

# *A Real-time Global Optimal Path Planning for mobile robot in Dynamic Environment Based on Artificial Immune Approach*

A. Eslami, S. Asadi, G.R. Soleymani, V. Azimirad

Center of Excellence in Mechatronics, School of Engineering Emerging Technologies, University of Tabriz, Tabriz, Iran

a-eslami88@ms.tabrizu.ac.ir, s-asadi88@ms.tabrizu.ac.ir, g.soleymani@gmail.com, azimirad@tabrizu.ac.ir

**Abstract**—This paper illustrates a method to finding a global optimal path in a dynamic environment of known obstacles for an Mobile Robot (MR) to following a moving target. Firstly, the environment is defined by using a practical and standard graph theory. Then, a suboptimal path is obtained by using Dijkstra Algorithm (DA) that is a standard graph searching method. The advantages of using DA are; elimination the uncertainty of heuristic algorithms and increasing the speed, precision and performance of them. Finally, Continuous Clonal Selection Algorithm (CCSA) that is combined with Negative Selection Algorithm (NSA) is used to improve the suboptimal path and derive global optimal path. To show the effectiveness of the method it is compared with some other methods in this area.

**Keywords**- Clonal Selection; Dijkstra; Path Planning; Negative Selection; MAKLINK

## I. INTRODUCTION

In mobile navigation problems, the first issue is this fact that which path is better to tracking (if exists)? Solving this problem consists of finding an optimal path from a starting point to a target point according to some conditions depends on the environmental situation. To achieve this goal, there are many methods such as potential field [1], visibility graph [2] and grid methods [3]. The potential field method is fast but it has some limitations such as oscillations in narrow passages and trap situations due to local tendency. In the visibility graph method the cost of process is very high especially in multi dimension problems and by increasing the dimension of environment, the cost will be increased in exponential form. Hence it has lower efficiency. The main issue in the grid methods, is that how to determine the size of grids. Because it is important to trade-off between the precise of representation of environment, the memory space usage and the time rate. In the recent years, the artificial intelligence is commonly used in many researches in path planning scope and some algorithms have been provided such as neural network, genetic algorithm, particle swarm, simulated annealing and Artificial Immune System (AIS) to make a safe and optimal path for an MR. AIS is inspired by biological immune system and it is relatively a new research area which can be traced back to Farmer *et al.*'s paper published in 1986 [4]. The main application domains which AIS researches have focused on are fault diagnosis,

computer security and data analysis. Recently, a few attempts have been made to extend AIS to the optimization field [5], [6]. Comparing with other soft computing algorithms, AIS is considered more practical due to the certain benefits is offered over the others. Simulated annealing is very slow; particle swarm such as ant colony optimization uses discrete data; when the number of parameters increases and when they are epistatic in nature, genetic algorithms cannot find the global optimum [7] whereas AIS is free from the above drawback. All heuristics have some problems. They don't guarantee the convergence and arriving to goal. Hence, to avoid these problems it is necessary to use a method such as graph searching methods to guide the algorithm to goal. DA [8] to compare with other graph searching methods is faster and AIS is free from problems that other heuristic methods suffer them. So, the combination of these two methods is a novel approach that will eliminate disadvantages of each other and can be used for path planning of MRs. In this paper the MAKLINK technique [9] is used as a method to introduce the available free space between obstacles in environment of MR and construct the graph to support the generation of a collision free path by DA. To construct of MR's environment we made some additional assumptions that are illustrated unabridged in [9], [14]. The developed CCSA that is combined with NSA is used simultaneously to improve speed of convergence. It ensures the finding the global optimal and obstacles free path for MR. Moreover, a method to find an optimal mutation formula is proposed to use in CCSA. In fact, this approach is an optimization of optimization. The parameters of CCSA's mutation formula are obtained by CCSA itself. It leads to obtain high speed convergence. Therefore, this high speed algorithm can be used in real-time applications easily. To prove this purporst, the algorithm is applied in a real-time global optimal path planning problem in which the environment is dynamic. Moreover, the performance of the algorithm is compared with ant colony and genetic algorithm. The rest of the paper is as follow: Section II presents review of CCSA that is combined with NSA, moreover proposes an optimal method for upgrade mutation too. In section III, we describe how to adapt the DA by the CCSA to improve the suboptimal path to the global optimal path; hence in this section two kinds of problems (static and dynamic

environment) are defined to indicate the high performance of this method. In section IV, the results are presented. Section V includes a discussion of the results and comparisons with analytic result and two kinds of ant colony simulated results. Finally, the conclusion is provided in section VI.

