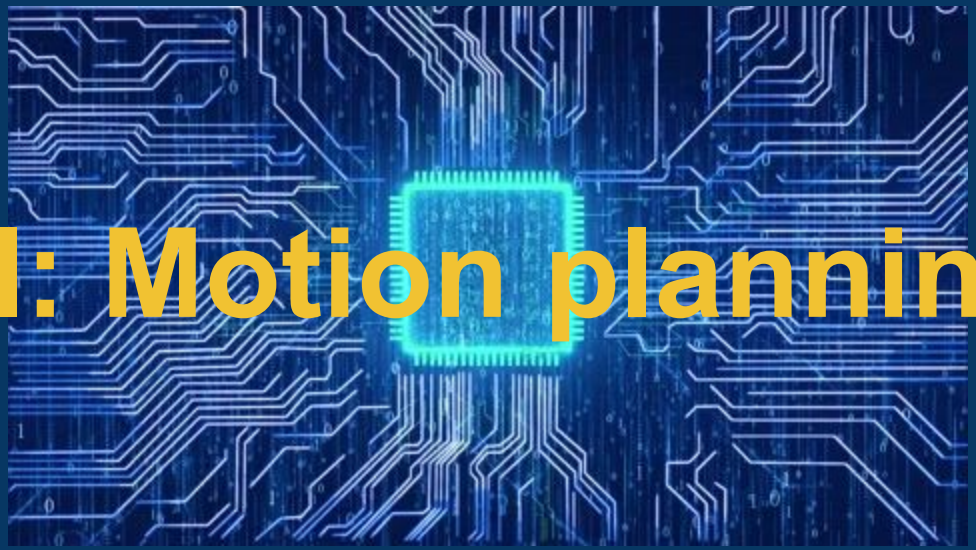
AI: Motion planning



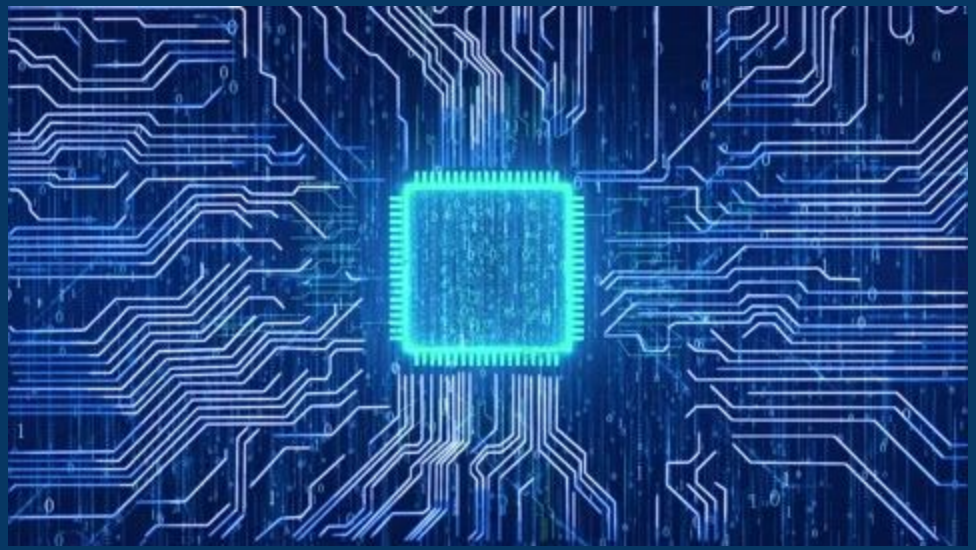
By Cheyenne Tanner

When is this used?

