

Path Planning Based on Two-layer Adaptive Genetic Algorithm

Xiang Xu, Kun Zou

University of Electronic Science and Technology of China Zhongshan Institute, Zhongshan 528402, China
E-mail: xuxiang8@mail2.sysu.edu.cn, cszoukun@foxmail.com

Abstract—Path planning for autonomous agent is an important issue in artificial intelligence, its purpose is to find a reasonable path by following certain optimization criteria, such as the length of path is shorter, the path can avoid collision, and the path is smooth, etc. This paper proposed a two-layer adaptive genetic algorithm (TAGA) and applied to path planning. On the one hand, we use one of genetic algorithm to find the optimal path (called as *pathfinding GA*), and adopt fuzzy logic to adjust crossover probability and mutation probability. On the other hand, we use another genetic algorithm to optimize the fuzzy reasoning rules and type of membership functions (called as *self-learning GA*). These two GA work cooperatively, self-learning GA provide the optimal individual for pathfinding GA, that means the optimal fuzzy reasoning rules and type of membership functions, at the same time, the selection of the optimal individual in self-learning GA need recur to pathfinding GA. The proposed TAGA method shows efficiency in path planning, and we demonstrate this point by applying it to the static and dynamic environments. Experimental results show that the proposed TAGA method overcomes premature convergence of the standard genetic algorithm (SGA), speed up convergence, and enhanced the application scope of the adaptive genetic algorithm (AGA).

Keywords—two-layer adaptive genetic algorithm (TAGA); path planning, crossover probability; mutation probability; adaptive genetic algorithm (AGA)

I. INTRODUCTION

The purpose of path planning for autonomous agent aims at generating a path with security, shorter, smooth and collision avoidance, and also provides agent the ability of sensing surrounding environment [1-4]. Genetic algorithm (GA) is a heuristic intelligent search algorithm that mimics the process of natural evolution. By means of natural evolution techniques, such as inheritance, selection, crossover and mutation [5], GA can bring good applicability in solving plenty of nonlinear optimization problems. Recently, GA has been widely used for path planning in the complicated unknown environment [6, 7]. In order to improve convergence speed of GA and quality of the solution, on the basis of the standard GA, Srinivas et al. proposed an adaptive genetic algorithm (AGA), which can automatically adjust crossover probability and mutation probability along with the fitness value [8]. Then based on the AGA, a series of different adaptive adjustment algorithms are put forward [9, 10]. Although these algorithms have different expression forms, they are divided into two categories according to their adjust strategy: 1) adjustment algorithm based on traditional

heuristic knowledge [11, 12], 2) adjustment algorithm based on intelligent control technology, especially based on fuzzy logic control strategy [13-16]. And even if adopting fuzzy logic control strategy, because of the difference in the selection of input and output variables, the type of membership functions, the fuzzy reasoning rules, and the defuzzification algorithms, etc., still bring considerable difference for the performance of fuzzy controller. Therefore, although many researchers adopt fuzzy logic to adjust crossover probability and mutation probability, their specific algorithms and experimental results are different.

On the basis of the predecessor's achievement, this paper adopted an adaptive GA for path planning, and introduced the fuzzy logic controller to adjust crossover probability and mutation probability. Simultaneously, in order to obtain the optimal crossover probability and mutation probability, the paper introduced a self-learning mechanism to construct the knowledge base of fuzzy controller, which means adopting another GA for optimizing the fuzzy reasoning rules and corresponding type of membership functions.

The remainder of this paper is organized as follows. Section 2 presents the proposed two-layer adaptive genetic algorithm (TAGA), which provides the details of the considered path planning. Section 3 discusses experimental results for the capabilities and performance of the proposed TAGA scheme. The experiments are conducted using static and dynamic environment respectively. Finally, Section 4 concludes the paper with some remarks and hints at plausible future research lines.

