilities at the end of the maze. The sixth step completes the algorithm. Of the many points at the end of the maze, the creator determines one point to be the end. It is at this point that we look at the value of the qubit that is in the computer. The order of the 1s and 0s in the qubit tells us the path that leads to the corresponding endpoint. At first glance, this seems silly, as knowing the end point of the maze is what appears we are trying to find. But, if we try to solve a normal maze, we are always given a beginning and an ending point, and our job is to find the path in between. The only limitation lies with the capability of the quantum search algorithm, the best one being used by us is the Grover’s algorithm. Subsequent search algorithms, with power more than that of the Grover’s algorithm, would boost the approach towards the optimum solution.

Acknowledgement: NK thanks the SURGE program of the IIT Kanpur and DG thanks MHRD Virtual Labs Funding programs.

6. REFERENCES

1. H.A. Priestley and M.P. Ward, “A Multipurpose Backtracking Algorithm”, *J. Symb. Comput.* **18** (1994) 1-40.
2. H.F. Legard, P.E. Mcquaid, A. Singer, *From Baker Street to Binary: An Introduction to Computers and Computer Programming* (New York: McGraw-Hill, 1983).
3. Quantum Computation and Quantum Information, Nielsen and Chuang. M.A. Nielsen, I. Chuang, *Quantum Computation and Quantum Information* (New York: Cambridge University Press, 2000).