2017 Asia-Pacific Engineering and Technology Conference (APETC 2017)
ISBN: 978-1-60595-443-1

A New Coding Method Based on Genetic Algorithm to Solve the Path Planning Problem

Pu Cuo, Cai Ren and Cai Guangbo

ABSTRACT

This paper presents a novel method for robot path planning using genetic algorithm (GA) under polar coordinates space. A concise encoding approach and a speedy computation of fitness means are introduced, and a refining process is adopted to remove the unnecessary points of path made by GA algorithm. Results of simulation show that this path planning method can be used to generate moving path in the complex and multi-obstacle environment for robot.

PREFACE

The purpose of the robot path planning is, based on the characteristics of the robot and the surroundings, to plan a path which is start from the pre-establish location, pass the barrier and then arrive the pre-establish ending and satisfy some optimize index.

People has tried to apply the GA(Genetic Algorithm) to solve the robot path planning problem[1][2][3] for many years and has got certain succeed. The general method is using the series of coordinate of intermediate dot location to be the individual of chromosome in the cartesian coordinates system and transfer to the binary string to do the genetic manipulation. There are some following problem we did not solve yet:

The path is generally composed of Cartesian coordinates (x,y) which belongs to a string of nodes. In order to shrink the rangs of the node coordinates and the search space of genetic factors, we often need to initialization processing of planning region, namely, do some division on great range[2,3]. But this division method often lost excellent path, because it limits the chromosome's way to construct.

Generally, using the starting point ,rank of intermediate dot and ending of the path to be the gene for doing the coding of the chromosome， but the path which is showed by the individual of elder generation of general crossover operation is not cross in the cross location. That is why the offspring is the path which is broken in the middle and need to be fixed by people.

The start path normally is many line segments， which will influent the forward velocity of the robot. For solving this problem, we need to refine the start path again for deleting the unnecessary inflection point.

School of Mathematics and Computer Science, Northwest University for Nationalities
730030 Lanzhou Gansu, China
*Corresponding author: 1224717240@qq.com

In order to solve these problem, this article tried to use a efficient and concise GA. This GA choose the polar coordinate system as the coordinate system of planning for solving the robot path planning problems efficiently.

STATEMENT OF THE PROBLEM

The range of planning and coordinate system are choosen as the picture 1 showed. The o is original point of polar coordinates，P_1 is the start point，P_M is the destination. The shadow in the picture are the actual barrier or the barrier of geology[4,5].

The purpose of path planning is to calculate locations of a series of the path node P_i(i=1,2,…，M). every node is showed as (ρ_i, θ_i).

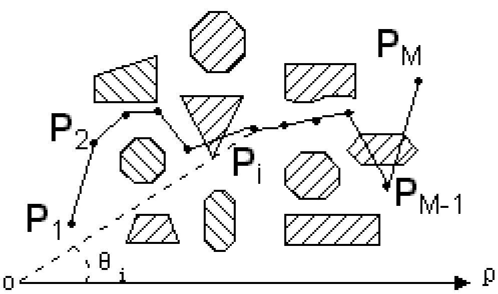

Figure 1. the description of the planning environment.

THE GA ALGORITHM OF PATH PLANNING

In order to solve the path planning problem using genetic algorithms under the environment of large-scale and multi-barrier, we need to adopt efficient coding method and calculated quickly and accurately fitness function and select the reasonable operating parameters. In this paper, we use to coding path position (polar angle, polar radius) directly, efficient chromosomal mutation strategy.

Coding

For individual coding we take the method which coding of path node directly as the picture 1 showed, LJ={ $\boxed{B_1}\boxed{B_2}\ldots\boxed{B_M}$ },the $\boxed{B_i}$ is synthetic binary encoding of polar angle and polar radius which is P_i location. For the coding of path of chromosome ,The more the path node number is ,the more the coding bits are. This will reduce the evolutional efficiency of the algorithm[6].

In order to reduce the length of the chromosome, we use following methods.

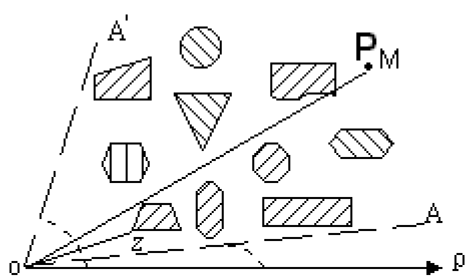

Figure 2. the range of polar angle and polar radius.

（1）identify the number of nodes in the path initially

The number of excessive number of nodes will inevitably affect the speed of genetic evolution. Choosing reasonable number of nodes so that the evolution of computing can be achieved without losing the fine individual path. In this paper, based on the number of obstacles to determine the number of nodes, a large number of simulation experiments shows that the number of nodes should take between $M-2 \sim M+2$, where M is the number of obstacles.

（2）limits node of Polar angle and polar radius range

Control synthetic binary coded bits of polar angle and polar radius is an important way to accelerate genetic progress, while the polar angle and polar radius range determines the number of bits of binary code. Within the framework of the polar coordinate system can be easily visually determine the scope of path node of the polar angle and polar radius.

According to the plan and the distribution range of obstacles, shown in Figure 2, set the boundary OA (polar angle $\theta_{\min}$) and boundary OA'(polar angle $\theta_{\max}$), while determining the nearest obstacle pole rationale point Z.

After taking into account the robot can sometimes border arrived in the target, the maximum and minimum, respectively taken as the polar angle$\theta_{\min}$-5^0 and$\theta_{\max}+5^0$.

The minimum of the polar radius is $||\boldsymbol{OZ}||(1-1/4)$， the maximum is $||\boldsymbol{OP}_M||(1+1/10)$.

