Journal of Zhejiang University SCIENCE
ISSN 1009-3095
http://www.zju.edu.cn/jzus
E-mail: jzus@zju.edu.cn

Neural network and genetic algorithm based global path planning in a static environment*

DU Xin (杜 歆)†1, CHEN Hua-hua (陈华华)[2], GU Wei-kang (顾伟康)[1]

([1]*Department of Information Science and Electronics Engineering, Zhejiang University, Hangzhou 310027, China*)

([2]*School of Communication Engineering, Hangzhou Dianzi University, Hangzhou 310018, China*)

†E-mail: duxin@zju.edu.cn

Received May 9, 2004; revision accepted Oct. 7, 2004

Abstract: Mobile robot global path planning in a static environment is an important problem. The paper proposes a method of global path planning based on neural network and genetic algorithm. We constructed the neural network model of environmental information in the workspace for a robot and used this model to establish the relationship between a collision avoidance path and the output of the model. Then the two-dimensional coding for the path via-points was converted to one-dimensional one and the fitness of both the collision avoidance path and the shortest distance are integrated into a fitness function. The simulation results showed that the proposed method is correct and effective.

Key words: Mobile robot, Neural network, Genetic algorithm, Global path planning, Fitness function

doi:10.1631/jzus.2005.A0549 **Document code:** A **CLC number:** TP242.6

INTRODUCTION

The path planning problem of a mobile robot is to find a safe and efficient path for the robot, given a start location, a goal location and a set of obstacles distributed in a workspace. The robot can go from the start location to the goal location without colliding with any obstacle along the path. In addition to the fundamental problem, we also try to find a way to optimize the plan, say to minimize the time required or distance traveled (Wu *et al*., 1996; Sadati and Taheri, 2002; Ramakrishnan and Zein-Sabatto, 2001; 2002).

The popular methods are the visibility graph algorithm and the artificial potential field algorithm. However, the former lacks flexibility and the latter is prone to suffer from difficulties with local minima (Alexopoulos and Griffin, 1992; Chen and Liu, 1997; Borenstein and Koren, 1989). Neural network and genetic algorithm have been shown to be very efficient in robot navigation (Zarate *et al*., 2002; Del *et al*., 2002). General path planning methods based on neural network always establish the neural network model for a robot from the start position to the goal position and entail much computational time. The input data of the model are the previous distance values and position or direction from the sensors. The output data are the next position or direction by self-learning process. Genetic algorithm is multi-search algorithm based on the principles of natural genetics and natural selection (Goldberg, 1989). Genetic algorithm provides a robust search in complex spaces and is usually computationally less expensive than other search algorithms. Genetic algorithm searches the solution from a population of points and is less likely to be trapped in a local optimum. Many results in the literature show the good application of genetic algorithm in robot path planning (Khoogar and Parker, 1991; Ram *et al*., 1994; Noboru and Hideo, 1997).

*Project supported by the National Natural Science Foundation of China (No. 60105003) and the Natural Science Foundation of Zhejiang Province (No. 600025), China

In general, autonomous navigation assumes an environment with known and unknown obstacles. It includes global path planning algorithms to plan the robot's path among the known obstacles, as well as local path planning for real time obstacle-avoidance (Ramakrishnan and Zein-Sabatto, 2001; 2002). There is a very important problem on how to plan an optimal path in a static environment where the positions of obstacles are known first. This planning, also called static path planning, has a wide range of applications (Wang *et al*., 2002).

Path planning algorithms generally find the global optimal path in all passable regions. If the search space composed of the found passable regions is smaller than the space of all the passable regions, the global optimum is not likely to be found. In this paper, a new technique based on the concepts of neural network and genetic algorithm is proposed. We constructed a neural network model of environmental information in the workspace for a robot and established the relationship between a collision avoidance path and the output of the model. Using this model, the proposed algorithm can know the environment, and search the global optimum simultaneously without knowledge of any passable region. To reduce the length of the path code, we converted the two-dimensional via-points of path to a one-dimensional one and integrate the fitness of both the collision avoidance path and the shortest distance into a simple fitness function. At last, we present the computer simulations of a robot path planning problem.

ENVIRONMENT MODELLING BY NEURAL NETWORK

First, we suppose that the robot moves in the workspace as described below:

(1) The robot moves in a limited two-dimensional space;

(2) The robot can be considered as a particle if the boundary of each obstacle is extended by the half size of the robot's maximal dimension in length or width direction;

(3) The obstacles in the workspace can be described as convex polygons.

