Method of robot route control with genetic programming

Peng Si[a], Chen Yuhan[b*]

The 28th Research Institute of China Electronics Technology Group Corporation, Nanjing 210007, China

[a]chen70311@sohu.com, [b]chen7225@163.com

Keywords: genetic programming; robot; route control; knowledge evolutionary

Abstract. This paper studies the robot path planning method,which is suitable for the complex environment. First of all, to abstract modeling of robot's work environment, set up the hexagonal lattice model of environment, every hexagonal has a cost value; This model meets the requirement of describing the different environment; through knowledge evolutionary algorithm based on genetic programming (GP) to get the optimal route choice; at last, through example show that the proposed method is effective.

Introduction

Autonomous mobile robot application is more and more widely, it's field from household appliances to self-driving cars. The main task of the robot is of the human cannot handle task, or at least can reduce the burden of human and strength. However, in the face of increasingly complex problems in using process, what the autonomous robots faced are often beyond the programming considering various occasions of range from the start. Therefore, developing robust adaptive ability of robot control program has been an active research problem.

In recent years, the development of the study tend to adopt knowledge evolution method, through evolutionary algorithm to automatically generate the appropriate application logic [1][2][3][4]. In this paper, using genetic programming (GP) evolution control algorithm for robot path planning. The main advantage of this method lies in the evolutionary control program has reusability, can be applied to different objective functions, such as looking for a route, generating flight lines, etc.

The rest of this article is organized as follows: the second part describe the problem and environmental simulation, as well as the related research work; The third part describes the evolution of genetic programming and related route control algorithm; The fourth part gives experiment results and efficiency analysis, and conclusion.

Automatic control of a model robot

Robot path planning problem description includes several aspects: 1) the environment, the environment can be predefined, as input of robot path planning, the path planning algorithm used by the topology information of environment. 2) the topology information of planning algorithm retrieval environment, to find a suitable route. In this article, we aim to develop a route planning algorithm ,which can adapt to the unknown environment. Environment model by using the hexagonal grid model, each case has a floating point value, as the cost of passing this grid. 3) Simulation model of the robot adopts the literature [5],the robot has six sensors, evenly distributed in 360 degrees, corresponding to the detection of around six directions. With the movement of the robot, through processed in the sensor data, realizes the information on the surrounding environment of exploration. Therefore, the ability of mobile robot is including six direction movement: FW-forward, LF-left forward, RF-right forward, BW-backward, LB-left backward and RB-right backward. Shown as fig.1.

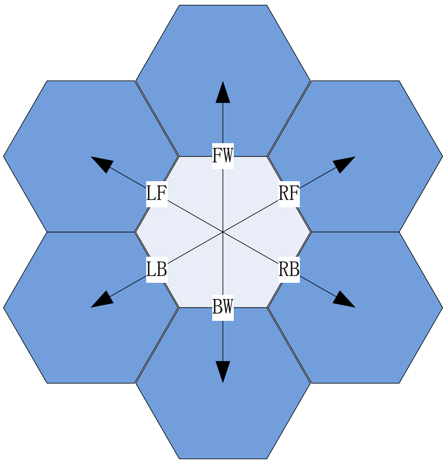

Fig.1 six direction movement

Route planning algorithm generates a result in the current state of robots and sensors data, which get some combination of input values. After movement, new input values are collected and may be next step. Finally, the robot can choose to stop any exercise if it satisfy certain conditions (i.e., to reach the target location or time intervals).

In the process of simulation, robot simulation in an appropriate software framework drived by discrete time steps. At each time step, the input sensors' values processed by the control program, it decided which direction to move. And then simulated by updating the robot position or direction and repeat the process operation. In the different applications, different target behavior can be developed. The control program itself may be artificial evolution and genetic programming, this is detailed in the next section.

Genetic programming applied to behavior modeling

A. Genetic programming

Genetic programming[5] is a kind of optimization, and machine learning techniques, using the concept of evolution to automatically discover the symbols (functions, procedures) of problems at hand. GP is the main idea behind, the problem can be expressed as (computer) program, in most applications in the form of a tree (allow mapping any programming language). Program elements (tree node) must be predefined by the user, must be sufficient to describe a solution to the problem (such as mathematics and logic functions, such as forward, turn left, etc.). Algorithm randomly generated program (possible solutions) and evaluate each project a predefined set of test cases (e.g. program how to control a robot). Every potential to obtain the solution of the quality estimation, fitness value, and then used in the selection process.

Selection process to imitate natural evolution, the weaker the individual (solution) is eliminated, and better personal survival. In addition, the better the individual also participate in the restructuring, two (or more) individual together to form a new solution. The algorithm also integrates a mutation mechanism, a person's change, the probability of a relatively small. The process repeats itself, from the old to build a new generation, until you reach a proper termination criteria. These standards are usually involves the quality we need to find a solution, or perform a predefined time algorithm. The results of the very few human genetic programming examples can be found in [6].

B. Evolving a robot with GP

In this work,route control program is represented in the form of a tree, each node serves as input, you can perform a logic function or an action. There are two main types of tree nodes, each of the generated programs contain node-end and function. The leaves of the terminal node tree program structure; They had no child nodes and their parents function returns a specific value could be the node. The terminal nodes in this application may be one of the following:

- 6 different node (S01, S02,... ,S06), corresponding to 6 sensors return from a simulated robot;
- The least cost of passing grid, a value returns dynamic calculation of all the sensors;

- Nodes perform 6 basic movement function (FW,LF,RF,DW,LB,RB);

C. The fitness function

In the process of evolution, a large number of candidate route to create (control program) must be evaluated. Assessment is through the simulation of the robot to perform actions in a given environment using the selected route. Each route (individual) is given a fitness value, used in the selection process, represents the individual ability to compete with other individuals.

