**A-I Literary Review: Maze Generation and AI Solutions**

Solving mazes using artificial intelligence (AI) is interesting and fun in a game setting, but it is also becoming increasingly important and relevant as it can be applied to things like navigation in the real world and is a great way to test the capabilities of different algorithms. Using AI to solve mazes can be accomplished using a variety of algorithms to navigate the state space of the maze from the start state (starting point) to the goal state (maze exit). There are quite a few steps required to solve a maze using AI, and each step must be realized in the correct order. Before any algorithms can be applied, there must be a maze. One common maze generation technique is the classic spanning tree algorithm. Another more cutting-edge technique is map generations is image based. Following the creation of the map, it is crucial to determine the algorithms that will be used to solve it. There are lots of different methods to choose from, for example some we have discussed during this course like bidirectional search, best first search, breadth first search, and depth first search to name a few. While not limited to maze solving, a more cutting-edge search practice is quantum search which makes use of black box algorithms. This method is more complicated and black box algorithms pose their own unique challenges. In this literature review, the focus will be exploring current maze generation practices and how AI algorithms are being used to solve said mazes.

**Maze Generation:**

In regard to maze generation, lots of techniques are based on the classic spanning tree. This method connects all vertices within the tree. One grid can contain more than one spanning tree, however if all nodes are not connected, it is not a spanning tree. Nowadays, for things like video games, it is very important for developers to be able to customize the layout. As a result, developers have begun taking an approach focused on the design of the map, using more input and less automation (Kim, et. al.). A newer practice generates mazes using an image provided by the developer or designer. This is achieved by manually defining the layout of the maze complete with style parameters within the image. The image is then uploaded into the system which then uses existing maze making programs to generate the maze based on the image. When using image-based maze generation, the desired solution can be provided, but is not required. While the methods based on the spanning tree are more automated, the image-based method allows the developer to turn an existing image into a maze (Xu et al.). This allows for high levels of customization for developers and designers, as they are able to create a maze by hand exactly how they want it to look. For example, a picture of a real-life mountain can be partitioned into sections for the maze, then uploaded into the system which applies the maze generation algorithms to create a maze out of a real mountain.

**Algorithms and Solutions:**

There are a multitude of options to search a maze for the goal state. To briefly discuss a few algorithms covered in the course: Bidirectional Search starts at the start state and goal state and works to meet in the middle, Breadth First expands all nodes on a “level” before moving to the next, Depth First expands all nodes in a branch before moving to the next, and Best First explores the most promising nodes based on the heuristic. Quantum search, which applies black box algorithms, is not quite as straightforward. Black box algorithms are difficult because as the name implies, some of the methodologies used to reach the conclusion are unknown. Additionally, this method is not very transparent (Rivers, Tauritz). As a result, it can be difficult to understand how the algorithm reached the goal, which can make the decision making difficult to trust. On the other hand, these highly specialized algorithms typically outperform more generic algorithms at the tasks they are designed to do. While the increased performance is a great benefit, the lack of transparency leads to an interesting tradeoff. The method in question, quantum search uses Grover’s algorithm which uses superposition and phase interference to locate the desired item in a disordered list (Oliviera et. al). Grover’s algorithm increases computational speed quadratically. By combining elements of classical computation with quantum computation, researchers are seeking to determine how this affects computation speed and looking to maximize efficiency (Ansis Rosmanis). No matter what algorithm is applied to the maze, a solution is reached when the maze has been successfully completed. Depending on the map, there could be only one solution or a few different ways to reach the goal. Upon successful completion of the maze, the results can be compared and scored to determine which method performed the best.