Without loss of generality, suppose that the workspace for a robot is as shown in Fig.1, and that the shadowed parts represent the obstacles. According to Zhu *et al*.(2002), the environment can be described by the neural network shown in Fig.2 which can be written as Eq.(1).

$$\left.\begin{aligned} C_i^1 &= f(T_I), \\ T_I &= \sum_{m=1}^{M} O_{Mm} + \theta_T, \\ O_{Mm} &= f(I_{Mm}), \\ I_{Mm} &= w_{xm}X_i + w_{ym}Y_i + \theta_{Mm} \end{aligned}\right\} \tag{1}$$

where C_i^1, C_i^2: the outputs of the nodes of the top layer; T_I: the input of the nodes of the top layer; θ_T: the threshold of the nodes of the top layer; O_{Mm}: the output of the mth node of the medium layer; I_{Mm}: the input of the mth node of the medium layer; θ_{Mm}: the threshold of the mth node of the medium layer; w_{xm}, w_{ym}: the weights from the input layer to the medium layer; $f(x)=1/(1+e^{(-x/T)})$: the excitation function; (X_i, Y_i): a random point in the workspace.

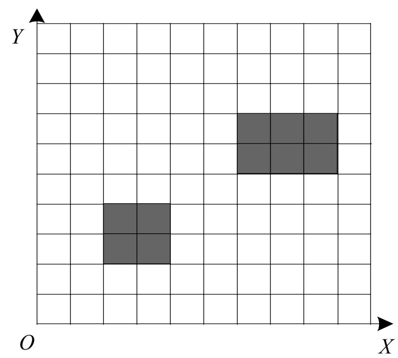

Fig.1 The workspace for a robot

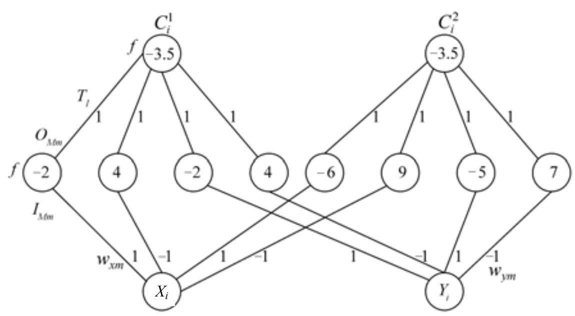

Fig.2 The neural network for the environment

The output of the neural network model of each point in the workspace is 0 or 1. $C_i^k=1$ (k=1,2) implies that (X_i,Y_i) is in the kth obstacle region, otherwise the point is not in it.

If the radius of the robot could be ignored, the robot can be regarded as a particle. If the neural network model of the point (X_i,Y_i) produces $C_i^k=1$ (k=1,2) when the robot arrives, it collides with the kth obstacle in the workspace. On the contrary, it does not. Thus the collision avoidance path can be described as the path where the output of the neural network model of each (X_i,Y_i) is $C_i^k=0$ (k=1,2). So the path planning algorithm can know the current environment information in real-time according to the output of the neural network model.

In Fig.1, only two obstacles are included in the workspace, so the neural network model is simple. If there are more than two obstacles in it, the model is more complexity, as shown in Fig.3.

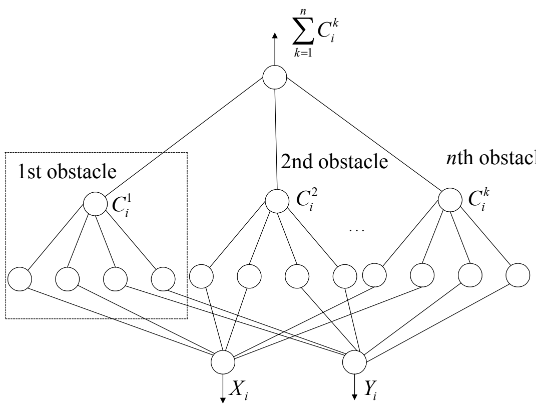

Fig.3 The neural network for the multi-obstacle environment

PATH PLANNING BASED ON GENNETIC ALGORITHM

Path coding

In genetic searching algorithm, the coding technique is important in that the length of binary strings from the parameter sets made up of the via-points of a path, as well as the size of the search space, determines the computational time for a given fitness function. We devised a simple coding technique to shorten the length of the binary string by projecting the two-dimensional data to one-dimensional ones as shown in Fig.4. The aim of the algorithm is to determine the node points (x_i,y_i) (i=1,2,…,n) that constitute the optimal path from the start to the goal position. In order to reduce the length of the string, we converted the workspace XOY, in which each node point of path is two-dimensional, to a new coordinate space $xo'y$, where x axis is the line determined by the start and the goal position. Then the set of the node points x_i is located at equal distances along the x axis. Therefore, y_i becomes the search space for each via-point of the robot path and the via-point candidates are specified by the one-dimensional data.

