An Immune Genetic Algorithm for AUV Local Path Planning

Hongli Xu and Yiping Li
State Key Laboratory of Robotics, Shenyang Institute of Automation, Chinese Academy of Sciences
Shenyang, Liaoning Province, China

ABSTRACT

This paper proposes a new immune genetic algorithm and applies it to generate local paths for an autonomous underwater vehicle (AUV) when it fails to avoid abrupt obstacles by reactive behaviors. The algorithm mixes both immune algorithm and genetic algorithm through introducing niche technology based on path segment number, maintaining population’s diversity by population clustering, enhancing local search capability and convergence velocity through cell clone operation and immune memory mechanisms. The simulation results verify that the proposed algorithm is not only more efficient than immune algorithm and genetic algorithm, but also feasible and effective in a typical semi-closed obstacle scenario.

KEY WORDS: AUV; path planning; immune genetic algorithm; obstacle avoidance.

INTRODUCTION

Obstacle avoidance and path planning are two necessary capabilities for an autonomous underwater vehicle (AUV) to operate in an unknown complex environment without human’s intervention. The objective of obstacle avoidance is to generate a series of actions on the basis of sensorial information. Path planning is mainly to find an optimal or hypo-optimal safety path from start point to goal point according to performance criteria (Zu, 2007). They are different but correlative each other. Obstacle avoidance is usually implemented by a reactive control law in the lower level of control system and may get into local extremes in the absence of apriori information and global map. But path planning can find an optimal obstacle-free path even in an entrapment and then direct AUV to break the deadlock of local extremes. We have presented a real-time obstacle avoidance method based on fuzzy control in the reference (Xu, 2008). And this paper mainly discusses the local path planning problem.

In recent years, researches on AUV local path planning are focused on modified virtual field force algorithms (Jiao, 2007), evolutionary algorithms (Alvarez, 2004), mixed integer linear programming methods (Yilmaz, 2005) and numerical nonlinear programming techniques (Petillot, 2001). It is widely accepted that evolutionary algorithms, especially genetic algorithms are more applicable to solve complex multi-objective optimization problems such as path planning (Chang, 2005). Genetic algorithm originates from the Darwinian theories of natural selection and survival. It is categorized as global search heuristics used to find exact or approximate solutions to optimization problems. Its computational complexity has been proved to increase linearly with the dimension of the solution space (Kanakakis, 2007). In the case of AUV real-time path planning, environments are space- and time-varying, and its solution space is a four-dimensional space. Then traditional genetic algorithm can not guarantee convergence to an optimal solution in limited time.

This paper introduces immunological theories into genetic algorithm and proposes a new immune genetic algorithm, which inherits all advantages from them. There are three main improvements. Firstly a niche technology based on path segment number is presented to reduce solution space of search. Then we apply antibody population clustering to maintain population’s diversity. At last local search capability and convergence velocity are enhanced through cell clone operation and immune memory mechanisms. The simulation results verify these improvements finally.

THE NICHE GENETIC ALGORITHM

In Ecology niche is named as a habitat supplying the factors necessary for the existence of an organism. It means that an organism strongly inclines to live with ones that have the same characters and behaviors. In this paper we introduce the niche concept to reduce solution space of search.

During searching an optimal path connecting a start point with a goal point, the segment number of the optimal path is unsure in advance, that may be one at least and may be infinite at most in theory. But in practice the shortest length of a path segment is limited by the AUV’s maneuverability and mission requirements, so that the path segment number is finite for a specific planning task. For this reason, the paths having the same segment number may form a niche. We can determine the number of niches as \( M(M > 2) \). Then the population \( P(t) \) will be divided into \( M \) son-populations.

\[
P(t) = \{P_1(t), \ldots P_m(t), \ldots, P_M(t)\}, \quad m = 1, 2, \ldots, M
\]

In every son-population \( P_m(t) \) each path comprises of \( m \) path segments. A path corresponds to a chromosome \( P_{ai} \) and a point is
represented by a gene $p^*_{k,i}$.

