http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 12 (18): 4659-4665, 2013
ISSN 1812-5638 / DOI: 10.3923/itj.2013.4659.4665

Path Planning with Danger Model Immune System

[1]Lihua Zhao and [1,2]Jin Pan
[1]Key Lab of Intelligent Perception and Image Understanding of Ministry of Education, Xidian University, Xi'an, 710071, People's Reublic of China
[2]Xi'an Communications Institute, Xi'an, 710106, People's Republic of China

Abstract: In this study, we present a Danger Model immune algorithm based path planning algorithm (DMIA-PP) for robot path planning. Different with the traditional immune algorithm, the system is not based on self-nonself mechanism, but on danger model theory. In a dynamic environment, if the changes of the obstacles affect the found optimal path, a danger signal will be send to the system. And the memory cell will be stimulated to propagate and to mutate, so as to a new optimization circle. The experimental results have shown the effectiveness and accuracy of the method in a complex searching map.

Key words: Path planning, danger model theory, artificial immune system

INTRODUCTION

In the field of mobile robotics, path planning is an important aspect. The objective of it is to find a collision-free path for a robot to move from a start point to target point in an environment consisting of many movable obstacles. The found path is always desirable to be optimal or near-optimal with respect to distance, smoothness, time and energy (Xiao *et al*., 1997).

Potential Field based planning algorithms are widely used, in which the collision-free path is generated along the negative gradient of the attractive and repulsive functions (Ge and Cui, 2000). The Voronoi diagram concept has also been used in some path planning algorithms (Garrido *et al*., 2011). The main shortcoming of them is that the roadmaps constructed by them are offline maps, the environment information must be known in advance. In some path planning algorithms, the environment was represented by grids and this grid is usually approximated by a graph. Many approaches of soft computing have also been proposed to solve the path planning problems which consists of fuzzy logic, evolutionary computation and neural networks (Banga *et al*., 2011). For example, fuzzy logic and neural network have been used for motion planning of mobile robot. But most of them are computationally very expensive.

Although many algorithms have been proposed to deal with the problem and each method has its own advantage over others in certain aspects, however, most of them tend to be inflexible. The reason lies in the changes of goals, the uncertain environment and the different computer resources. Traditional off-line algorithms assume that the environment is static and wholly known, the objective of the algorithm is to find an optimal path based on some special criteria. On the contrary, the on-line algorithms are often purely reactive and not to find an optimized path (Ferguson and Stentz, 2007). To deal with two problems, we propose a Danger Model Immune Algorithm (DMIA) for robot path planning which can search the solution place efficiently and can deal with changeable goals and environment. The obtained results in simulation have shown the potential of the proposed algorithm.

RELATED WORK

The mobile robot moves in a given environment which can be described by 2-D or 3-D map. And in this map, a finite number of obstacles and via points were given. The objective of path planning is to find a collision-free path between the start point and the goal point. In fact, if the 2-D map includes a large number of via points, the solutions space may be very huge. In general, there are three different goals can be used to design the collision-free optimal path (Nakhaeinia *et al*., 2011): Shortest path, smooth trajectory and clearance requirements.

Although many research has been done to deal with this problem, there still existed a lot of factors which hinder conventional algorithms to be used. For example, the changes of goals and different optimization goals need more flexible searching algorithms, uncertain environment make the searching progress more complex, different constraints and computational resources need good flexibility for an algorithm.