## II. CONTINUOUS CLONAL SELECTION

### A. The immune overview

When our body is attacked by the infectious foreign elements (called pathogens), the immune system has a main role to defend us against them. The immune system has three defensive layers which the third layer plays the major role. This layer has two types of cells, namely B and T cells (lymphocyte). Antigens are pieces of protein suspended in the blood separated from human body's cells and infectious foreign elements. Using these antigens, immune system is able to distinguish the foreign cells from inner cells. When one antigen is recognized by a B cell, this kind of B cell begins to proliferate. Many of these proliferated cells become a special kind of cells called Plasma and some of them become memory cells. The plasma cells stick to the antigens and destroy the cells which are related to antigens and then memory cells keep history of this illness [10], [11].

### B. Continuous Clonal Selection Algorithm (CCSA)

Clonal selection algorithm (CSA) is inspired by the clonal selection theory (Burnet, 1959). De Castro and Von Zuben [12] developed the first CSA based on the population search which had a set of memory suitable antibodies which are the candidates of solution. In this paper the CCSA is developed that is an improved version inspired from CSA [5]. This approach prevents losing sight of any point because of its Continuous behavior. This different CSA is described as follow:

#### 1) Initializing

In this phase the parameters take default values that are described in Table I.

TABLE I. INITIAL PARAMETERS OF CCSA

parameters	description
$\mu_{max}$	maximum rate of mutation defined in the scope [0,1]
$\mu_{min}$	minimum rate of mutation defined in the scope [0,1]
$n$	number of main population
$\beta$	reproduction factor defined in the scope [0.1]
$N_{gen}$	maximum allowable number of repeats algorithm
$N_{mem}$	The number of algorithm repeats that one member dos not change. This member should be kept in a memory cell.
$N_{rep}$	The allowable number of algorithm repeats that algorithm allowed to hang on without finding new answer.
$N_{res}$	The number of answers that algorithm have to produce.
$F_{aff}$	Maximum allowable affinity between members of population.

#### 2) Generating initial population

In this stage the Negative Selection (NS) approach [13] has been used. It has two main steps as shown in **Error! Reference source not found.1.**

a) Generate candidate of AntiBodies (AB)s randomly in [0, 1].

b) Measure the distance between candidate AB and other ABs and accept if it is more than  $D_{ff}$ .

This leads to generate a valid, normalized and sufficient distributed initial population that covers entire space of problem. Moreover faster convergence to solution will be obtained by the combination of NS and CSA.

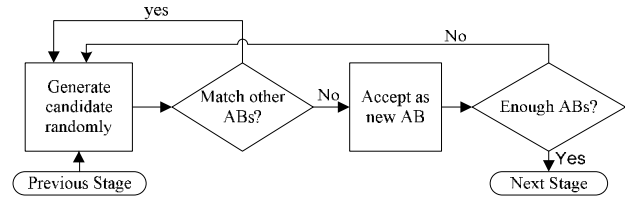


Figure 1. Generation process based on negative selection approach

Primary population of ABs is produced as follow

$$\begin{aligned}
 P &= \{x_1, x_2, \dots, x_n\}; \\
 x_1 &= (x_{11}, x_{12}, \dots, x_{1d}); \\
 x_2 &= (x_{21}, x_{22}, \dots, x_{2d}); \\
 &\vdots \\
 x_n &= (x_{n1}, x_{n2}, \dots, x_{nd});
 \end{aligned} \quad (1)$$

Where  $x_{ij} = \text{rand}(1, 1)$ ; In the above relations  $\text{rand}(x, y)$  will produce a matrix  $(x \times y)$  which entire of members are produced randomly in scope [0,1],  $d$  is the dimension (number of variations),  $X_i$  is the  $i$ th AB and  $P$  is the population.