The coding is done in float type and its structure is shown in Fig.5.

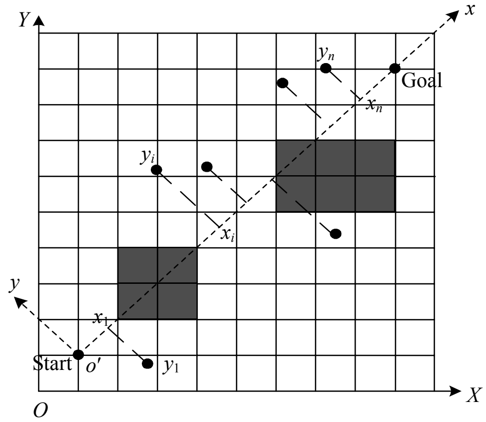

Fig.4 Projecting of search space

y_1	y_2	y_3	…	y_n

Fig.5 The coding structure

Fitness function

The fitness function is an important factor for the convergence and the stability of genetic algorithm. The collision avoidance and the shortest distance should be considered in path planning. The summation of each evaluation function weight is a typical method to construct the fitness function (Noboru and Hideo, 1997; Woonggie *et al*., 1997), but it is prone to be instable and its weight coefficients are difficult to tune and determine as they are as variable as the path and the obstacles change. So when we construct the fitness function, the number of evaluation functions is as small as possible. On the other hand, the two evaluation functions, the collision avoidance and the shortest distance, must be integrated into a fitness function.

Collision avoidance is essential to path planning and makes the mobile robot travel in the workspace safely. Collision avoidance can be depicted as:

1. Each via-point y_i is not in the any obstacle region. The fitness value is $fit11=1$;

2. Each section y_iy_{i+1} does not intersect any obstacle region. The fitness value is $fit12=1$.

If the workspace for the robot is as shown in Fig.4, the via-point y_i should not be in the obstacle regions. Combined with the output of the neural network model, the fitness function of collision avoidance $fit1$ can be written as Eq.(2).

$$fit1=fit11\times fit12 \tag{2}$$

$$fit11=\begin{cases}1 & \text{if } \sum C_i^k=0 \quad (k=1,2)\\ 0 & \text{others}\end{cases}$$

$$fit12=\begin{cases}0 & y_iy_{i+1}\cap obstacle \quad (k=1,2)\\ 1 & \text{others}\end{cases}$$

where i is the ith point in the path. Eq.(2) implies that the path is collision avoidance if the fitness value of the each via-point in the path is 1, otherwise the value is 0.

In addition to collision avoidance, the path can be optimized for minimum distance whose fitness function $fit2$ can be described by Eq.(3).

$$fit2=\sum_{i=0}^{n-1}\sqrt{(x_{i+1}-x_i)^2+(y_{i+1}-y_i)^2} \tag{3}$$

Thus the final fitness function is constructed as shown in Eq.(4). When the fitness function fit reaches the maximum, the global optimal path is found. This not only makes computation simple but also overcomes the disadvantage of the instability from the summation of evaluation function weights.

$$fit=fit1/fit2 \tag{4}$$

Genetic operations definition

Initial population can be chosen randomly on the lines perpendicular to the x axis, each of which is through different one of the $x_1,x_2,\ldots,x_n$ points in the workspace. The population size indicates the number of paths in the workspace. Larger size produces more accurate path and the global optimum is likely to be found, but computational time is longer correspondingly. In general, the size of the population is between [20,100].

1. Selection or reproduction

It is a process in which individual strings are copied into the next population according to their fitness, i.e., the "better" strings survive to reproduce and the less highly fit strings "die". Fitness of a string is determined by the objective function which is to be optimized. There are several different methods for determining how effectively a given string "competes" in the reproduction process. In this paper, Monte Carlo method is used. Besides, the individual smallest fitness value in the next generation is replaced by the individual highest fitness value in the previous one so that the optimal individual is not destroyed during the evolutionary process.