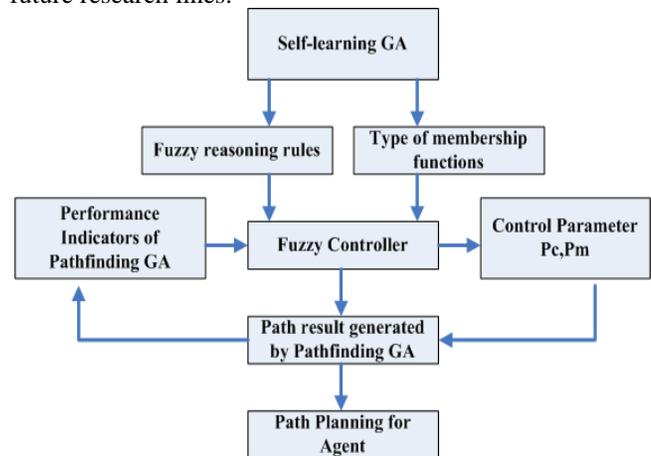
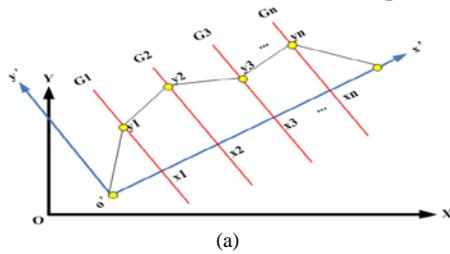


Figure 1. Architecture diagram of the proposed two-layer adaptive genetic algorithm (TAGA).

II ALGORITHM DESCRIPTION

The proposed method adopted a two-layer GA, one of GA is used to find the optimal path (called as *pathfinding GA*). In order to improve the performance of pathfinding GA, we adopted a fuzzy controller to adaptively adjust crossover probability and mutation probability; Another GA (called as *self-learning GA*) is used for self-learning of fuzzy controller, means to optimize the fuzzy reasoning rules and corresponding type of membership functions, which can help fuzzy controller to find the best crossover probability and mutation probability for pathfinding GA. At the end of each iteration of pathfinding GA, self-learning GA and fuzzy controller will generate the optimal crossover probability and mutation probability according to the fitness value of current individuals in the pathfinding GA, which can accelerate the process of pathfinding GA, speed up convergence and precision of results. The architecture diagram of the TAGA is shown in Fig.1.

In the proposed TAGA method, the self-learning GA is used to optimize the knowledge base of fuzzy controller. The fuzzy reasoning rules commonly derived from expert knowledge or long-term experience, which often keep unchanged in the process of fuzzy reasoning. But for different control objects or different input conditions, this kind of invariance does not always guarantee a good control effect, so we need to adopt certain optimization method to obtain the best matching rules. Also for the type of membership functions, which directly affect the fuzzification of input variables and the defuzzification of output variables,



different type of membership functions can obtain different degree of membership even if the input value is the same. Therefore for each fuzzy reasoning rule, it is necessary for optimizing the corresponding type of membership functions.

II. DESIGN OF THE PATHFINDING GA

The goal of path planning for autonomous agent is to find an optimal or suboptimal path from starting point to end point in an obstacles environment. The path should meet criteria as follows: 1) path can't collision with any static or dynamic obstacles; 2) autonomous agent should keep a safe distance from the obstacles as far as possible; 3) the length of path should be as short as possible; 4) the path should be as smooth as possible.

1) *Path coding*: The length of path coding is an important factor that will affect the convergence speed of GA, so we should try to use simple genetic coding [17]. For path planning issue, we naturally consider path node sequence as coding pattern. In order to reduce the length of path coding, we perform transformation on the old coordinate system and generate a new coordinate system, this new coordinate system $x'o'y'$ regards the connecting line from starting point to end point as axis x' , and then equally divide the axis x' into X_1, X_2, \dots, X_n , through each point X_i we make a perpendicular to the axis x' , then the path node is simplified into one-dimensional coordinates coding, marked as y' and represented as Y_1, Y_2, \dots, Y_n , as shown in Fig.2(a).

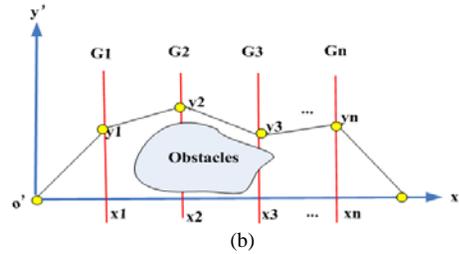


Figure 2. (a) One-dimensional path coding transformed from the original coordinate system. (b) Selection of the initial path.

2) *Population initialization*: Due to the original coordinate system has been transformed, that means the abscissa of each path node is known, we only need to randomly select ordinate y' in each perpendicular line G_i . Considering the selected path nodes may be not feasible, which will determine the evolution speed and the performance of individuals, so we adopted regional obstacles avoidance technology for selecting the path node on each perpendicular line, as shown in Fig.2(b).