\[ P_m = \{ P_{m,1}, P_{m,2}, \ldots, P_{m,n} \} \]

\[ P_M = (p^*_{k,1}, p^*_{k,2}, \ldots, p^*_{k,j}, \ldots, p^*_{k,n}) \]

\[ p^*_{k,i} = (x_{i,0}, y_{i,0}), \quad x_{i,0} \in [x_{min}, x_{max}], \quad y_{i,0} \in [y_{min}, y_{max}] \]

where $m = 1, \ldots, M$, $k = 1, 2, \ldots, N_m$, $l = 1, 2, \ldots, m$, and

\[ N = \sum_{m=1}^{N_m} N_m. \]

All the evolutionary operations will be only used in son-population, so $p^*_{k,i}$ can be abbreviated to $p_k$. In general the total length of the path, the safety of trajectory and the smoothness of trajectory are three criterions for evaluating the quality of the planning paths. In order to navigate AUV through complex hazards, these three fitness functions are revised for real-time obstacle avoidance.

The Total Length Measure

The total length of a path $P_{ab}$ is defined as follows:

\[ f_1(P_{ab}) = \sum_{j=0}^{n} d(p_{j}, p_{j+1})/d(p_{j}, p_{k}) \]

where $p_j$ and $p_k$ denote the start point and the goal point respectively, and $d(p_{j}, p_{j+1})$ denotes a distance between two adjacent path points.

When no forbidden area crosses between the start point and the goal point, an optimal path just is the connection line of these two points and $f_1(P_{ab}) = 0$. But actually the total length of an optimal path is longer than that of the ideal path because of obstacles or mission requirements. So we can conclude $0 \leq f_1(P_{ab}) < \infty$.

The Safety Measure

The safety of a path $P_{ab}$ is measured by two penalty values:

\[ f_2(P_{ab}) = f_2(P_{ab}) + f_3(P_{ab}) \]

\[ f_2(P_{ab}) = \sum_{j=0}^{n} b(p_{j}, p_{j+1}, \Omega_k) \]

where $b(p_{j}, p_{j+1}, \Omega_k)$ denotes the penalty value for the path segment from the point $p_j$ to the point $p_{j+1}$, if it is blocked by any obstacle.

Given regions of obstacles $\Omega_1, \ldots, \Omega_k, \ldots, \Omega_g$ have been recognized and the penalty value is:

\[ b(p_{j}, p_{j+1}, \Omega_k) = \begin{cases} 0 & \text{if } d(p_{j}, p_{j+1}, \Omega_k) \geq d_{min} \\ \Omega_k & \text{otherwise} \end{cases} \]

in which $d_{min}$ is a permitted shortest distance between AUV and obstacle. $d(p_{j}, p_{j+1}, \Omega_k)$ is the shortest distance between the path segment $p_{j}, p_{j+1}$, and the obstacle $\Omega_k$.

\[ f_3(P_{ab}) = \sum_{j=0}^{n} \delta \left( g(p_{j}, p_{j+1}, T_{av}) \right) \]

where $g(p_{j}, p_{j+1}, T_{av})$ denotes the penalty value for the crossover between the path $P_{ab}$ and AUV trajectory $T_{av}$.

The Smoothness Measure

The third criterion is mainly used to guarantee a smooth path that can be maintained by the AUV according to the following formula:

\[ f_3(P_{ab}) = \sum_{j=1}^{\infty} \frac{\psi_j}{\pi} \]

where $\psi_j \in [0, \pi]$ denotes the angle between the path segment $p_{j-1}, p_j$ and the extension of the path segment $p_{j}, p_{j+1}$ on a plane determined by both above segments.

Assume that AUV holds the same depth when it is following a path segment, then AUV path planning can be formulated as a multi-objective optimization problem in a two dimension space. The algorithm aims at finding a short, safety and smooth path for AUV. The optimal solution can be considered by sum of all the fitness values:

\[ J(P_{ab}) = \sum_{j=1}^{\infty} \omega_j f_j(P_{ab}) \]

The Added Immune Operations

In addition to three basic genetic operations, including selection, crossover, and mutation, this paper presents three immune operators such as population clustering, cell clone and immune memory.