Corresponding Author: Lihua Zhao, No. 8, Zhangba East Road, Xi'an, 710065, China

Danger model immune algorithm: Artificial immune system is a distributed, parallel, self-adapted system which is based on the biological immune system (Xu *et al*., 2012). Farmer introduced the applications of artificial immune system in 1986 (Farmer *et al*., 1986). Since then, AIS system has gained more and more attentions. However, many medical researchers have discovered that the self-nonself mechanism served as the foundation of traditional AIS is unsuited to resolve many problems. So, in 1994, Matzinger proposed the famous danger model theory which declared that the biological immune system reacts to danger signals released by unnatural death cell, not to the discrimination of self and non-self (Matzinger, 1994; Matzinger, 2002). From then on, many scholars have studied the danger model theory and the theory was used in many different areas. Aickelin proposed an algorithm to solve intrusion detection problems, in the algorithm the behavior of dendritic cell was introduced (Aichelin and Cayzer, 2002). In his later articles, Aickelin compared danger theory with traditional self-nonself mechanism used in AIS. The danger model theory was also been used in web mining and classification. In fact, the Danger model believes that the immune responses are triggered by danger signals. Then, it gives a different method to partition the antigens: Some of them associated with danger and some of others don't. The antigens associated with danger should be dealt with, others should be tolerated. Under this model, self need not be ignored and non-self need not always be attacked. This makes the body changes, matures, procreates and growing possible. The danger model expands the innate immune system and allows us to live without keeping a rigid sterility which segregates us from environment.

Path costs: Let p_i be a point in the environment map, the found path linking the initiating point p_s and the terminal point p_t can be represented by ordered points in the lattice space:

$$P = \{p_S, p_1, p_2, \dots p_t\} \tag{1}$$

where, the set of points are contiguous and no repeated points in it. In general, the cost of a given path is an arbitrary function of the path, F(P). To make it easy, we assumption that the cost function is comprised of three parts:

$$F(P) = \sum_i A(p_i) + \sum_i B(p_i, p_{i+1}) + \sum_i C(p_i, p_{i+1}) \tag{2}$$

$A(p_i)$ is the traversing cost of the point p_i. It should be set according to the domain characteristics. In general, each of the points in the set has the same traversing cost.

$B(p_i, p_{i+1})$ presents the cost come from clearance requirements. If the point p_{i+1} near an obstacle, additional cost should be added to the cost.

$C(p_i, p_{i+1})$ presents the cost come from the requirements of smooth trajectory. If a turning appears between points p_i and p_{i+1}, additional cost should be added to the cost function too.

In fact, the path length can be taken into account by assigning B and C to a proportional distance.

PATH PLANNING WITH DANGER MODEL IMMUNE ALGORITHM

The environment map is defined as a discrete lattice Ld. Each of the lattice was represented by a point p_i. The variables in DMIA were defined just like in variables of convention AIS. Antigen corresponds to the objective function and the affinities of antibodies were defined as the matching degree between the optimization roads and the objective function. The found optimal path P is a path represented by ordered points in the lattice space, each of these points p_i was not been occupied by an obstacle. Let p_i be a lattice in the environment. If p_i is occupied by an obstacle, the value of it was set to 1. Otherwise, its value was 0. In the immune response, the danger signal plays a key role in the danger theory. We can define the danger signal as follows:

- **Definition 1:** (The optimal path) The optimal path is a path P^M, $P^M=\{p_1, p_2, \dots, p_i\}$, where $p_i = 0$ and $i \in N$
- **Definition 2:** (Danger signal) In a dynamic environment, if the environment changes were perceived and a point p_i (x_0, y_0) in the found optimal path P^M was occupied by an obstacle, the system will send a danger signal s = 1. Otherwise, s = 0
- **Definition 3:** (Danger area) Let danger signal s = 1. Let R be the radius of danger area. Let point p_i (x_0, y_0) in P^M was the point occupied by an obstacle. The positions of the points $p_{j(x,y)}$ were defined as danger area, where $(x_0\text{-}R < x < x_0\text{+}R$ and $y_0\text{-}R < y < y_0\text{+}R)$
- **Definition 4:** (Danger mutation) Different with traditional mutation operators, the Danger mutation operator was mutation operators which were permitted to carried out only in the danger area. Let $P = \{p_1, p_2, \dots, p_j, \dots, p_i\}$ be a antibody and the value of point p_j is 1. Then, the mutation operator appeared in the range $[p_{j\text{-}R}, p_{j\text{+}R}]$ were defined as Danger mutation operator

In the proposed algorithm, the selection, crossover and mutation operators are used to update the population. The framework of the DMIA is shown if Fig. 1. The