3) *Fitness function*: The pathfinding GA select collision avoidance, path length and path smooth as the fitness evaluation factors. fit_1 represents the collision avoidance factor, as shown in Formula (1), where n denotes the number of path node, m denotes the number of obstacles within the sensing range of the agent, d_{ik} denotes the

distance between path node p_i and the center point of the k -th obstacle, S_d denotes the sum of the safety radius for the agent and obstacle.

$$fit_1 = \begin{cases} \min \sum_{i=1}^n \sum_{k=1}^m d_{ik} / S_d, & \text{if } \min(d_{ik}) > S_d \\ 0, 1, & \text{else} \end{cases} \quad (1)$$

fit_2 represents the evaluation of path length, as shown in Formula (2), where $d(p_i p_{i+1})$ denotes the actual length of the path segment, considering the path segment $p_i p_{i+1}$ may collision with obstacles, we introduced a penalty function ω_i to punish the path length. The penalty function is defined in Formula (3), where $\varepsilon (> 1)$ is the penalty factor, which make the length of infeasible path longer.

$$fit_2 = \sum_{i=1}^{n-1} \omega_i \times d(p_i p_{i+1}) \quad (2)$$

$$\omega_i = \begin{cases} \varepsilon & , if \min(d_{ik}) \leq s_d \\ 1 & , if \min(d_{ik}) > s_d \end{cases} \quad (3)$$

fit_3 represents the path smooth, as shown in Formula (4), we adopted turning corner to denote the path smooth, means the smaller turning corner between adjacent path segments, the smoother path.

$$fit_3 = \frac{1}{\sum_{i=1}^{n-1} \left(\frac{y_i - y_{i-1}}{x_i - x_{i-1}} - \frac{y_{i+1} - y_i}{x_{i+1} - x_i} \right)} \quad (4)$$

Finally we can obtain the fitness function of pathfinding GA by multiplication form, as shown in Formula (5), which indicates when the path is shorter, more safety and more smooth, the value of fit will be larger, and the path is more optimal.

$$fit = fit_1 \times fit_3 / fit_2 \quad (5)$$

4) *Genetic operators*: The pathfinding GA adopted four genetic operators, including selection, crossover, mutation and smooth. Selection operator combined tournament selection method with the elite reserve strategy [18]. Crossover operator adopted single point crossover method [11]. For two parent chromosomes, we randomly selected an intersection point (except starting point and end point) and exchange two chromosomes section from the intersection point to the end. Mutation operator adopted a simple mode by randomly selecting a path node and replace it with another non path node. Smooth operator only act on the biggest turning corner, it randomly insert three new path nodes near this corner point, if the new path is feasible, then delete three old path nodes and add new nodes.

The selection of crossover probability and mutation probability directly affects the convergence speed of GA and quality of the final solution, improper adjustment is easy to appear premature convergence phenomenon, thus result in obtaining the local optimal solution. In this paper, we improved adjustment strategy of the crossover probability and mutation probability, and put forward a self-learning fuzzy genetic algorithm. By means of fuzzy logic controller, the proposed algorithm can adaptively adjust the crossover probability and mutation probability for pathfinding GA.

III. DESIGN OF SELF-LEARNING FUZZY CONTROLLER

When performing the pathfinding GA, we adopted fuzzy logic controller to adaptively adjust the crossover probability

and mutation probability. In order to optimize the fuzzy reasoning rules and type of membership functions, we introduced another self-learning GA to get more suitable crossover probability and mutation probability in the pathfinding GA. Here we choose the convergence of population and diversity as input variables of fuzzy controller and normalized, and still choose the crossover probability p_c and mutation probability p_m as output variables. The convergence of population expressed as the difference between average fitness value from two generation of population, and normalized to ensure the range of input variables in [-1,1] (see Formula (6), where N is the population size, $f(k, i)$ is the fitness value of the i th individual at the k th iteration). The population diversity expressed as the dispersion degree of individuals, it can be characterized by the standard deviation of the fitness value. Also in order to characterize the change of population diversity between two iterations, we select difference of the standard deviation as input, as shown in Formula (7). Where $f_{avg}(k)$ is the average fitness value of the k th generation of population. For the input and output variables, we implemented division of fuzzy set, and selection of corresponding type of membership functions. The input variable of ES and AD are all divided into {NB, NS, Z, PS, PB}, respectively for negative big (NB), negative small (NS), zero(Z), positive small(PS), positive big(PB). The output variables of p_c and p_m are divided into {lower, low, medium, high, higher}.