Population Clustering

In the first place the definitions of affinity, similarity, concentration and stimulation are introduced.

Affinity is defined as the matching degree of antibody and antigen (Huang, 2005), $A: P \rightarrow [0,1]$ . In this algorithm, the fitness value difference with current optimal solution $B_0$ represents the affinity of an antibody, as follows:

\[ \Delta(P_{ab}) = \frac{1}{1 + 1/J(P_{ab}) - \min_{j=1}^{\infty} J(P_{ab})} \]

The similar degree of two antibodies is defined as:

\[ M(P_{ab}, P_{ab+1}) = \sum_{j=1}^{\infty} d(p_{j,1}, p_{j,2}) \]

One path is same as the other only if all the distances between the corresponding points of two paths are zero.

The antibody concentration denotes the proportion of similar antibodies in the whole population, and is defined as:

\[ C(P_{ab}) = \frac{|P_{ab} \in X | |M(P_{ab}, P_{ab}) \leq \delta|}{N_{\text{ab}}} \]

This formula indicates that $C(P_{ab}) = 1/ N_{\text{ab}}$ when no antibody is similar with $P_{ab}$ and $C(P_{ab}) = 1$ when all the antibodies are similar with $P_{ab}$ in population $X$. 

397
The stimulation intensity of an antibody represents the ability of 
stimulation by an antigen and the other antibodies and is defined as 
\( F: X \subset P_n \rightarrow \mathbb{R}^+ \),
\[
F(P_{x}) = A(P_{x}) e^{-(C(P_{x})/\beta)}
\]
where \( \beta \) is a regulatory factor and \( \beta \geq 1 \).

The clustering operation is used to get rid of similar antibodies 
and maintain the population’s diversity. Given population \( P \):
\[
P = \{ P_1, P_2, \ldots, P_{mN} \}
\]
\( P \) is divided into \( q \) son-populations in which the \( k \)-th son-population \( Q_k \) is expressed by:
\[
Q_k = \{ P_{i_1}, P_{i_2}, \ldots, P_{i_{q_k}} \}, \forall P_{i_k}, P_{i_{k+1}} \in Q_k, \; M(P_{i_k}, P_{i_{k+1}}) \leq \delta
\]
According to the inequation:
\[
F(P_{i_k}) - F(P_{i_{k+1}}) \leq \delta, \; P_{i_k}, P_{i_{k+1}} \in Q_k
\]
one with the lower stimulation is punished by a positive value in \( P_{i_k} \) 
and \( P_{i_{k+1}} \). Finally the individuals having smaller penalty values 
are selected as the clustering results.

**Cell Clone**

Cell clone is a mapping from an antibody to a set of clones under the 
defined number of reproduction. It provides \( N_{i_n} \) clones propagated by 
each antibody with a super mutation operation with all the antigens.
The other clones will be used by a uniform mutation operation with one 
randomly selected from the antigen population. There is an antibody population \( X \) 
and an antigen population \( Y \), assume that \( X_i \in X \) and \( Y_i \in Y \). The reproduction number of the 
antibody \( X_i \) is expressed by \( N_{i_n} \):
\[
N_{i_n} = \left( \frac{1}{\lambda A(X_i)} \right)^{\theta}, \; \lambda \in \left[ \frac{1}{2(1+A(X_i))}, \frac{1}{1+A(X_i)} \right]
\]
where \( \lambda \) is a random number, that represents the reproduction ratio 
of the antibody \( X_i \), and \( \theta \) is a defined constant, \( 1.0 < \theta < 1.5 \).

The super mutation operation of the antibody \( X_i \) and the antigen \( Y_i \) 
is illustrated by the formula:
\[
X_i \leftarrow X_i + \beta(Y_i - X_i), \; \beta \in [0, \alpha], \; \alpha = 1 - e^{1/\lambda A(X_i)}
\]
where \( \beta \) is a random number in \([0, \alpha]\).