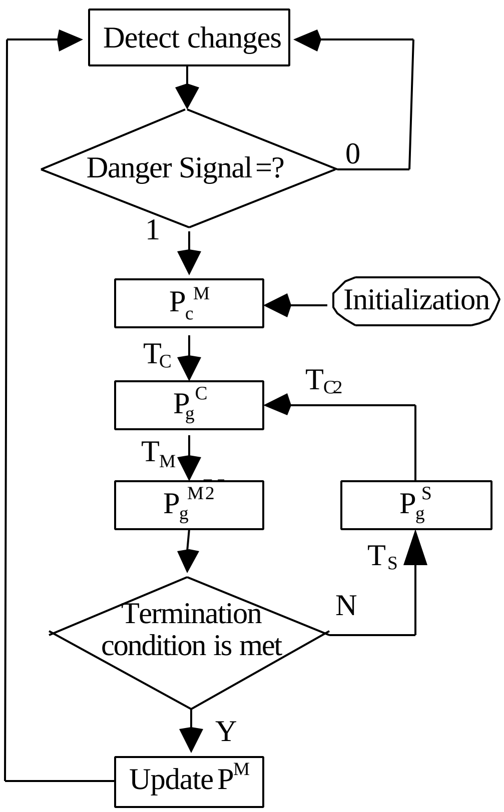

Fig. 1: Procedure of the DMIA-PP algorithm

danger area was used to limit the scope of the crossover and mutation operators. The found optimal path was put into memory and was named memory cell. The flow of the proposed algorithm is described as follows:

Step 0: Detect the changes of the environment. If a point p_i in the found optimal path P^M was occupied by an obstacle, then s = 1. Otherwise, s = 0

Step 1: Initialize the termination condition, clonal probability T_c, selection probability T_S, T_{S2}, mutation probability T_M, generation counter g (g = 1), change counter c (c = 1) and danger signal s (s = 1). Initialize the memory cell P^M which contains a series of via points $p_i(x, y)$ in a path

Step 2: If signal s = 1, then c = c+1 and go step 3; else, go step 0

Step 3: Proliferate N clones for P^M and form the population P^C_g, $P^C_g = T_c(P^M) = \{P^{1,C}_g, P^{2,C}_g, \ldots, P^{N,C}_g\}$

Step 4: Mutate each clone in population P^C_g with danger mutation operator and delete the same one, form the population P^{M2}_g, $P^{M2}_g = T_s(T_m(P^C_g)) = \{P^{1,M2}_g, P^{2,M2}_g, \ldots, P^{N,M2}_g\}$. In population P^{M2}_g, the affinity of antibody $P^{i,M2}_g$ is calculated like this:

$$A_{ffinity}(P^{i,M2}(g)) = \sum_j A(p_j^{i,M2}) + \sum_j B(p_j^{i,M2}, p_{j+1}^{i,M2}) + \sum_j C(p_j^{i,M2}, p_{j+1}^{i,M2}) \quad (3)$$

where, $A(p_j^{i,M2})$ is the traversing cost of the point $p_j^{i,M2}$. In this article, it is set to 1. $B(p_j^{i,M2}, p_{j+1}^{i,M2})$ presents the cost come from clearance requirements. If the point $p_{j+1}^{i,M2}$ near an obstacle, additional cost should be set to 1, otherwise, $B(p_j^{i,M2}, p_{j+1}^{i,M2}) = 0$. $C(p_j^{i,M2}, p_{j+1}^{i,M2})$ presents the cost come from the requirements of smooth trajectory. If a turning appears between points $p_j^{i,M2}$ and $p_{j+1}^{i,M2}$, additional cost should be set to 1. Otherwise, $C(p_j^{i,M2}, p_{j+1}^{i,M2}) = 0$

Step 5: If termination condition is met, the antibody $P^{i,M2}_g$ with the largest affinity value in population P^{M2}_g is the optimal path and will become a new memory cell. Otherwise, g = g+1 and go step 6. The termination condition adapted for the algorithm is defined like this: the optimal antibody in P^{M2}_g keeps unchanged for continuous g* generation