#### 3) Proliferation

In this phase each member of population is reproduced in  $n_c$  number. Each member with its copies will be cloned. The  $n_c$  can be determined with equation  $n_c = \beta \times n$ ; and the number of entire population  $N_c$  after this phase will be  $N_c = n \times n_c$ . The ABs are produced as follow:

$$\begin{aligned}
 C &= \{C_1, C_2, \dots, C_n\}; \\
 C_1 &= \{X_1, X_1, \dots, X_1\}; \\
 C_2 &= \{X_2, X_2, \dots, X_2\}; \\
 &\vdots \\
 C_n &= \{X_n, X_n, \dots, X_n\};
 \end{aligned} \quad (2)$$

In the above relations,  $C_i$  is the clone of  $X_i$ th AB, and  $C$  is the set of entire clones.

#### 4) Mutation of proliferated population

The amount of mutation is static in various previous applications and ordinary it is produced in a specified range randomly. It is noticeable that the amount of mutation influences intensely on the speed of convergence. So, it is better to find an adaptive mutation for each repetition instead. In this stage entire ABs of proliferated population are mutated, except the main members of each group.

$$\begin{cases} C_i^* = q_i + \mu \times \text{Rand} & \text{for } i = 1, 2, \dots, n_c + 2, n_c + 3, \dots, 2n_c, 2n_c + 2, \dots \\ C_i^* = q_i & \text{for } i = 1, n_c + 1, 2n_c + 1, 3n_c + 1 \end{cases} \quad (3)$$

Where  $\mu$  represents the mutation rate,  $\text{Rand}$  is a random matrix and has the same dimension of population. The  $C_i^*$  is a antibody which is mutated from  $q_i^{\text{th}}$  AB. Notice that in these

relations, members of the proliferated population must be confined in scope  $[0,1]$  like their parents. This can be done by several methods like follow:

$$\begin{cases} \text{If } X_{ij}^* > 1 & \text{then } X_{ij}^* = 1; \\ \text{If } X_{ij}^* < 0 & \text{then } X_{ij}^* = 0; \end{cases} \quad (4)$$

Another method which can be used, is mapping all members from the scope  $[-1,1]$  to the scope  $[0,1]$ .

There are several kinds of mutations. The mutation rate that is used in this study can be determined by (5) in which  $gen$  is the number of iterations and  $N_{gen}$  is the maximum number of iterations. The  $N_{gen}$  is initialized in first step. The  $\mu_{min}$  and  $\mu_{max}$  are the minimum and the maximum mutation rates that are initialized too.

$$\mu = \mu_{min} + (\mu_{max} - \mu_{min}) \times \exp(-a \times gen / N_{gen}) \quad a > 0 \quad (5)$$

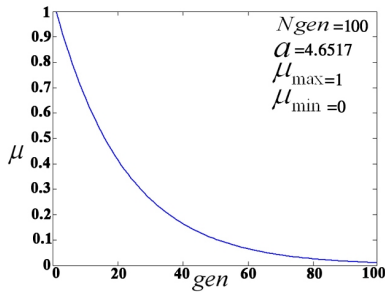


Figure 2. Amount of mutation in each iteration

**Error! Reference source not found.** illustrates that in the (5) and at the beginning of iterations the amount of mutation is approximately  $\mu_{max}$  and at the end of iterations, it is  $\mu_{min}$ . It is important to adjust the  $a$  in (5) to achieve these goals. To obtain a suitable and optimal amount for  $a$ , CCSA is used to minimize the number of iterations without losing the accuracy. Note that decreasing number of iterations will lead to increase the variation of results in various simulation trials. So, we have to compromise between speed of algorithm and precision of results. But, by using the amounts of  $a$  (Table II), which are obtained by CCSA, it is guaranteed that the results will have the best accuracy versus the number of its iterations in comparison with other amounts of  $a$ . This interpolation approach is an optimization of optimization in fact.

TABLE II. AMOUNT OF OPTIMAL  $a$  VERSUS NUMBER OF ITERATIONS

$N_{gen}$	50	100	150	200	300	400
$a$	3.9919	4.6517	5.0443	5.3249	5.7229	6.0065

#### 5) Evaluation of proliferated population

In this stage of algorithm we have to present entire proliferated ABs to the cost function and determine the vector  $f$  that contains the affinity of the ABs related to the cost function. Then eliminate a subset of antibodies (for instant  $K$  percent of them) with the least affinities in clone, and replace the eliminated ABs by new generated ones.