To Obtain the Fitness Function

The obstacle avoidance capability of path must be consider firstly and then do compare path length on this basis to determine the fitness function of individual path. In order to determine whether an individual is caught in the path environment obstacles, various obstacles can be approximated showed by polygons. During the fitness function calculations, we can determine whether each individual line segment paths and obstacles each polygon edges intersect. If there is a polygon with a one side intersection, we will stop the calculation of the individual's fitness function which means the fitness function that is considered to be zero.

Testing whether two line segments intersect, the endpoint coordinates of two line segments disposed respectively as (X_{a1},Y_{a1})、(X_{a2},Y_{a2})、(X_{b1},Y_{b1})、(X_{b2},Y_{b2})， now , there are：

$$\begin{cases} \alpha=(Y_{b2}-Y_{b1})(X_{b2}-X_{a1})-(X_{b2}-X_{b1})(Y_{b2}-Y_{a1}) \\ \beta=(X_{a2}-X_{a1})(Y_{b2}-Y_{a1})-(Y_{a2}-Y_{a1})(X_{b2}-X_{a1}) \\ \gamma=(X_{a2}-X_{a1})(Y_{b2}-Y_{b1})-(Y_{a2}-Y_{a1})(X_{b2}-X_{b1}) \end{cases}$$

When the symbols of α、β、$\gamma-\alpha$和$\gamma-\beta$are all same, these two lines intersect.

In order to avoid any comparison between the any two lines of the line segments. We need to do an external rectangular segments. Only when the two segments overlapping rectangles with the external part, we further compare the intersection condition, otherwise the two segments is certainly not intersect.

REFINE THE PATH POINTS

The initial path generated by the genetic algorithm is consist of a series of connected nodes fold line segments. Sometimes it will lead to some unnecessary inflections (Fig. 3) affecting the moving speed of the robot[4]. In this paper, a simple algorithm excluding these inflection points, the algorithm is as follows:

Input： path_ga — node vector generate by genetic algorithm， namely[P_1, P_2, …,P_i， …， P_M].

Output result： path vector after refining[7,8].

```
start=0; target=M-1;
push_vector(path_ga[start]);
finsh=false;
while(!finsh){
if(see(path_ga[start]), path_ga[target])){
 push_vector(path_ga[target]);
      start= target;
      target=size(path_ga)-1;
      if(start= = target)
        finsh=true;}
 else{
 target--;
 if(start= = target){
     start++;
  push_vector(path_ga[start]);
     target=size(path_ga)-1;
}}}
```

SIMULATION TEST

Disorders of a region shown in Figure 3, the number of barriers 10, so select nodes M = 12. The θ range is limited by the boundaries of computing threats in between 12^{0}-76^{0}, ρ range is limited to between 24m-152m. The parameters used are: population size -50; crossover probability -0.55; mutation probability is relatively large which is 0.25, in order to avoid the algorithm premature convergence. Win2000 environment in use VC ++ 6.0[6] the algorithm is implemented.

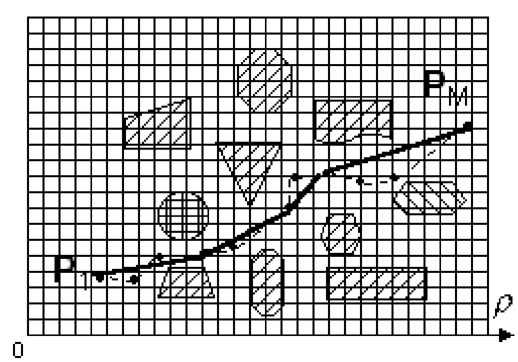

Figure 3. results of simulation.

After Run 3800 generations， we obtain the initial path polyline set shown by dotted lines in Figure 3, it takes time of 20 minutes. After refining the final path shown as black lines, we can see clearly reduce unnecessary inflection point.

CONCLUSIONS

(1)Using polar coordinates for describing robot space environment reasonable reasonably, we simplified modeling of robot path planning. By limiting the scope of θ and ρ to narrow your search, so as to take advantage of genetic algorithm to make it possible and applicable to solve the problem of robot path planning.

(2)In a complex environment, we have overcome the deficiencies that the usage of genetic algorithms for robot path planning must be initialized with a wide range in the past[2][3]and also to avoid the loss of good path.

(3)Using simple coding method and fast fitness calculation process, making the whole process of evolutionary computation simulations can be performed on a PC.

(4)We refined the initial path generation and remove the excess inflection point which make the final path to meet the actual needs of the robot. The simulation test results show that the method is applicable to a wide range of solving robot path planning of complex obstacle environment.

ACKNOWLEDGMENT

The work is supported by Natural Science Foundation of China (No. 61662065) and Scientific and New Silk Road Economic project of Northwest University for Nationalities (No. XSCZLl201605).

REFERENCES

[1] O. Ono, B. Kobayashi, M. Shimizu., Proc. ACC, **57** (2004).
[2] Z.H. Ming, S.Sh .dong, P . Y.wu, AAEAS, **21** (2000).
[3] C.H. Gang, S.H.L. Cheng. Genetic, Robot, **23** (2001).
[4] W.C.H, IEEE T A C,**AC-35** (2005)
[5] Estrin D., Govindan R., Heidemann J., Kumar S., In: P. 5th ACM/IEEE I C MCN, Seattle: IEEE C S, **263～270 (**1999).
[6] Heinzelman W. [Ph. D. Thesis], B.: M IT, (2000).
[7] Y. Jin, L. Wang, CN, **52**(2008).
[8] L. Ch. Fa, CH. G. Hai, CJC, **vol. 30 No. 1**(2007).