Step 6: Select paths in population P^{M2}_g and form the population P^S_g, $P^S_g = T_s(P^{M2}_g)$. The selection operation T_s is defined by the following: Sort the paths in the population P^{M2}_g according to their affinity value and choose the first R highest affinity paths to form the population P^S_g

Step 7: Proliferate N_i clones for each paths $P^{i,S}_g$ (i = 1, 2,..., Q) in the population P^S_g and form the population P^C_g, $P^C_g = T_{c2}(P^S_g) = \{P^{1,S}_g, P^{2,S}_g, \ldots, P^{N,S}_g\}$. The number N_i is decided by the following equation, Q is the size of population P^C_g. Go step 4

$$N_i = Q \times A_{ffinity}(P^{i,M2}_g) / \sum_{j=1}^{N} A_{ffinity}(P^{i,M2}_g) \quad (4)$$

EXPERIMENTS

In this section, we have conducted several experiments on a 3.0 GHz Pentium PC with 512MB of memory running with Microsoft Windows XP to measure the performance of the proposed approach.

Static environment: We implemented the proposed DMIA-PP algorithm for polygonal obstacles to test its tuning ability and overall performance in static environment. Figure 2 shows the searching procedure. The green and red circles are Initiating and terminal

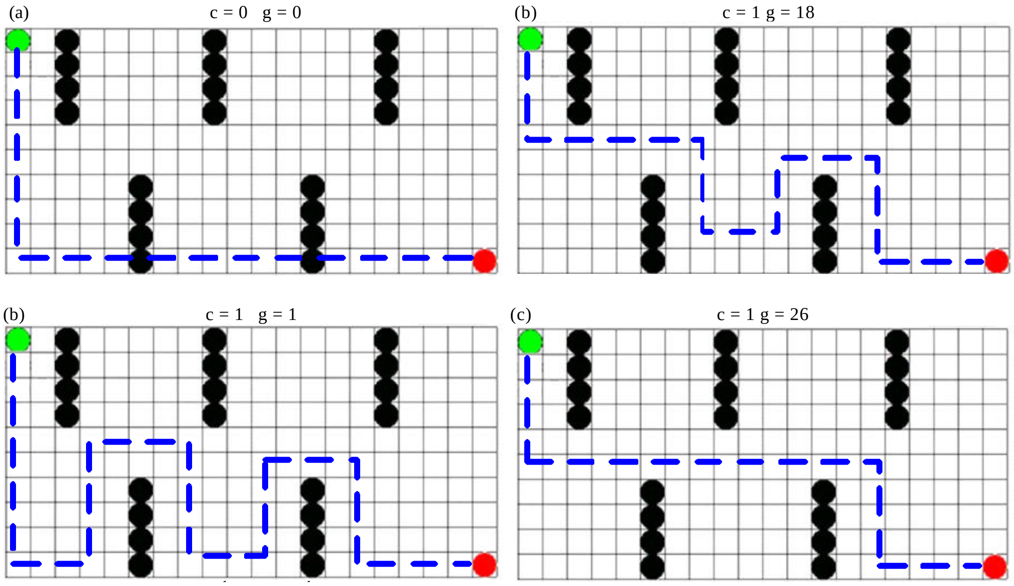

Fig. 2(a-d): Performance results on five obstacles

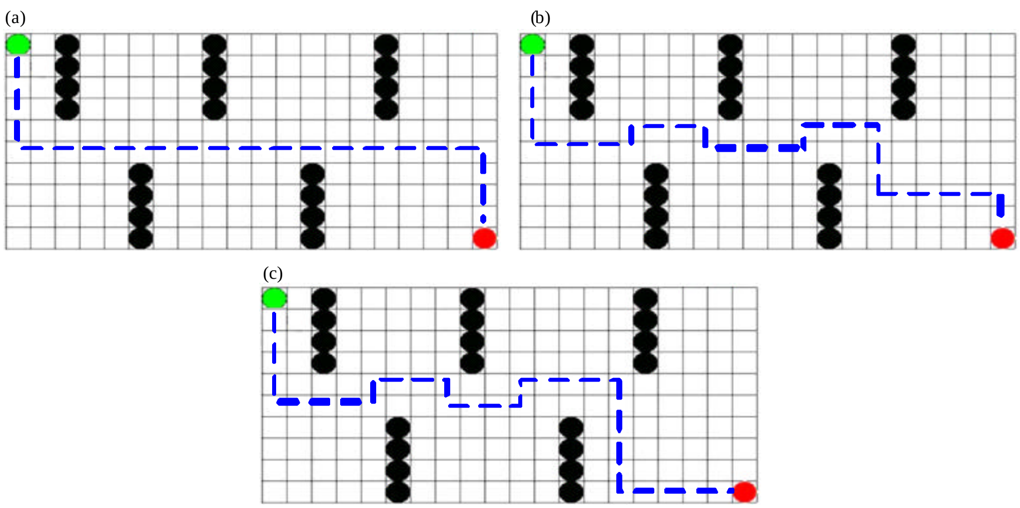

Fig. 3(a-c): Results on different optimal criterions

points. At first, we randomly set an inappropriate path P_0^M for memory cell and set the danger signal s to 1. So, the automatic optimization progress was activated. It can be seen in Fig. 2 that when the change counter c was 0 and the generation counter g was 0, an inappropriate path P_0^M was given. After one generation of evolution, a feasible path appeared. But the path was not an optimal one. Then, with the development of optimization progress, a more feasible path appeared. Finally, the algorithm stopped when the generation counter g is 26 and the found path is one of the most feasible paths. In this example, the shortest path was seen as optimal path.

In order to examine the effects of different optimal criterions, we further give some experiments which can be seen in Fig. 3. If we make the shortest path as the optimal path, we can get many optimal paths. One of them was shown in Fig. 3a. However, if we consider the requirement of clearance, one of the feasible paths was shown in Fig. 3b. The optimization criterion of it was set to "shortest path" + "clearance requirements". If we further ask for smooth trajectory or a path with minimum turns, we can get the optimal paths shown in Fig. 3c.

Dynamic environment: We have implemented an experiment for online navigation with the following