a) Video games

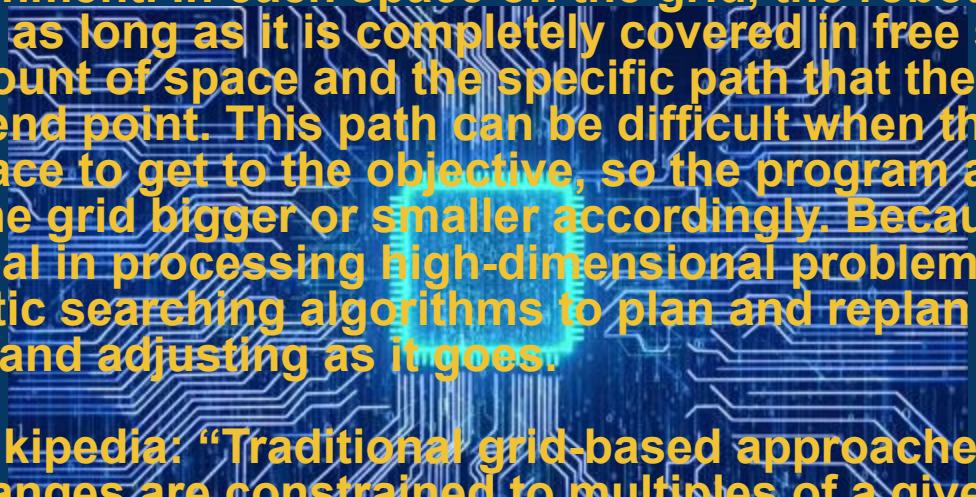
b) robotics

c) Multidimensional data analysis

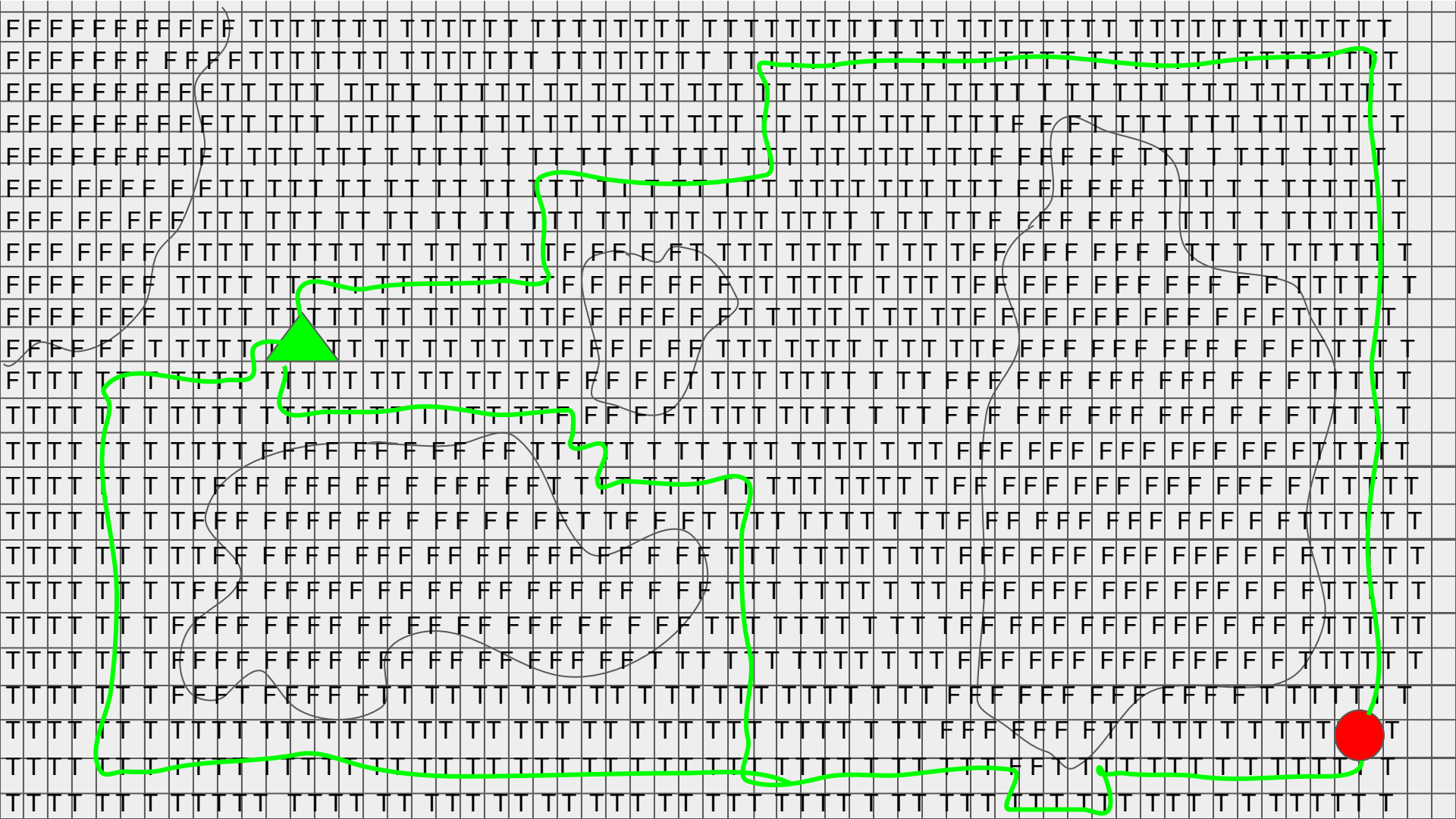


Grid-based search

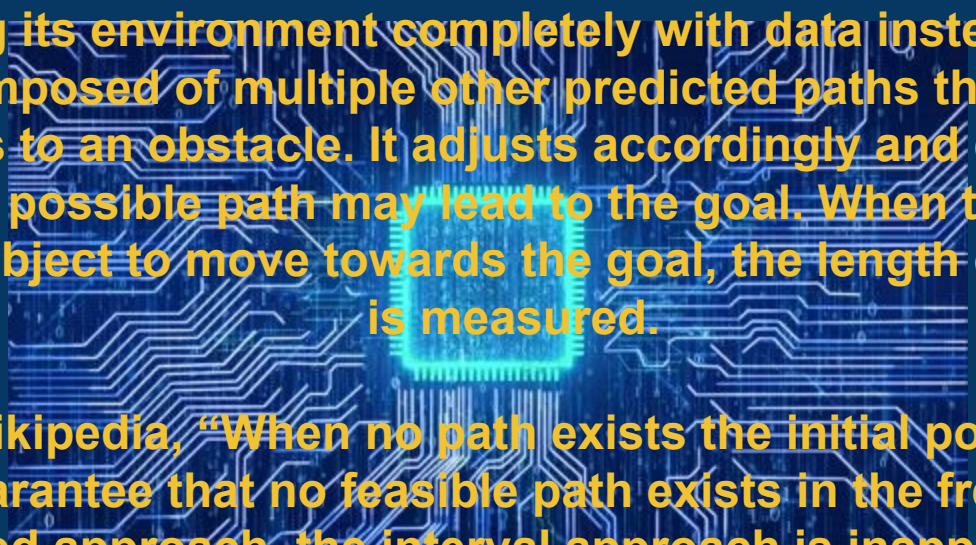
This method lays a grid on its environment and associates its obstacles with certain spaces and points of the grid. It also adjusts the dimension of the grid based on its environment. In each space on the grid, the robot is allowed to move amongst the space as long as it is completely covered in free space. This determines the amount of space and the specific path that the robot can make in order to get to it's end point. This path can be difficult when there are smaller portions of free space to get to the objective, so the program adjusts and makes the dimension of the grid bigger or smaller accordingly. Because of this try-fail method, it's not ideal in processing high-dimensional problems. It used incremental heuristic searching algorithms to plan and replan paths to the objective, learning and adjusting as it goes.



According to wikipedia: "Traditional grid-based approaches produce paths whose heading changes are constrained to multiples of a given base angle, often resulting in suboptimal paths. Any-angle path planning approaches find shorter paths by propagating information along grid edges (to search fast) without constraining their paths to grid edges (to find short paths)."