#### 6) Improving main population

In this phase the best member of each clone with the best affinity related to cost function must be selected and then the main population members are replaced with them.

#### 7) Detecting and registering answers in memory

For detecting the answers, a negative selection approach is used. In this method if an antibody is not changed or is not improved in  $N_{mem}$  time repetition, it will be registered in a memory cell as one of the problem's answers. Then the registered AB must be eliminated and replaced with one new generated antibody.

#### 8) Eliminating similar antibodies

In this step the affinity of memory cells is determined with entire population and if their affinity is more than predetermined  $F_{aff}$ , the AB must be eliminated from population and then a new valid antibody will be generated and replaced instead. The affinity between two AB is determined as  $F_{aff}(i,j) = 1/D_{ij}$ . In this relation,  $D_{ij}$  is Euclidean distance between  $AB_i$  and  $AB_j$ .

#### 9) Evaluation of concluding conditions

If conditions are not satisfied, go to the stage 4 and continue. Notice that satisfying each one of the following conditions will lead to conclude the program:

- The algorithm has been iterated  $N_{gen}$  times.
- Not finding new answers in predetermined  $N_{rep}$  times iteration.
- Finding certain number of answers ( $N_{res}$ ).

### III. IMPROVING SUBOPTIMAL TO GLOBAL OPTIMAL PATH

The path which is obtained by implementing DA is valid and a global suboptimal path. Before applying DA, it is needed to present a feasible environment by any suitable graph theories. The suboptimal path must be improved to achieve the global optimal path. For this purpose CCSA is used to adjust nodes of path into the span that already has been identified by graph theory used in environment presentation. To illustrate how to utilize this method in path planning problems, two kinds of problems will be presented in coming sections.

#### A. Path planning of an MR

##### 1) Define problem

Consider that there is an environment with obstacles as shown in **Error! Reference source not found.**3 which has been used in [14], [15]. The MR must go from start point (S) to the target point (T), with considering minimum measuring distance (cost function) and obstacle collision avoidance for MR. In order to avoid a moving path to be too close to the obstacles, the boundaries of every obstacle are expanded.

##### 2) Solving the problem

As mentioned above, it is needed to present feasible environment by any graph theories. In this case study MAKLINK graph theory is used. To construct of MR's environment we made some assumptions that are illustrated unabridged in [9], [14]. First, for constructing MAKLINK graph, free MAKLINK lines must be drawn. In **Error! Reference source not found.** 3, dash lines are free MAKLINK lines and solid lines are MAKLINK graph which are produced by middle points of free MAKLINK lines [15].

After creating the graph, the DA on this graph is used and a suboptimal path is obtained. In **Error! Reference source not found.4**, suboptimal path that is obtained from the DA is shown.

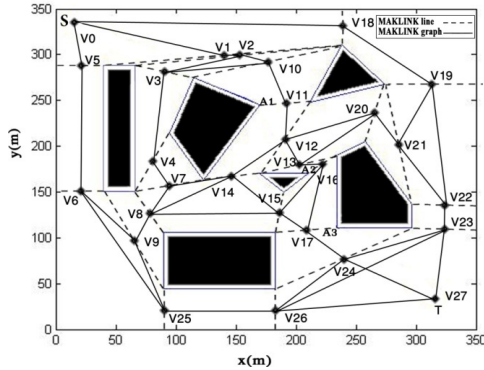


Figure 3. An environment, MAKLINK line and MAKLINK graph

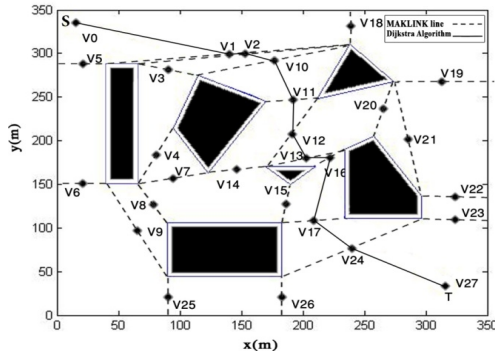


Figure 4. Dijkstra suboptimal path

The suboptimal path involves some nodes of graph which are located on free MAKLINK lines. In order to achieve the shortest path, these nodes are displaced within their own free MAKLINK lines hence CCSA is used for this purpose.