The uniform mutation operation of \( X_i \) and \( Y_i \) is namely that the genes 
of \( X_i \) is randomly changed at the mutation ratio \( \alpha_n \):
\[
\alpha_n = 1 - e^{1/\lambda A(X_i)}
\]

**Immune Memory**

During the initial stage of evolution, there always generate a lot of low-
fitness and unfeasible solutions in a population. This has a bad effect on 
search efficiency of the path planning method. So we present an 
immune memory mechanism. The definition of vaccine is borrowed to 
represent good gene. In initialization we can use some apriori 
information such as the online generated map and AUV trajectory to 
initialize the vaccine set \( M_{i_n}(t) \). \( M_{i_n}(t) \) is updated by the genes of the 
optimal solutions with the evolution of the son-population \( P(t) \). It will 
be saved at the end of evolution and be added in the initial vaccine set 
of next son-population \( P(t) \). So the good genes are memorized by the

The former genetic algorithms generally use delete operator or insert 
operator to modify unfeasible solutions. But it’s low effective when 
there are too many unfeasible solutions in evolutionary population. 
In the proposed immune genetic algorithm, vaccination is applied to make 
all the solutions feasible and increase the adaptability of a solution.

**PROCESS OF THE IMMUNE GENETIC ALGORITHM**

There are many types for integrating genetic algorithm and immune 
algorithm. This paper introduces the above immune mechanisms into 
the niche genetic algorithm and proposes a new immune genetic 
algorithm, whose process is described as the following sixteen steps.

1. If the connection line of the start point and the goal point meets 
   the safety measure, it is the optimal solution \( p^*_k \) and 
   the process ends. If not meet, set \( m = 2 \) and initialize the vaccine 
   set \( M(t) \) and the max of path segment number \( M \).
2. Set \( N_{\text{ve}} \) — the size of the son-population \( P\alpha(t) \), \( T_n \) — the max 
   generation of evolution, \( N_{\text{ve}} \) — the size of the antigen 
   population \( B\alpha(t) \), and \( t = 1 \).
3. Read \( M\alpha(t) \) from the memory and randomly generate \( N_{\text{ve}} \) 
   antibodies making up of \( P\alpha(t) \).
4. Calculate the fitness values of each antibody in \( P\alpha(t) \).
5. Select the antibody with max fitness value as a antigen candidate and 
   perform the population clustering operation on it and \( B\alpha(t) \) to update \( B\alpha(t) \).
6. Choose some new genes as vaccines to update the \( M\alpha(t) \) 
   memory.
7. Calculate the affinity of each antibody in \( P\alpha(t) \), select \( N_{\text{ve}} \) 
   antibodies at ratio \( \rho \) from them to make up of immune 
   population \( P\alpha(t) \) and make up of genetic population \( P\alpha(t) \) by 
   the remainders.
8. Apply the cell clone operation to \( P\alpha(t) \) and form one part of 
   next generation, \( P\alpha_{i+1}(t) \).
9. Apply the crossover and mutation operations to \( P\alpha_{i+1}(t) \) and 
   form the other part of next generation, \( P\alpha_{i+2}(t) \).
10. Let \( P\alpha_{i+1} = \left[ P\alpha_{i+1} \right] \), calculate the fitness value of 
    each antibody in \( P\alpha_{i+1} \), and apply vaccination operation to the 
    unfeasible antibodies.
11. Cut similar antibodies from \( P\alpha_{i+1} \) by the population clustering 
    operation.
12. Select \( N_{\text{ve}} \) optimal antibodies as the next generation \( P\alpha(t+1) \) 
    if \( P\alpha_{i+1} \geq N_{\text{ve}} \); Add random antibodies into \( P\alpha_{i+1} \) to build 
    \( P\alpha(t+1) \) if \( P\alpha_{i+1} < N_{\text{ve}} \).
13. Calculate the fitness values of each antibody in \( P\alpha(t+1) \).
14. Let \( t = t + 1 \). Return to the fifth step if \( t \leq T_n \) or there is not a 
    Pareto optimal solution.
15. Let \( m = m + 1 \). Save the Pareto optimal solution as \( p\alpha^* \) and 
    return to the second step if \( m \leq M \).
16. Select an optimal solution for the set \( \{ p\alpha_1, p\alpha^*, \ldots, p\alpha_m \} \) 
    to generate the planning path.