$$ES(k+1) = \frac{\frac{1}{N} \sum_{i=1}^N f(k+1, i) - \frac{1}{N} \sum_{i=1}^N f(k, i)}{\max\left(\frac{1}{N} \sum_{i=1}^N f(k+1, i), \frac{1}{N} \sum_{i=1}^N f(k, i)\right)} \quad (6)$$

$$AD(k+1) = \frac{\sqrt{\frac{1}{N} \sum_{i=1}^N (f(k+1, i) - f_{avg}(k+1))^2} - \sqrt{\frac{1}{N} \sum_{i=1}^N (f(k, i) - f_{avg}(k))^2}}{\max\left(\sqrt{\frac{1}{N} \sum_{i=1}^N (f(k+1, i) - f_{avg}(k+1))^2}, \sqrt{\frac{1}{N} \sum_{i=1}^N (f(k, i) - f_{avg}(k))^2}\right)} \quad (7)$$

We adopted self-learning GA to optimize the knowledge base in the fuzzy controller. Firstly, we used binary bit string for coding the knowledge base [22], the coding with the type of membership functions can be expressed as: 00,01,10,11, represent Triangle (T), Left Triangle (LT), Right Triangle (RT) and Gaussian Function (G) respectively, as shown in Table 1. Each row in Table 1 contains the selection of fuzzy reasoning rules and corresponding type of membership functions. When self-learning GA is running, the corresponding bit string in Table 1 can be translated into the actual fuzzy reasoning rules and type of membership functions, as shown in Table 2. After finished knowledge base coding and translation, we can use self-learning GA to search the optimal fuzzy reasoning rules and corresponding type of membership functions.

TABLE I. KNOWLEDGE BASE CODING

input variables(ES)					input variables(AD)					output variables(pc/pm)					type of membership function
NB	NS	Z	PS	PB	NB	NS	Z	PS	PB	lower	low	medium	high	higher	
1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	00(T)
1	0	0	0	0	0	1	0	0	0	1	0	0	0	0	00(T)
1	0	0	0	0	0	0	1	0	0	1	0	0	0	0	01(LT)
.
0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	11(G)

TABLE II. FUZZY REASONING RULES AND CORRESPONDING TYPE OF MEMBERSHIP FUNCTIONS

rule No.	rule antecedent				rule consequent		membership function type
		ES		AD		pc/pm	
rule-1	if	ES=NB	and	AD=NB	then	pc=lower	Triangle (T)
rule-2	if	ES=NB	and	AD=NS	then	pc=lower	Triangle (T)
.
rule-25	if	ES=PB	and	AD=PB	then	pc=higher	Gaussian (G)

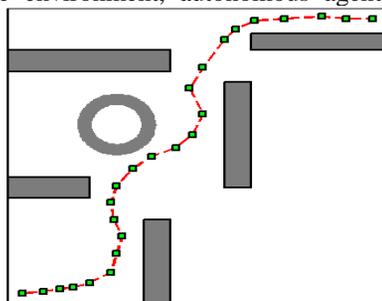
TABLE III. PARAMETER SETTING

	Pathfinding GA	Self-learning GA
Population size	50	20
Path node number	25	×
Maximum iteration number	100	50
Tournament size	5	×
crossover probability	(0.1,1)	0.5
mutation probability	(0.01,0.09)	0.05
Smooth probability	0.1(below 50 generations) 0.5(above 50 generations)	×

V EXPERIMENTAL RESULTS

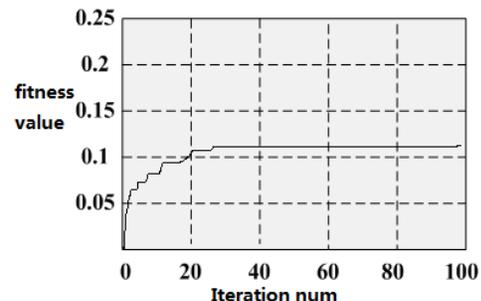
In this section, we evaluate the quality of the proposed TAGA method. We perform experiments on a simulated environment with multiple static and dynamic obstacles, parameter setting is shown in Table 3

For static and dynamic obstacles environment, we run the TAGA method 20 times, and select the best path result, as shown in Fig.3 and Fig.4. Fig.3 is the experimental result in a static obstacle environment, autonomous agent found an



(a)

optimal path without collision, and from the iterative process, about 30 times of iteration to achieve the convergence. Fig.4 is the experimental result in a dynamic obstacle environment, the red circles indicates the movement of the dynamic obstacle, the simulation results show that the autonomous agent also has good adaptability for dynamic obstacle environment, and about 60 times of iteration to achieve the optimal solution.



