A SURVEY ON ROBOT MOTION PLANNING ALGORITHMS

Sai Manoj Jalam (75386398)

Abstract:

Robots have had a tremendous impact in 21st century . Motion planning algorithms are central to allow the movement of robots and are necessary to move robots in a cost-effective and collisions free manner. Due to its wide applications, there have been many path planning algorithms developed , to move robots better. The objective of this paper is to present a state-of-the-art survey of some algorithms of motion planning for robots. A brief introduction on various algorithms on robotic motion is made. The analysis of each algorithm is presented along with their relative advantages and disadvantages.

1.Introduction:

Due to the high application of motion planning in real-world scenarios, the challenge of high-performance and realistic paths has been an important research topic. Over the years, famous algorithms such graph searching algorithm (A^*) , random sampling algorithm (rapidly exploring random tree (RRT)), classical algorithm (artificial potential field(APF)), Intelligent bionic algorithm (ant colony , genetic algorithm) Artificial Intelligence (fuzzy logic control, neural network) have been developed .Motion planning algorithms can be classified into two major types i.e. global planner and local planner.

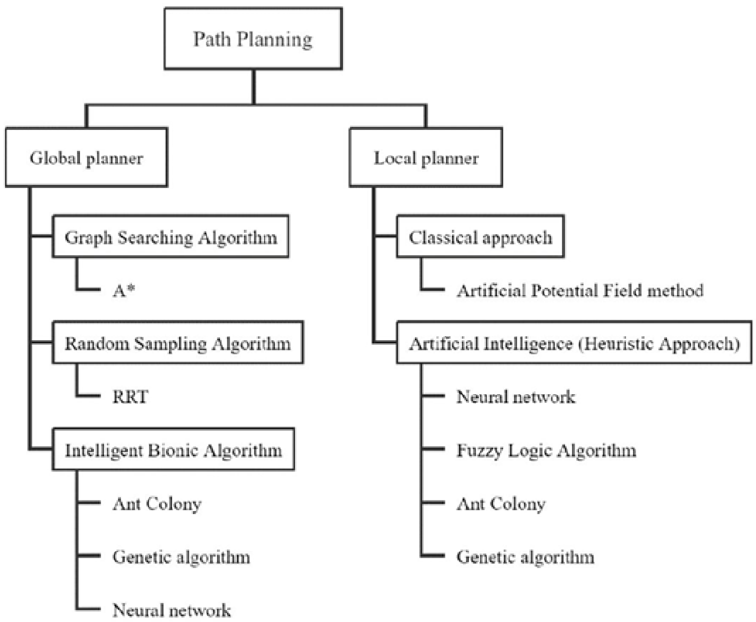

Fig. 1. A categorization of major motion planning algorithms.

2.CLASSICAL METHODS:

Classical methods lay a solid foundation for the solution of path planning problems, such as graph-based Dijkstra's algorithm, A^* ,etc.

2.1 A* algorithm:

The A* method is based on the best-first search and employs a heuristic function to determine the optimal path by calculating the total cost. When estimating route cost, the approach differs from the Dijkstra's algorithm. The following is how A* estimates the cost of a node i in a graph:

(1) calculate the distance between the starting node and node i.

(2) identify the node i's closest neighbour j and calculate the distance between nodes i and j.

(3)calculate the distance between the jth node and the target node.

These three elements are added together to create the overall projected cost:

$$C_i = c_{start,i} + min_j(d_{i,j} + d_{j,goal}).$$

Robotics route planning has traditionally used the A* algorithm. There are several variations of the A* algorithm, including hybrid A* and AD* as well as dynamic A* and dynamic D*, Field D*, Theta*, Anytime Repairing A* (ARA*), and Anytime D*.

2.1.1 Analysis of A^* algorithm:

The generic advantage of this algorithm is that this approach comes from the heuristic's ability to help the algorithm swiftly converge. The drawbacks include it does not take into account obstructions to avoid collisions , having a sluggish searching pace and being unsuitable for large-scale path searches .The detailed analysis of various A^* are summarized below.