The above algorithm is characteristic of three aspects. The first aspect
is that the population clustering operation may maintain population’s diversity. Secondly the cell clone operation may improve local search capability. At last the immune memory mechanisms may improve the quality of candidate solutions then speed the convergence velocity.

SIMULATION RESULTS

To evaluate the proposed algorithm’s performance, we selected an under actuated AUV as research object and built the AUV motion simulation system, shown as Fig. 1. The controllers of heading, depth/altitude and velocity are consistent with that of the actual AUV. The models of the AUV, its rudders and main thruster can be found in the reference (Liu, 2003). The sonar module is used to model the obstacle avoidance sonar, whose outputs represent the distances between the AUV and obstacles in forward, forward-down, down, left and right directions. To indicate the influences of external environment, some noises and errors may be added into the outputs of sonar model. The global map is built by the operator prior to mission and is unknown to the AUV. The data fusion module comprises of data processing, data fusion and local map generator, and provides a local map and reliable data to path planning and obstacle avoidance respectively.

Fig.1 The AUV motion simulation system

When AUV sails normally and no obstacles are in sight, the path planning module provides the motion controllers with the desired variables such as heading, depth/altitude and velocity. If any obstacles are detected, the obstacle avoidance module begins to respond to the changes of environment and controls these desired variables. But once obstacle avoidance fails in leading AUV to pass around the obstacles, path planning will be activated to generate a new obstacle-free path. As an example shown as Fig. 2, AUV is at the point (24,10) and its goal point is (24,50). Suppose that the path segment number is limited in M=2, we use the immune genetic algorithm (IGA), genetic algorithm (GA) and immune algorithm (IA) to plan the obstacle-free paths respectively. All the three algorithms adopt the above mentioned fitness functions. The optimal paths plotted in Fig.2, demonstrate that the length of the paths planned by the IGA is shorter than that by the other algorithms. Fig. 3 shows the convergence of fitness values of the best solutions in each generation during evolution. All the algorithms are convergent in 100th generation. But the convergence values are different. The IGA’s optimal fitness value is 0.36967 that is smaller than 1.02796 and 0.41559 which are acquired by the GA and IA.

Fig.2 Optimal paths planned by the three algorithms

A typical semi-closed obstacle scenario is illustrated in Fig.4. If there is not the path planning module and only obstacle avoidance is in work, the AUV will go inside the obstacle and turn right for avoiding the obstacle continuously. After the AUV has not detected the obstacle for a time around H point, it will switch to sail toward the goal point B and go inside the obstacle again.

Fig.3 Fitness values of the optimal antibodies in each generation during evolution by the three algorithms

When the proposed immune genetic algorithm is added in the AUV motion simulation system, the AUV trajectory is plotted in Fig.5. From it we can see that the AUV can be directed to pass around the obstacle by the online planning paths. During the whole mission in simulation, the path planning module runs for four times. Fig.6 and Fig.7 show the online generated local map and the online planning paths at the
simulation time $t=643.5$ and $t=1347$ respectively.

![Fig.5 The AUV trajectories with the path planning module](image)

Fig.5 The AUV trajectories with the path planning module

Further investigation into alternative code such as binary or real, should be implemented. A real-time obstacle avoidance demonstration on a semi-physical simulation platform is required to validate the algorithm’s real timeliness.

ACKNOWLEDGEMENTS

Our research is supported by Chinese National Natural Science foundation (60805050), the Chinese National Basic Research Program (6138102005-4 6138101004-3) and the Key Project of Innovation Knowledgement of CAS (YYYJ-0917).

REFERENCES