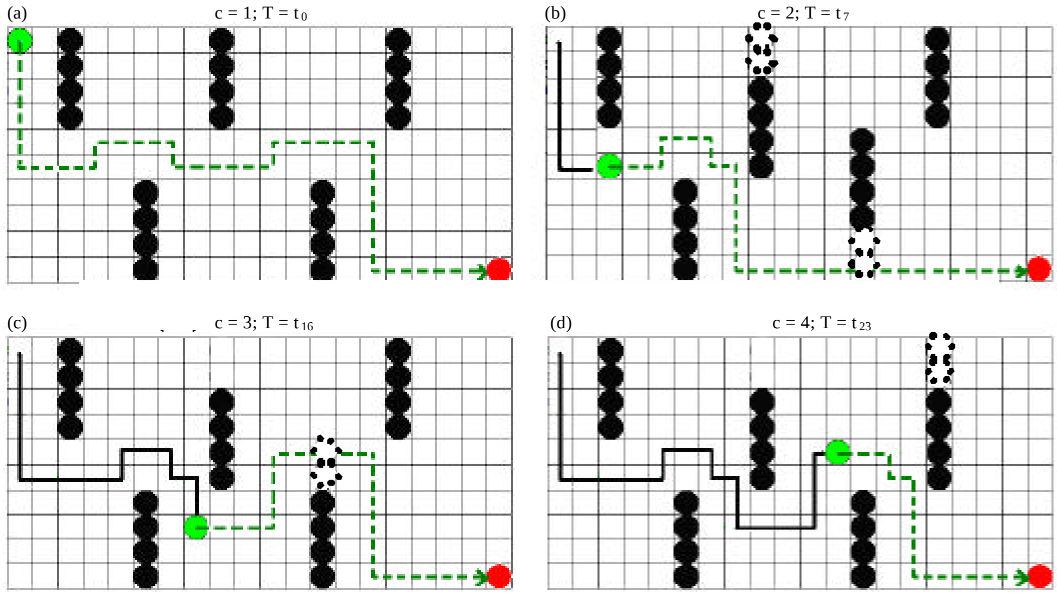

Fig. 4(a-d): On-line navigation-1

assumptions: (1) Obstacles in the environment are known. But with the time passes by, all of them were permitted to move. (2) Every times the environment changes were perceived, the robot will re-plan its path from the current position to the goal position. (3) The optimal criterion we adopted is compound one which contains the requirements of shortest path, clearance and smooth trajectory.

We can see the process of path planning in a dynamic environment. Firstly, the robot plans its path map at the start point. This procedure is a static process. After 20 generations, we get the optimal path P^M which was shown in Fig. 4a. Then, the robot began to move. At time t_7, two of the five obstacles in the environment moved into the positions where memory cell P^M occupied and this was perceived by the robot. So, as to the danger signal s was set to 1 and the robot has to re-plan its new path map with the current position and the changed environment. We can see that in Fig. 4b. At time t_{16} and t_{23}, the environment E1 kept changing and the robot re-planed its path accordingly.

Another experiment was also been carried out in another environment E2 which can be seen in Fig. 5. At time to, the robot calculated its path at the start point. And then, it gone to position (1, 7). Having sensed the changes of the obstacles at time t_7, t_{16} and t_{22}, the robot changed its path maps accordingly.

Different danger radius: Danger area is an important parameter in DMIA and the scope of it was decided by danger radius R. The effects of the danger mutation operator Tm are largely decided by the danger radius R. In this section, we implement an experiment to compare the effects coming from different danger radiuses R. Figure 6 shows the running time to find paths with different R. In the two environments, when the danger radius R was set to a large vale (for example, R = 16), we found that the running times of the two algorithms are very similar. However, when the danger radius R was set to a small one, the efficiency of DMIA-PP algorithm is higher than that of AIS-PP algorithm. The reason is that too large danger radius will enlarge the range of the danger mutation operator so as to an inefficient searching.

Algorithms comparison: Another experiment was conducted to compare the proposed DMIA-PP algorithm with the traditional AIS based path planning algorithm. Figure 7 shows the environment and the static obstacles which were presented by black squares. The dynamic obstacle $O_1O_2O_3O_4O_5O_6O_7O_8$ was presented by blue squares which has the ability of randomly crawling in the given environment map. Every times a discrete time step arrives, the obstacle $O_1O_2O_3O_4O_5O_6O_7O_8$ can select to crawl forward a square or to keep unchanged. The green square was the start point of the robot and the red one was the goal point. After 20 times running, 40 paths were found by the two algorithms. The average path costs of unit distance, direction changes and obstacles were shown in Table 1, from which we can get the total path