Algorithm	Advantages	Disadvantages
A*	Apply graph theory on a discretized configuration space with complete resolution.	High memory and time cost.
ARA*	An efficient anytime heuristic search.	Ineffectively cost the effort in deeper areas of the search space.

D*	It can plan the path in the changing ,known, unknown areas with an efficient and optimal method.	The method can only work on the point which is restricted to the change of cost.
Theta*	The path is a shorter distance.	It costs more time.

TABLE I. ADVANTAGE AND DISADVANTAGE OF VARIANTS OF A*

2.2 RRT algorithm

Sub-optimal pathways are produced by sampling-based algorithms, which randomly sample a given workspace. The RRT algorithm is more well-liked and frequently employed for industrial and commercial reasons. It builds a tree that makes an effort to quickly and consistently search the workspace. Non-holonomic limitations like the vehicle's maximum turning radius and velocity can be taken into account via the RRT algorithm.

2.2.1 Pseudocode

ALGORITHM 1: FIND THE NEXT POINT

Input G $x_{nearest}$ x_{rand}

Output $x_{nearest}$

$x_{reachest} \leftarrow x_{nearest}$

while $x_{reachest} \neq x_{nearest}$ **do**

  if $CollisionFree(x_{rand}, Parent(x_{reachest}))$ then

    $x_{reachest} \leftarrow x_{nearest}$

  else

    Return $x_{reachest}$

  end

end

Return $x_{reachest}$

2.2.2 Analysis

This approach has the following relative benefits:

(i) It is effective at resolving path-planning issues in high-dimensional spaces.
(ii) can be used for online, real-time planning.
(iii) stays out of areas that have already been examined.

The drawbacks include the following:

(i) It is inappropriate when road planning calls for tight corridors.

(ii) Since the planning procedure is only a random exploration of the space, the result is sub-optimal.

3. Bionic Methods.

Bionic algorithms are a type of intelligent computer technique that mimic the structure and operation of living things or the intelligent behaviour of biological communities. In the complicated solution space, bionic algorithms often find a globally optimum solution.

3.1 Genetic algorithm

A simultaneous and global search method that mimics natural genetic operators is called a genetic algorithm (GA). It is more likely to reach the global optimal since it assesses several locations in the parameter space concurrently.

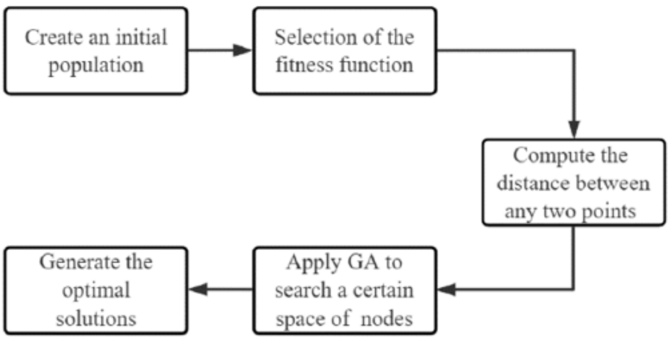

Fig : GA applied to path planning

The GA's central tenet is based on Darwin's theory of survival of the fittest , that is the greatest chromosomal genes should endure and be passed down to future generations. The best chromosomes must be chosen through a selection process. The three phases in this technique are as follows:

1) The values of all chromosomes' objective functions are calculated.

2) Chromosomes are given fitness scores based on their objective function values. In place of the proportional assignment approach, the rank-based fitness assignment is employed . This keeps a small number of superior chromosomes from dominating the population.

3) To create new chromosomes, chromosomes are put into a mating pool after being chosen based on their fitness ratings.