(b)

Figure 3. (a) Path planning result in a static environment. (b) Iteration process.

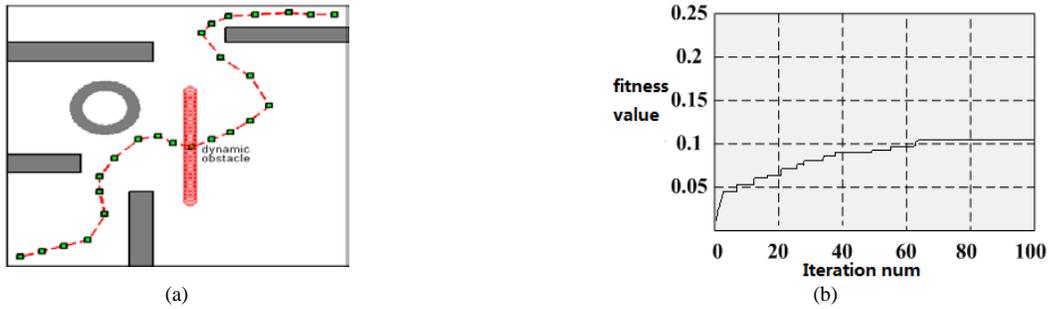


Figure 4. (a) Path planning result in a dynamic environment. (b) Iteration process.

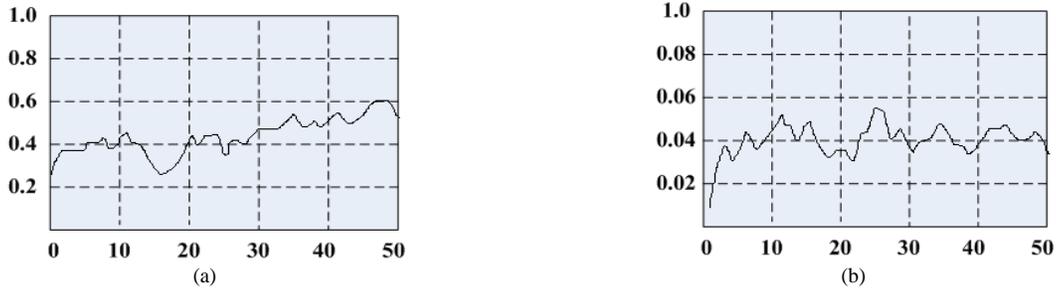


Figure 5. (a) Iteration process of crossover probability. (b) Iteration process of mutation probability.

According to the optimal knowledge base, the corresponding adjustment process of p_c and p_m is shown in Fig. 5. As can be seen, the proposed method can get better adjustment results.

In order to further verify the performance of TAGA, we compared with the standard genetic algorithm (SGA) and the adaptive genetic algorithm proposed by Srinivas (AGA) [8] in the same experiment environment. For SGA the crossover probability=0.8 and mutation probability=0.05; For AGA the parameter k_1 in the crossover probability formula is set to 0.8 and k_2 in the mutation probability formula is set to 0.05. Compared results are shown in Table 4, which presents the path length and running time from above mentioned methods. We performed 20 times experiments and then average. It can be seen that both path length and running time obtained by TAGA are superior to SGA and AGA. This is because TAGA adopted an adaptive crossover probability and mutation probability, which can adjust automatically according to the fitness value. At the same time, by combining with fuzzy controller, TAGA can effectively reduce the number of infeasible solution, accelerate the evolution speed, and more likely converge to the global optimal solution.

TABLE IV. COMPARED RESULTS FOR SGA, AGA AND TAGA.

	SGA	AGA	TAGA
path length/(m)	92.47	86.51	83.32
running time/(ms)	89.84	81.55	79.12

VI SUMMARY

This paper puts forward a two-layer adaptive genetic algorithm, and applied on path planning for autonomous agent. By combining with fuzzy logic, the proposed algorithm realized adaptively adjustment of crossover probability and mutation probability. By introducing another self-learning genetic algorithm, the proposed algorithm realized self-learning optimization for the knowledge base of fuzzy controller, and generated the optimal fuzzy reasoning rules and corresponding type of membership functions. By comparing with SGA and AGA proved that the proposed algorithm has faster convergence speed, and stronger adaptively adjustment ability. The proposed algorithm optimized the fuzzy reasoning rules and type of membership functions in the fuzzy controller, which has a wider adaptability to different optimization problems. Next step is to further improve the self-learning ability of the knowledge base, such as the self-learning of fuzzy control parameters, etc.