2. Crossover

The optimum in the existing population can be found during the above operation but the individual that is different from the previous one cannot be produced. Then crossover can do this by swapping characters according to some probability to create new strings so that "better" ones can be produced. The probability determines the frequency of crossover. The higher the frequency, the faster the optimum speed reaches, but is probable that convergence occurs before the optimum is reached. In general, the probability is between [0.4,0.9]. In the proposed algorithm, one-point crossover is done with the crossover position and the number of points being generated randomly.

3. Mutation

The reproduction and the crossover can find the optimum in the existing character arrays, but fail to produce new characters to find the best solution because of premature convergence. The mutation changes the characters in an individual string with a very small probability between [0.001,0.4]. Mutation brings in new possibilities for improvements and ensures that some important information is produced during crossover and that reproduction should not be lost. Then the fitness values of the new population's strings are evaluated. To mutate an individual, zero-mean white gauss noise between [−0.15,0.15] is added to y_i.

SIMULATION RESULTS

In this section, simulation and experimental results are obtained by applying the proposed path planning method to navigate a mobile robot. Fig.6 and Fig.7 show the robot navigations. Obstacles, and start and goal positions are marked as rectangles and circles respectively. Table 1 gives their configurations and computation times under the platform of Visual C++ 6.0 on a personal computer with a 2.4 G PIV processor. Fig.6 and Fig.7 show that the robot can go from the start to the goal position without colliding with any obstacle in a static environment. The experiment results indicating successful application of global path planning show that the algorithm is correct and effective.

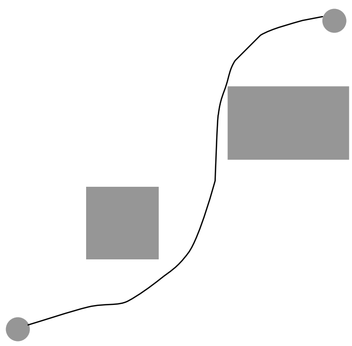

Fig.6 Simulation results of global path planning for Fig.4

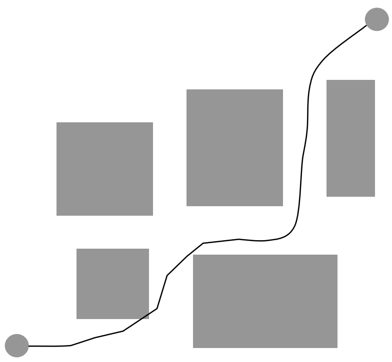

Fig.7 Simulation results of global path planning for multi-obstacle environment

Table 1 Configurations and computational times of the two examples

Example	Search space	String length	Sample distance	Computational time (ms)
Fig.6	30	17	0.71	in 500
Fig.7	45	51	0.31	in 3000

Visibility graph is a classical method in path planning. Fig.8 shows the result of the visibility graph method. Compared with the proposed algorithm, visibility graph method requires less computational time and shorter travel distance because its search space, consisting of vertices of the obstacles only, is much smaller than that of the genetic algorithm, which is one of the steps of the proposed method in which the optimal path is composed of some vertices. Seventy percent of the computation time is due to the neural network introduced to construct and compute the environment model for each character of the string in the workspace. Furthermore, the path produced by visibility graph method is the sum of the straight segments while the result of the proposed algorithm is not so. Though the results are different, each of them produces a "shortest" path in the respective space but not the "shortest" path (Dozier *et al*., 1998). The visibility graph fails when the obstacles can move (Alexopoulos and Griffin, 1992). Path planning using the visibility graph method in moving obstacles environment has been intensively studied. The proposed method in this paper has been successfully extended to dynamic obstacle avoidance (Chen *et al*., 2004).

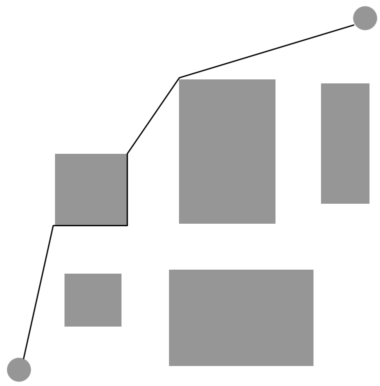

Fig.8 Simulation results of global path planning for multi-obstacle environment using visibility graph algorithm

CONCLUSION

In this paper, we propose a new path planning method based on neural network and genetic algorithm. The method constructs a neural network model of environmental information in the workspace for a robot and establishes the relationship between a collision avoidance path and the output of the model. This genetic algorithm was applied to find the global optimal path in static environment. Computer simulation and experimental results are given to show the feasibility of the proposed algorithm.