3.1.1 Pseudocode

Algorithm 1 Pseudo-code of a Genetic Algorithm implementation of robotic motion planning.

```
1: Input: P1 : First parent;P2 : Second Parent; VF : Length of P1, VS : Length of the second parent.
2: Output : Fittest optimal path.
3: for i = 1 to VF do
4:     for k = 1 to VS do
5:         if (isFeasible(Vi+1,V j ) = True and isFeasible(V~ j+1,Vi ) = True) then
6:             CrossPointList ← Vi
7:             MatchPointList ← V~ j
8:             end
9:         end
10:    end
11:    fork = 0 to CrossPointListSize do
12:    off spring1 ← (V0,V1,...,Vi−1,Vi,V~ j+1...,V~ s )
13:    off spring2 ← (V~ 0, V~ 1, ..., Vj−1, V~ j, Vi+1,,V~ F )
14:    child ← off spring1 ∪ off sprin2
15:    end
16:sort(child)
```

3.1.2 Analysis:

Studies have shown that GA increases the speed of convergence. And hence can be applied in various application with modifications.

4.Artificial Intelligence Methods.

The fundamental idea of artificial intelligence's motion planning algorithm is to mimic people's thought processes, which are known to be very adaptable. After extensive training, an algorithm with learning capability will likely produce superior operation results.

4.1 Fuzzy logic classification.

The program employs fuzzy logic, which is a less precise categorization method than using a threshold, to classify issues that resemble people. The path planning algorithm based on fuzzy logic can integrate the return data from heterogeneous sensors and task objectives, taking the form of decision-making for path planning that is like that of a human. The fuzzy logic-based route planning algorithm creates a path planning rule by incorporating human path planning and driving experience into the fuzzy logic table.

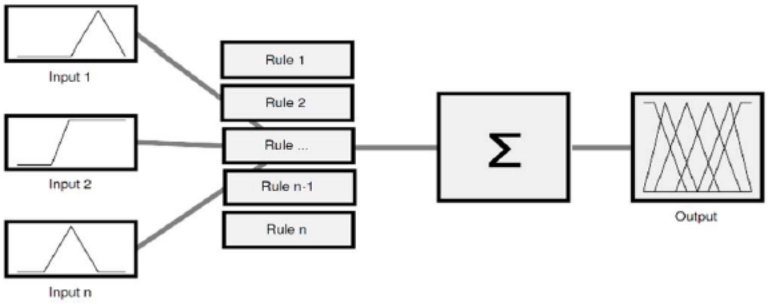

Fig: The structure of fuzzy logic system applied to robotic motion planning.

4.1.1 Analysis

The method's size may be minimized by the application of fuzzy logic, allowing for the creation of a path planning algorithm that is both effective and computationally light. Because of this, it is thought that the path planning method based on fuzzy logic is better appropriate for real-time dynamic environment path planning. And a well-designed fuzzy logic table can help the algorithm get around issues like local poles, which the potential field technique readily produces, and help the program attain high resilience. Therefore, it is better to apply fuzzy logic path planning in an unknowable environment. However, experience-based algorithms will always have certain drawbacks because of the limits of human experience. The reasoning rules and fuzzy tables expand dramatically as the input amount increases, or more specifically as there are more obstacles on the map, which significantly increases the demand for computer resources. When a robot is faced with symmetrical objects, symmetric hesitation in fuzzy logic path planning can also result in a deadlock.

4.2 Reinforcement learning algorithm (Q-learning).

Supervised learning algorithms like CNN are only capable of avoiding static obstacles via one-step prediction; as a result, they are unable to handle avoiding obstacles over time. RL algorithms are suitable for time-sequential problems. Motion planning is accomplished by rewarding destinations and safe routes with large sums of reward, while penalizing barriers (negative reward). The best route is chosen based on the total rewards from the starting point to the destination.