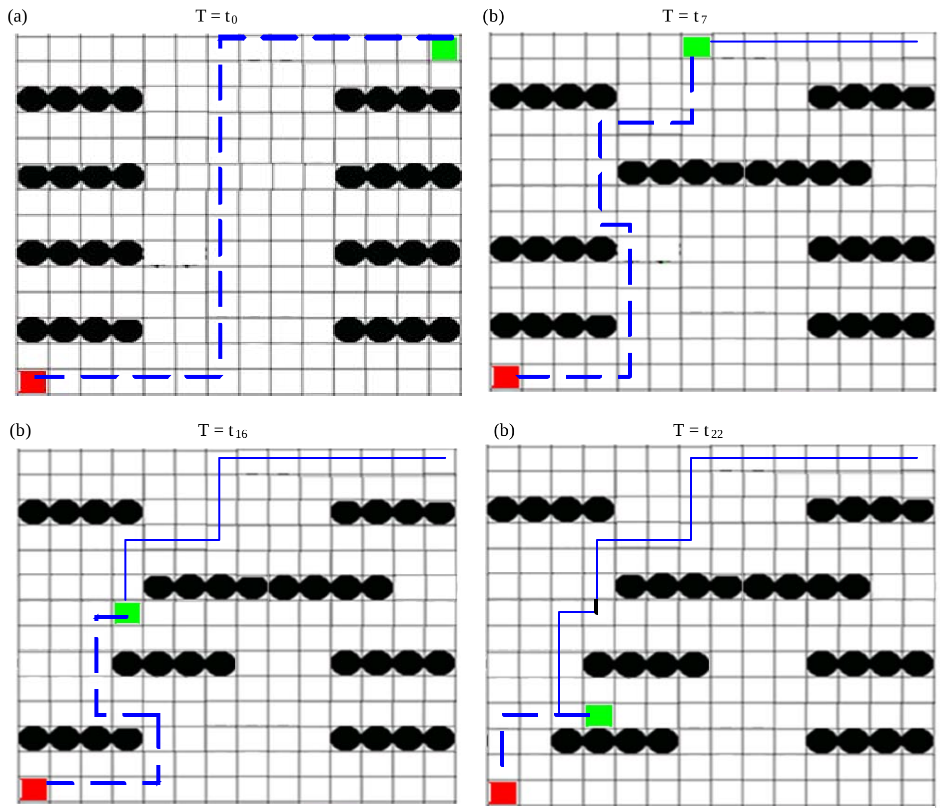

Fig. 5(a-d): On-line navigation-2

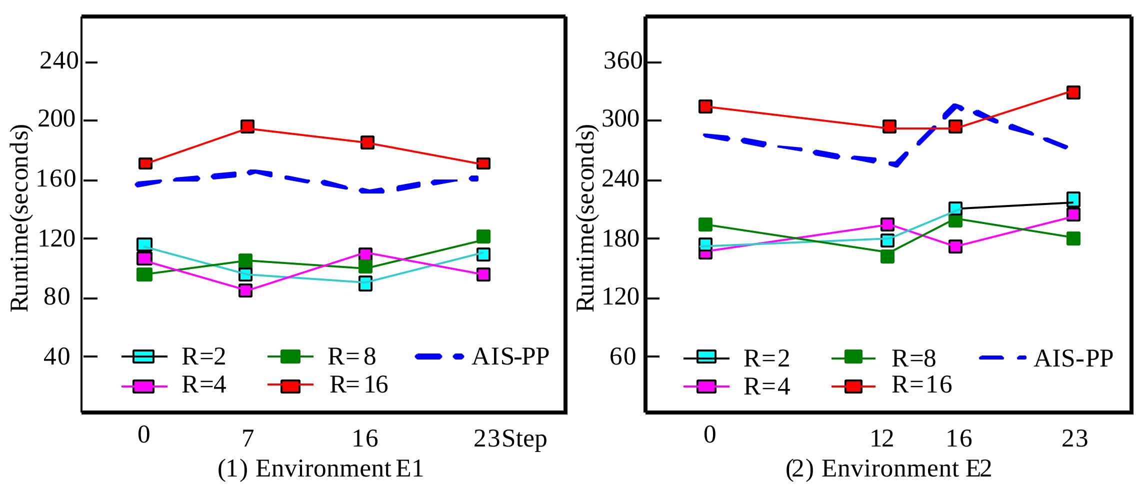

Fig. 6(a-b): Runtime of the optimization process with different danger radius

Table 1: Average costs generated by AIS-PP algorithm and DMIA-PP algorithm

	Unit distance	Direction changes	Obstacles	F(P)
AIS-PP	49	9	7	65
DMIA-PP	46	7	6	59

cost F(P). In this experiment, the index of direction changes B was set to 1 and the index of obstacles C was set to 1 too. In AIS based algorithm, the population size, clonal rate and mutation rate were separately set to 50, 0.9,