Used in this work, the goal is to require more than a set of robot routes. Due to this property, the definition of fitness function value of access to target grid. The target grid in the process of simulation is set in a uniform grid surface. Control program is that the total number of fitness value is expressed as access to the target grid.

In addition, each grid robot perform (an action or a rotating) assume a fixed cost. If two different evolutionary program to access the same number of target location, use less cost to visit them one step is thought to be better.

It is important to note that the development of a plan or other, does not need to be modified in GP evolutionary system in addition to the appropriate fitness function calculation.

In other words, different (custom) the development of behavior, only need to change the fitness function, and function and terminal elements may be of the same.

D. The initial population

The method of genetic programming needs a population of members to be initialized. The initial population by the iterative procedure of optimization gradually forms the good population. Each member represented as a robot route is hierarchically structured tree consisting of grid number and expresses a feasible solution within the population.

E. Genetic operators

In this paper, two fundamental genetic operators, crossover and mutation, and two advanced operators, elimination and separation, are used in the proposed algorithm. The elimination and separation operators are designed to solve the problems such as redundancy and contradiction.

1) Crossover: In the crossover operation, two parent routes are partially exchanged to form two new offspring routes. The crossover point may occur within a route.

2) Mutation: The mutation operation begins by selecting a parent route from the population at random. A node within the route is randomly selected. The mutation operation then replaces the selected node(including its child nodes) with a randomly generated route in its place.

3) Elimination: The elimination operation is used to solve the redundancy problem, in which two or more routes with the same grids appear in routes set. The operation will remove redundant routes and can thus reduce the route complexity.

4) Separation: The separation operation solves the contradiction problem, in which two routes with the same grids conclude to different classes. It removes some parts of route from a contradictive route such that the route can become consistent.

Result and analysis

The following experiment to assess the efficiency of the route planning of the robot. The following parameters are used in all the experiments, in addition to the provisions:

- a population of 500 ;
- 500 generations of termination conditions;
- at least 10 repeat configuration.

A: the initial tree depth parameter

The purpose of this experiment is the empirical estimates of the corresponding initial depth of the tree, any solution can be reached. This parameter is one of the most influential genetic programming [5]. On the one hand, it should be large enough to allow the complexity of the required to start the program, which can represent a solution to the problem. On the other hand, it should not be too big so appropriate search space is limited, so the complexity of maintain the lowest possible level.

B. Single - population and multiple - population GP

The evolution process can be very time consuming, but also should be repeated several times to obtain the best results, a common extension, many evolutionary system is their parallel [7][8]. In this work, we choose to implement a distributed algorithm that can be deployed in a compute node network, and a multicore machines [9].

C. GP generalization ability

Finally experiment that evolutionary quality solutions in an environment that is invisible. In the process of learning, using simulated evolutionary robot route control program in one or more different environment. Control program is the evolution, it should be in the unprecedented environment test, to evaluate its universality. GP generalization ability studies [10] before and the research results show that includes the number of test cases (such as analog room) may be useful in the search for a more general solution.

Conclusions

This paper introduces the use of genetic programming control procedures in the evolution of the robot in a simulated environment. Results show that the method can produce route, meet the demand of the robot's route planning. For further work in this field may include the implementation of this method to a real environment and the actual robot.

References

[1] C. Lazarus and H. Hu, “Using genetic programming to evolve robot behaviours,” Nature, no. April, 2001.

[2] P. Nordin and W. Banzhaf, “An on-line method to evolve behavior and to control a miniature robot in real time with genetic programming,” Adaptive Behavior, vol. 5, no. 2, pp. 107–140, 1997.

[3] ——, “Real time control of a khepera robot using genetic programming,” CYBERNETICS AND CONTROL, vol. 26, pp. 533–561, 1997.

[4] J. Busch, J. Ziegler, C. Aue, A. Ross, D. Sawitzki, and W. Banzhaf, “Automatic generation of control programs for walking robots using genetic programming,” in Proceedings of the 5th European Conference on Genetic Programming, ser. EuroGP ’02. London, UK: Springer-Verlag, 2002, pp. 258–267. [Online]. Available: http://dl.acm.org/citation.cfm?id=646810.704243

[5] J. R. Koza, “Genetic programming – on the programming of computers by means of natural selection.”

[6] ——, “Human-competitive results produced by genetic programming,” Genetic Programming and Evolvable Machines, vol. 11, pp. 251–284, September 2010. [Online]. Available: http://dx.doi.org/10.1007/s10710-010-9112-3

[7] E. Cantú-Paz, “Efficient and accurate parallel genetic algorithms,” 2001.

[8] T. Alba, “A survey of parallel distributed genetic algorithms,” Complexity, vol. 4, no. 4, 1999.

[9] E. Cantú-Paz, “Topologies, migration rates, and multi-population parallel genetic algorithms,” in Banzhaf et al., eds, Proceedings of the Genetic and Evolutionary Computation Conference, vol. 1, pp. 91–98.

[10] I. Kushchu, “Genetic programming and evolutionary generalization,” Evolutionary Computation, IEEE Transactions on, vol. 6, no. 5, pp. 431 – 442, oct 2002.