To locate any path node belongs to that free MAKLINK line, the following formula is used:

$$P_i = K_{i1} + (K_{i2} - K_{i1}) \times h_i \quad (6)$$

$$h_i \in [0,1] \quad i = 1, 2, \dots, d$$

Where  $P_i$  denotes path nodes,  $K_{i1}$  and  $K_{i2}$  are the starting and ending points of the free MAKLINK lines which  $i$ th node is located on it and  $d$  is the number of free MAKLINK lines that intersect the suboptimal path. Note that  $P_0$  and  $P_{d+1}$  are  $S$  and  $T$  points respectively.

In CCSA, the  $h_i$  is expressed as an antibody and (7) is expressed as an antigen.

$$L = \sum_{i=0}^d \text{length}\{P_i(h_i), P_{i+1}(h_{i+1})\} \quad (7)$$

As another benchmark, consider the environment that is shown in **Error! Reference source not found. 5** [16]. Yang *et al.* have improved the ant colony algorithm of Guan-Zheng *et al.* [15].

The simulation results will be presented in section IV.

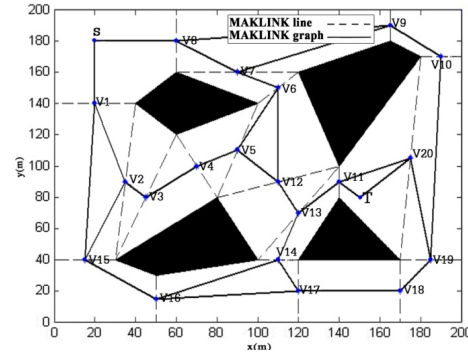


Figure 5. An environment, MAKLINK line and MAKLINK graph

#### B. Following a mobile target in a dynamic environment

Consider the environment as shown in **Error! Reference source not found.3**. The MR must follow the mobile target point (T) from current point (S), with considering minimum measuring distance (cost function) and obstacle collision avoidance for MR. Stages of algorithm are the same as previous problem but entire of obstacles in environment as well as the MR and the target are moving dynamically. The simulation results are shown in section IV.

#### IV. SIMULATION RESULT

In these studying cases, we develop our algorithms in MATLAB and simulation results are executed on a personal computer with 2GHz dual core CPU and 2GB RAM. The parameters are set as  $\mu_{max}=0.05$ ,  $\mu_{min}=0.0001$ ,  $N_{gen}=300$ ,  $n=4$ ,  $\beta=1.7$ ,  $N_{mem}=10$ ,  $N_{res}=1$ ,  $N_{rep}=50$  and  $F_{ait}=5$ . The simulation results are shown in **Error! Reference source not found.6** and **Error! Reference source not found.7**.

As shown in **Error! Reference source not found.6**, a dash line denotes a suboptimal MR path which is derived from DA with the length of 507.692 meters and solid line denotes a global optimal MR path which is derived after improving suboptimal path by CCSA with the length of 439.0128 meters. The optimal values for parameters of  $h$  (antibodies) for this example are  $h(1, \dots, d(=9)) = [0.3370, 0.1428, 0.0135, 0.0000, 0.4774, 0.5352, 0.0009, 0.9999, 0.3379]$ . **Error! Reference source not found.7** shows the optimal and the suboptimal paths that are obtained by CCSA and DA respectively. Also, **Error! Reference source not found.8** shows the paths are obtained by ant colony and improved ant colony algorithms that are developed by Guan-Zheng *et al.* [15] and Yang *et al.* [16] respectively. Lengths of paths are shown in Table III. The optimal values for parameters of  $h$  (antibodies) for second case are  $h(1, \dots, d(=5)) = [0.0065, 0.008, 0.000, 0.9149, 0.9269]$ .