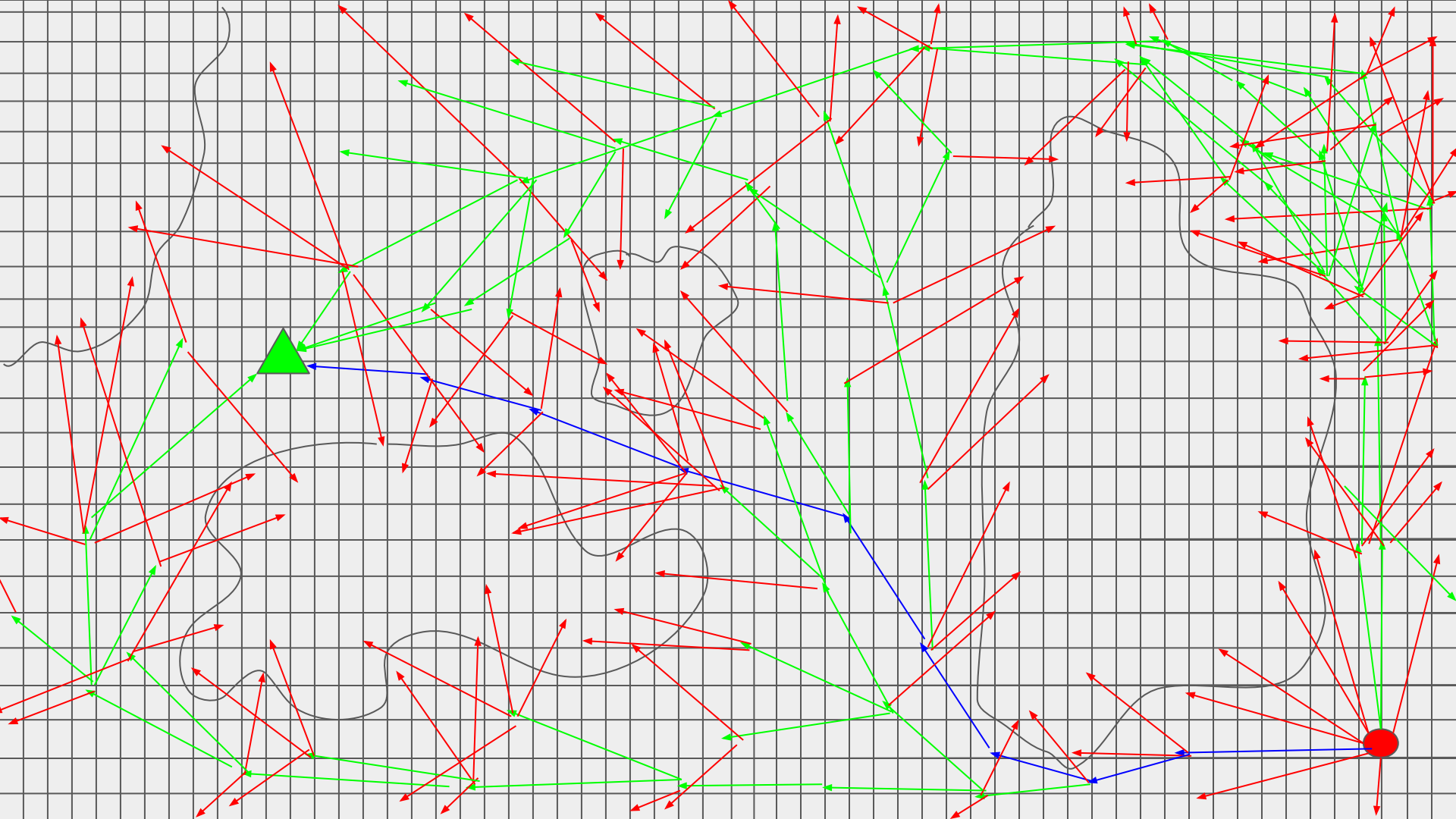


Interval-based search



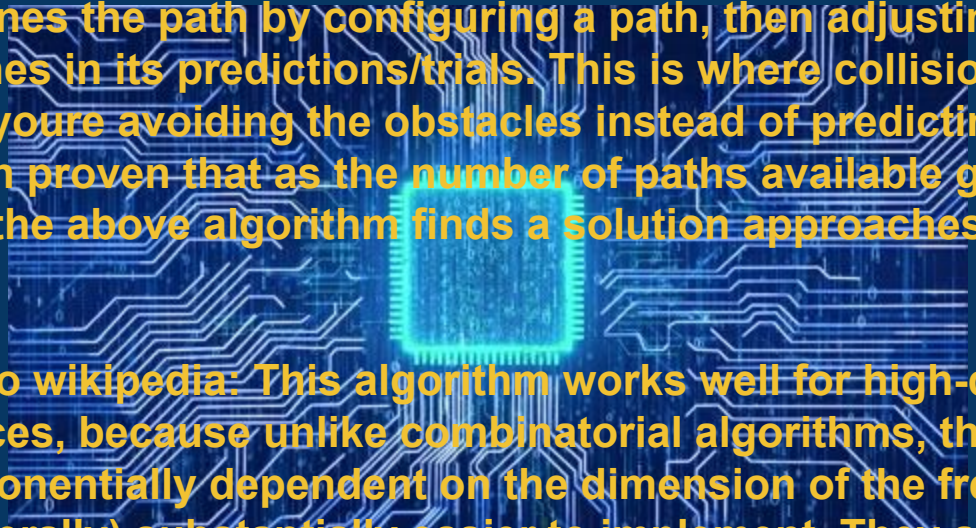
This search algorithm is similar to grid-based search, except it generates a path by covering its environment completely with data instead of using a grid. Each path is composed of multiple other predicted paths that all either lead to the goal or leads to an obstacle. It adjusts accordingly and creates an array of places where a possible path may lead to the goal. When the angle is found that allows the object to move towards the goal, the length of the next interval is measured.

According to wikipedia, “When no path exists the initial position to the goal, we have the guarantee that no feasible path exists in the free area. As for the grid-based approach, the interval approach is inappropriate for high-dimensional problems, due to the fact that the number of boxes to be generated grows exponentially with respect to the dimension of configuration space.”