References

Alexopoulos, C., Griffin, P.M., 1992. Path planning for a mobile robot. *IEEE Transactions on Systems, Man and Cybernetics*, **22**(2):318-322.

Borenstein, J., Koren, Y., 1989. Real-time obstacle avoidance for manipulators and mobile robots. *IEEE Transactions on Systems, Man and Cybernetics*, **5**(19):1179-1187.

Chen, L., Liu, D.Y., 1997. An efficient algorithm for finding a collision-free path among poly obstacles. *Journal of Robotics Systems*, **7**(1):129-137.

Chen, H.H., Du, X., Gu, W.K., 2004. Path planning Method Based on Neural Network and Genetic Algorithm. International Conference on Intelligent Mechatronics and Automation. Sichuan, China, p.667-671.

Dozier, G., McCullough, S., Homaifar, A., Tunstel, E., Moore, L., 1998. Multiobjective Evolutionary Path Planning via Fuzzy Tournament Selection. The 1998 IEEE International Conference on Evolutionary Computation Proceedings. Alaska, p.684-689.

Del, H.A.R., Medrano, M.N., Martin, D.B.B., 2002. A Simple Approach to Robot Navigation Based on Cooperative Neural Networks. IEEE 28th Annual Conference of the Industrial Electronics Society. Spain, p.2421-2426.

Goldberg, D.E., 1989. Genetic Algorithm in Search, Optimization and Machine Learning. Addison-Wesley Publishing Company, Inc., p.23-25.

Khoogar, A.R., Parker, J.K., 1991. Obstacle Avoidance of Redundant Manipulators Using Genetic Algorithms. Proceedings IEEE International Conference on Robotics and Automation. Sacramento, p.317-320.

Noboru, N., Hideo, T., 1997. Path Planning of agricultural mobile robot by neural network and genetic algorithm. *Computers and Electronics in Agriculture*, **18**:187-204.

Ram, A., Arkin, R., Boone, G., 1994. Using genetic algorithms to learn reactive control parameters for autonomous robotic navigation. *Adaptive Behavior*, **2**(3):100-107.

Ramakrishnan, R., Zein-Sabatto, S., 2001. Multiple Path Planning for A Group of Mobile Robots in A 3D Environment Using Genetic Algorithms. Proceedings of IEEE Southeast Con. South Carolina, p.359-363.

Ramakrishnan, R., Zein-Sabatto, S., 2002. Multiple Path Planning for A Group of Mobile Robots in A 2D Environment Using Genetic Algorithms. Proceedings IEEE Southeast Con. Columbia, p.65-71.

Sadati, N., Taheri, J., 2002. Genetic Algorithm in Robot Path Planning Problem in Crisp and Fuzzified Environments. IEEE International Conference on Industrial Technology. Bangkok, Thailand, p.175-180.

Wang, C., Soh, Y.C., Wang, H., 2002. A hierarchical Genetic Algorithm for Path Planning in A Static Environment with Obstacles. IEEE Canadian Conference on Electrical and Computer Engineering, Canada, p.1652-1657.

Woonggie, H., Seungmin, B., Taeyong, K., 1997. Genetic Algorithm Based Path Planning and Dynamic Obstacle Avoidance of Mobile Robots. IEEE International Conference on Computational Cybernetics and Simulation. Orlando, p.2747-2751.

Wu, K.H., Chen, C.H., Lee, J.D., 1996. Genetic-based Adaptive Fuzzy Controller for Robot Path Planning. Proceedings of the Fifth IEEE International Conference on Fuzzy Systems. New Orleans, p.1687-1692.

Zarate, L.E., Becker, M., Garrido, B.D.M., Rocha, H.S.C., 2002. An Artificial Neural Network Structure Able to Obstacle Avoidance Behavior Used in Mobile Robots. IEEE 28th Annual Conference of the Industrial Electronics Society. Spain, p.2457-2461.

Zhu, Y., Chang, J., Wang, S., 2002.A New Path-planning Algorithm for Mobile Robot Based on Neural Network. IEEE Region Tenth Conference on Computers, Communications, Control and Power Engineering. Beijing, p.1570-1573.