ACKNOWLEDGEMENT

This research has been jointly supported by the Science and Technology Program of Guangdong Province under Grant NO. 2013B090500035, and the Science and Technology Program of Zhongshan City under Grant NO. 2013A3FC0284.

REFERENCES

- [1] T.R.Wan, H.Chen, and R. Earnshaw, Real-time path planning for navigation in unknown environment, *Theory and Practice of Computer Graphics*, 2003. Proceedings (2003), pp.138–145.
- [2] R.Allan, S.Willms, and X.Yang, An efficient dynamic system for real time robot path planning, *IEEE Transactions of System, Man and Cybernetics - Part B: Cybernetics* (2006), vol. 36, no. 4, pp. 755–765.
- [3] P. Bhattacharya and M.L. Gavrilova, Roadmap-based path planning-using the voronoi diagram for a clearance-based shortest path, *IEEE Robotics & Automation Magazine* (2008), vol. 15, no. 2, pp. 58–66.
- [4] C.Y. Yang, J.S. Yang, and F.L. Lian, Safe and smooth: mobile agent trajectory smoothing by svm, *international Journal of Innovative Computing, Information and Control* (2012), vol. 8, no. 7B, pp. 4959–4978.
- [5] D. E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*, New York: Addison Wesley (1989).
- [6] J. Maria, A. Pangilinan, and G. K. Janssens, Evolutionary algorithms for the multiobjective shortest path planning problem, *International journal of computer and information science and engineering* (2007), pp. 54–59.
- [7] O. Castillo, L. Trujillo, and P. Melin, Multiple objective genetic algorithms for path-planning optimization in autonomous mobile robots, *Soft Computing - A Fusion of Foundations, Methodologies and Applications* (2007), vol. 11, pp. 269–279.
- [8] M.Srinivas and L.M.Patnaik, Adaptive probabilities of crossover and mutation in genetic algorithms, *IEEE Transactions of System, Man and Cybernetics* (1994), vol. 24, no. 4, pp.656–667.
- [9] H.J.Wang, J.Zhao, and X.Q.Bian, An improved path planner based on adaptive genetic algorithm for autonomous underwater vehicle, *Proceedings of the IEEE International Conference on Mechatronics and Automation* (2005), vol. 2, pp. 857–861.
- [10] X.Du, H.H.Chen, and W.K.Gu, Neural network and genetic algorithm based global path planning in a static environment, *Journal of Zhejiang University. Science* (2005), vol. 6A, no. 6, pp. 549–554.
- [11] G.L.Chen, X.F.Wang, and Z.Q.Zhuang, *Genetic algorithms and applications*, Beijing:The People Post and Telecommunications Press (1999).
- [12] L.Fang, H.C. Zhang, and Y.Z. Jiang, A new fuzzy adaptive genetic algorithm, *Journal of Electronic Science and Technology of China* (2005), vol. 3, no. 1, pp. 57–71.
- [13] Y.Yun and M.Gen, Performance analysis of adaptive genetic algorithm with fuzzy logic and heuristics, *Fuzzy Optimization and Decision Making* (2003), vol. 2, no. 2, pp. 161–175.
- [14] N. E. Hodge, L. Z. Shi, and M. B. Trabia, A distributed fuzzy logic controller for an autonomous vehicle, *Journal of Robotic Systems* (2004), vol. 21, pp. 499-516.
- [15] G.Chrysanthakopoulos, W.L.J. Fox, R.T.Miyamoto, R.J. Marks, M.A.El-Sharkawi, and M. Healy, A fuzzy-logic autonomous agent applied as a supervisory controller in a simulated environment, *IEEE Transactions on Fuzzy Systems* (2004), vol. 12, no. 1, pp. 107–122.
- [16] K.H.Su, L.L.Feng, and Y.Y.Chan, Navigation design with svm path planning and fuzzy-based path tracking for wheeled agent, in *Fuzzy Theory and it's Applications(iFUZZY)*, 2012 International Conference on (2012), pp. 273–278.
- [17] A.Fernandes da V.Machado, U.O. Santos, H. Vale, R.Goncalvez, T. Neves, L.S.Ochi, and E.W.G. Clua, "Real time pathfinding with genetic algorithm," in *Games and Digital Entertainment (SBGAMES)*, 2011 Brazilian Symposium on (2011), pp. 215–221.
- [18] X. E. Liu and X. M. Zhao, Path planning for mobile robot based on multi-sensors, *Computer Engineering*(2008), vol. 34, no. 8, pp. 213–215.