TABLE III. LENGTHS OF PATHS

METHOD	AIS	IACS <sup>a</sup>	ACS <sup>b</sup>	Dijkstra	analytic
LENGTH (METERS)	167.5473	168.217	169.388	223.203	167.5452

a.[15] b.[13]

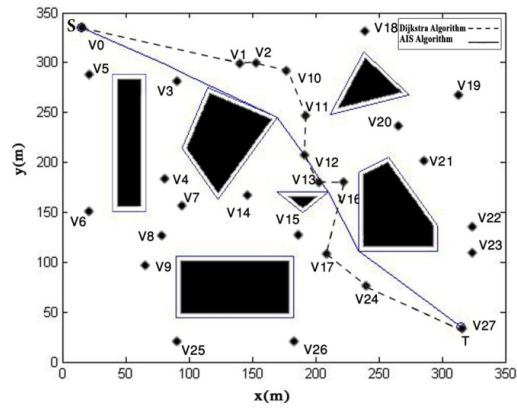


Figure 6. Optimal path that derived from AIS and Dijkstra

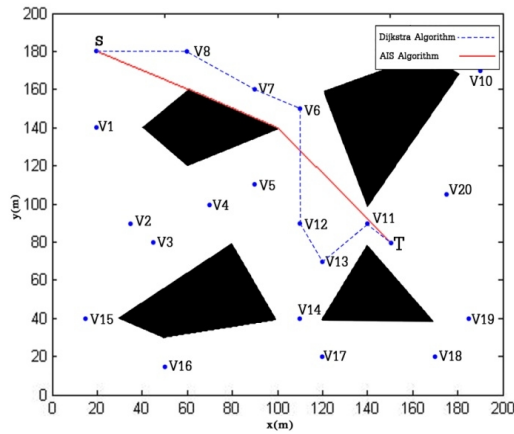


Figure 7. Optimal path that derived from AIS and Dijkstra

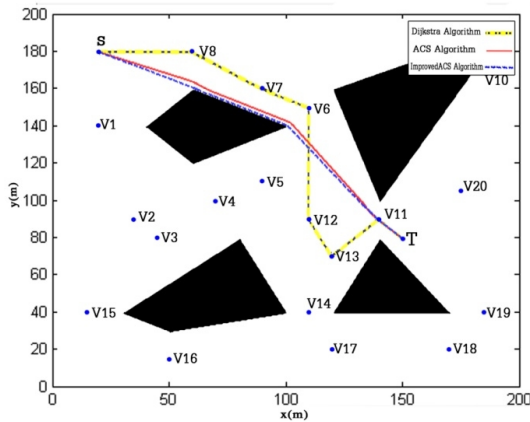


Figure 8. Optimal path that is derived from Dijkstra, ACS algorithm and Improved ACS algorithm [15], [16]

As the third application, for prove that the algorithm can be used in dynamic and real-time situations, the environment of MR (**Error! Reference source not found.3**) is supposed dynamic and the MR (S) follows the mobile target point (T).

Entire of obstacles in environment are moving dynamically. The simulation results are shown in **Error! Reference source not found.9**. The paths are obtained with considering minimum measuring distance (cost function) and obstacle collision avoidance for MR dynamically.