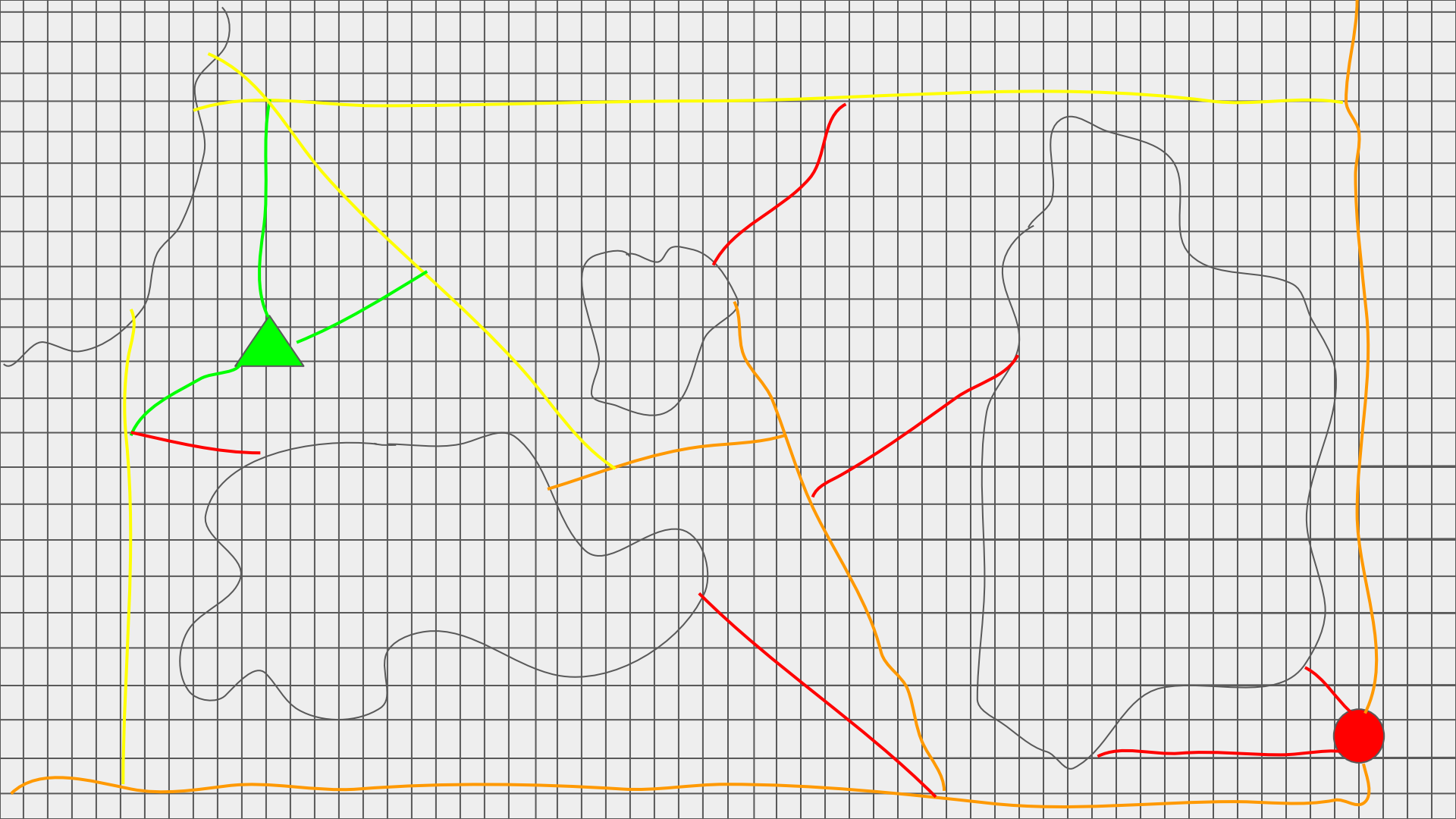


Sampling-based algorithms

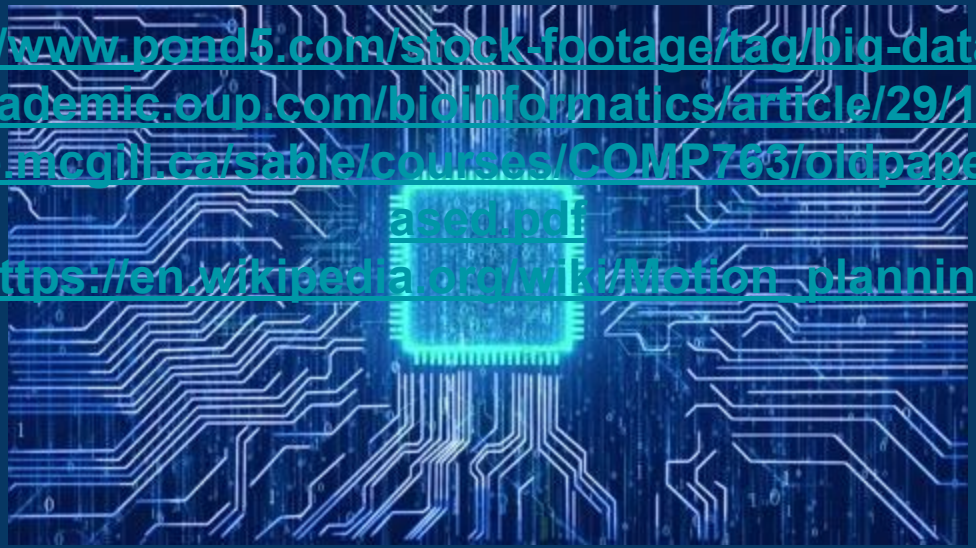
Sampling-based algorithms represent the configuration space with a map of sampled paths. It determines the path by configuring a path, then adjusting depending on milestones it reaches in its predictions/trials. This is where collision detection comes into play, since youre avoiding the obstacles instead of predicting a way around them. it has been proven that as the number of paths available grows higher, the probability that the above algorithm finds a solution approaches 1 exponentially



According to wikipedia: This algorithm works well for high-dimensional configuration spaces, because unlike combinatorial algorithms, their running time is not (explicitly) exponentially dependent on the dimension of the free space available. They are also (generally) substantially easier to implement. They are probabilistically complete, meaning the probability that they will produce a solution approaches 1 as more time is spent. However, they cannot determine if no solution exists.”



Sources



https://www.pond5.com/stock-footage/tag/big-data-chip/
https://academic.oup.com/bioinformatics/article/29/1/1/273289
https://svn.sable.mcgill.ca/sable/courses/COMP763/oldpapers/yap-02-grid-based.pdf
https://en.wikipedia.org/wiki/Motion_planning