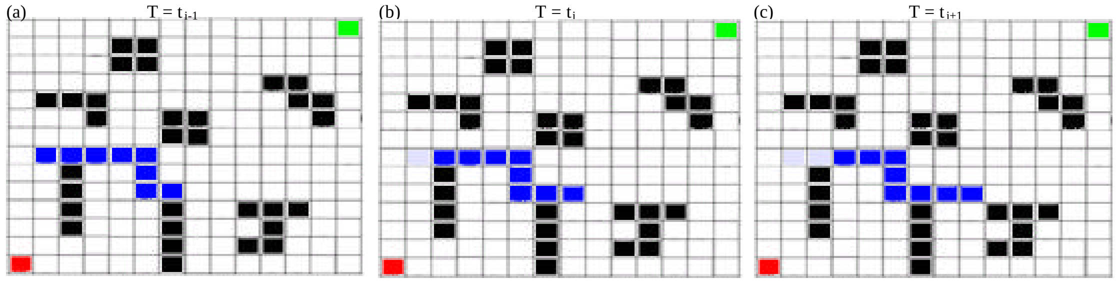

Fig. 7(a-c): Algorithms comparison

0.2. For making a fair comparison, we terminated the AIS-based algorithms when the time it cost is the same with DMIA-PP.

From the Fig. 7 and Table 1, we can see that the average path cost of unit distance, direction changes and obstacles generated by algorithm DMIA-PP were always cheaper than that generated by AIS-PP. The total costs of the two algorithms were 49+9+7 = 65 and 46+7+6 = 59. What makes the two searching results so different? The reason lies in the different mutation manners. In AIS-PP, the area of the mutation operator was the whole path. However, for a dynamic obstacle, it always affects only a fragment of the path. In DMIA-PP, every time a danger signal was given, the danger mutation operator was limited only in the danger area. That made the search progress more efficient than that of AIS-PP algorithm.

CONCLUSION

In this study we represent the DMIA-PP algorithm which is based on danger model theory for robot path planning. The performance result has shown that the DMIA-PP has the capability of dealing with static and dynamic obstacles and achieve good accuracy and time performance in comparison with conventional AIS-PP algorithms. In the future, we will make the algorithm more suitable for on-line navigation.

REFERENCES

Aichelin, U. and S. Cayzer, 2002. The danger theory and its application to artificial immune systems. Proceedings of the 1st International Conference on Artificial Immune Systems, September 9-11, 2002, Canterbury, UK., pp: 141-148.

Banga, V.K., R. Kumar and Y. Singh, 2011. Fuzzy-genetic optimal control for robotic systems. Int. J. Phys. Sci., 6: 204-212.

Farmer, J.D., N.H. Packard and A.S. Perelson, 1986. The immune system, adaptation and machine learning. Physica, 22: 187-204.

Ferguson, D. and A. Stentz, 2007. Field D: An Interpolation-Based Path Planner and Re-Planner. In: Robotics Research: Results of the 12th International Symposium ISRR, Thrun, S., R.A. Brooks and H. Durrant-Whyte (Eds.). Springer, New York, pp: 239-253.

Garrido, S., L. Moreno, D. Blanco and P. Jurewicz, 2011. Path planning for mobile robot navigation using voronoi diagram and fast marching. Int. J. Robotics Automation, 2: 42-64.

Ge, S.S. and Y.J. Cui, 2000. New potential functions for mobile robot path planning. IEEE Trans. Robot. Autom. Soc., 16: 615-620.

Matzinger, P., 1994. Tolerance, danger and the extended family. Annu. Rev. Immunol., 12: 991-1045.

Matzinger, P., 2002. The danger model: A renewed sense of self. Science, 296: 301-305.

Nakhaeinia, D., S.H. Tang, S.B.M. Noor and O. Motlagh, 2011. A review of control architectures for autonomous navigation of mobile robots. Int. J. Phys. Sci., 6: 169-174.

Xiao, J., Z. Michalewicz, L. Zhang and K. Trojanowski, 1997. Adaptive evolutionary planner/navigator for mobile robots. IEEE Trans. Evolutionary Comput., 1: 18-28.

Xu, Q., S. Wang and C. Zhang, 2012. Structural design of the danger model immune algorithm. Inform. Sci., 205: 20-37.