**Evaluation and Application:**

After successfully solving the maze, the results from the different algorithms can be compared based on certain criteria or performance metrics. There are a few ways this can be accomplished. First is by counting expanded nodes. The algorithm that expands the least nodes takes the shortest and therefore most efficient path. Another way is based on time. Ignoring node expansion, whichever algorithm solved the maze in the shortest amount of time performed the best. Using these metrics, it is assumed the algorithms are all solving the same, static maze. Additionally, there is the Similarly Constructed Maze Problem (SCMP) and Differently Constructed Maze Problem (DCMP) methods. These methods compare the performance of different algorithms on multiple different mazes- all the mazes are different, however SCMP uses mazes with the same connectivity, while DCMP uses mazes with different connectivity’s. Four of the ten mazes are used for training, while the other six are used for test. The amount of mazes successfully completed by each algorithm is then totaled and can be compared (Alaguna, Gomez). When it comes to application, there are tons of uses for maze solving algorithms. One everyday use is GPS; getting from one location to another in the most efficient manner given a starting point and a desired end point. Additionally, maze navigation can be used in a variety of ways when it comes to robots in the workforce. One potential option are robots operating in a static working environment with one standard maze that is functionally a map. Additionally, maze solving algorithms can be applied in real time in robots working in a dynamic environment navigating their area as a maze, reacting to obstacles as they are encountered. There are many more applications outside of those mentioned, as well.

**Conclusion:**

Clearly maze solving algorithms are very relevant for daily problems, and there is lots of cutting-edge research being done on maze generation techniques, search algorithms, and performance evaluation. Image based maze generation is an up-and-coming alternative to the traditional automated spanning tree method which requires a little more work but allows lots more freedom. State of the art algorithms like quantum search are making use of black box algorithms which are useful and efficient, but leave some things to be desired, for example transparency. There is always some give and take, so it is crucial to determine the goals and desires of the project beforehand to allow for proper selection of algorithms.

**Bibliography:**

Camilo Alaguna and Jonatan Gomez. 2018. Maze benchmark for testing evolutionary algorithms. In Proceedings of the Genetic and Evolutionary Computation Conference Companion (GECCO '18). Association for Computing Machinery, New York, NY, USA, 1321–1328. <https://doi.org/10.1145/3205651.3208285>

Paul Hyunjin Kim, Jacob Grove, Skylar Wurster, and Roger Crawfis. 2019. Design-centric maze generation. In Proceedings of the 14th International Conference on the Foundations of Digital Games (FDG '19). Association for Computing Machinery, New York, NY, USA, Article 83, 1–9. <https://doi.org/10.1145/3337722.3341854>

Kim, P. H. (2019). Intelligent Maze Generation. In ProQuest Dissertations and Theses Global. ProQuest LLC.

Ansis Rosmanis. 2024. Hybrid Quantum-classical Search Algorithms. ACM Transactions on Quantum Computing 5, 2, Article 8 (June 2024), 18 pages. <https://doi.org/10.1145/3648573>

Ivan S. Oliveira, Tito J. Bonagamba, Roberto S. Sarthour, Jair C.C. Freitas, Eduardo R. deAzevedo, Brief Historical Survey and Perspectives, Editor(s): Ivan S. Oliveira, Tito J. Bonagamba, Roberto S. Sarthour, Jair C.C. Freitas, Eduardo R. deAzevedo, NMR Quantum Information Processing, Elsevier Science B.V., 2007, Pages 1-7, ISBN 9780444527820, <https://doi.org/10.1016/B978-044452782-0/50002-6>. (https://www.sciencedirect.com/science/article/pii/B9780444527820500026)

Rebecca Rivers and Daniel R. Tauritz. 2013. Evolving black-box search algorithms employing genetic programming. In Proceedings of the 15th annual conference companion on Genetic and evolutionary computation (GECCO '13 Companion). Association for Computing Machinery, New York, NY, USA, 1497–1504. <https://doi.org/10.1145/2464576.2482728>

Jie Xu and Craig S. Kaplan. 2007. Image-guided maze construction. ACM Trans. Graph. 26, 3 (July 2007), 29–es. https://doi.org/10.1145/1276377.1276414