Given an episode <*S1, A1, R2, S2, A2, R3, ..., St, At, Rt+1, ..., ST*>, *Q* learning use the *ε-greedy* approach to select an action A_t at the time step *t*. *Q* learning directly uses maximum estimated action value max at time step *t*+1 to update its action value by

$$Q_{QL}(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha(R_{t+1} + \gamma \max_{A_{t+1}} Q(S_{t+1}, A_{t+1}) - Q(S_t, A_t));$$

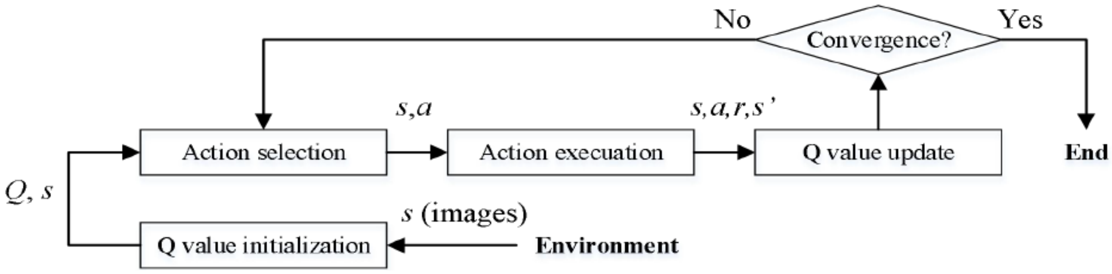

Fig: Steps of Q-learning Algorithm

4.2.1 Pseudocode:

Algorithm: Q-leaning

Initialize $Q(s,a)$ randomly
Repeat (for each episode):
  Initialize s

  Repeat (for each step of episode):
    Choose a from s using policy derived from Q (e.g. E-*greedy*)
    Take action a, observe r, s'
    $Q(s,a)$+-$Q(s,a)$+a[r+ymax$_a$ $Q(s',a')$-$Q(s,a)$]
    s4-s';
  until s is terminal

4.2.2 Analysis:

Features of Q-Learning algorithm includes replay buffer and Weight update method which will ensure that safe distance is optimal distance and speed is optimal speed. This algorithm reaction speed is also is also swift and path obtained is also optimal. Main drawbacks of this method are its speed and stability of convergence is low.

5.Conclusions.

In this survey, various path planning algorithms have been analysed. The earliest algorithms, such as A* and RRT, are usually based on a certain mathematical model. Classical path planning algorithms can find the shortest collision-free path in certain cases, but due to many preconditions required by the algorithm, classical algorithms are usually lacking flexibility. Later genetic algorithms were used for path planning .Recently Artificial intelligence-based path planning algorithms have been developed imitating human like thinking. Even AI-based algorithms usually cannot solve the problems where large amount of computation involved , are can easily fall into local optimal solution, or are unable to converge stably at the same time. In order to integrate the advantages and discard the disadvantages of the above algorithms, hybrid algorithms are the most popular in recent years.

References:

[1] Zhou, C., Huang, B. & Fränti, P. A review of motion planning algorithms for intelligent robots. J Intell Manuf 33, 387–424 (2022). https://doi.org/10.1007/s10845-021-01867-z

[2] Chaymaa Lamini, Said Benhlima, Ali Elbekri,
Genetic Algorithm Based Approach for Autonomous Mobile Robot Path Planning,
Procedia Computer Science. https://doi.org/10.1016/j.procs.2018.01.113.

[3] An Approach of Multi-agent control of Bio-Robots Using Intelligent Recognition Diagnosis Of Persons With Moving Disabilities
Antanas Andrius Bielskis Affiliation ; Dalė Dzemydienė Affiliation ; Vitalij Denisov ; Arūnas Andziulis ; Darius Drungilas
DOI: https://doi.org/10.3846/1392-8619.2009.15.377-394

[4] J. Valasek, J. Doebbler, M. D. Tandale and A. J. Meade, "Improved Adaptive–Reinforcement Learning Control for Morphing Unmanned Air Vehicles," in IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 38, no. 4, pp. 1014-1020, Aug. 2008, doi: 10.1109/TSMCB.2008.922018.

[5] Zhang, Baochang, Li, Ce and Lin, Nana. "10. Reinforcement learning". *Machine Learning and Visual Perception*, Berlin, Boston: De Gruyter, 2020, pp. 127-140. https://doi.org/10.1515/9783110595567-011