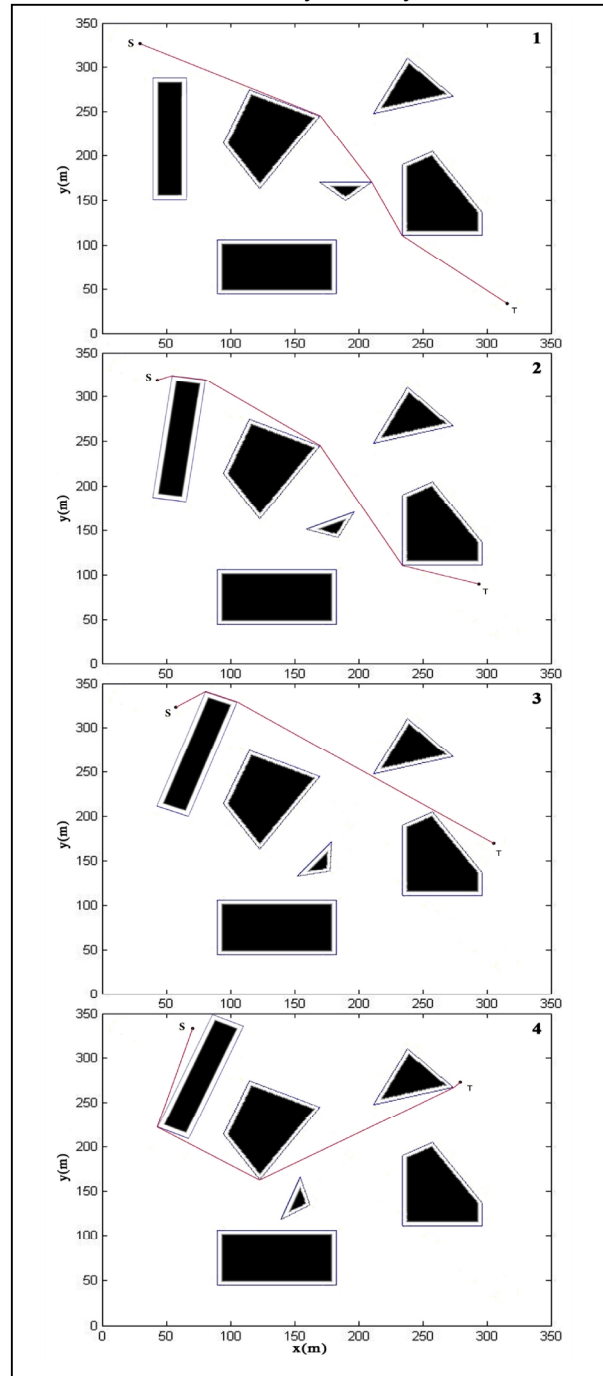


Figure 9. Optimal path that derived from CCSA in dynamic environment

## V. COMPARISON WITH OTHER ALGORITHMS

As a comparing object, ant colony algorithm and GA, which had been applied by TAN Guan-Zheng *et al.* [15] and improved ant colony algorithm by Along Yang *et al.* [16] as well as the analytical solution are used for path planning example. In analytical solution, it is clear that the shortest path from *S* to *T* passes the points *A1*, *A2* and *A3* as is shown in **Error! Reference source not found.**3. So the analytic length of this path is 439.0011 meters. In ant colony algorithm and GA as were presented in [15], the minimum path length was obtained 440.233 meters for both. It is obvious that AIS has better result than ACS and GA moreover AIS has more accurate results with considering analytical solution. For another comparing object, 100 simulation trials are performed for AIS and the variations of results are observed and compared with ant colony algorithm and GA. As shown in Table IV, the AIS is more precise than other two algorithms. As shown in Table IV, the result of AIS is even better than improved ant colony algorithm [16].

TABLE IV. VARIATION OF RESULTS COMPARISON (METERS)

METHOD	min	max	Standard deviation
AIS	439.0128	439.0441	0.0062
ACS <sup>a</sup>	440.233	447.020	1.5644
GA <sup>a</sup>	440.233	452.743	1.5227
Dijkstra	507.692	507.692	0

<sup>a</sup>[13]

The amount of time that it takes to run the program is 0.526 second and this is appreciable to use in real time applications. It is interesting that if we choose  $N_{gen}=100$  and  $a=4.6517$ , the time will be decreased approximately to 0.15 second without significant variation in accuracy.

## VI. CONCLUSION

This paper proposes a new method for global optimal path planning based on combination of two AIS based algorithms (clonal selection and negative selection) and a DA. Combination of the adaptive DA with CCSA improves the speed of convergence of CCSA and ensures the finding a valid, global and the optimal solution. The speed of algorithm is appreciable to use in real time applications. Also, in this paper a method to find the optimal amount of the mutation is presented and this innovation improved the speed of CCSA significantly. In compression with other methods such as ACS and GA to solve real time problems this algorithm is faster and more precise.

To show the advantages of CCSA, this method is applied in two kinds of cases. The first kind is global path planning just with obstacle collision avoidance and as shown above the result of AIS (CCSA) in comparison with ACS, IACS and GA is more accurate and precise. In the second case, a global path planning is found with considering minimum measuring distance (cost function) and obstacle collision avoidance for MR. In this case entire of obstacles in environment as well as

the MR and the target are moving dynamically and the simulation shows that this proposed novel method can be applied in real world applications.

In future, we will further improve this proposed novel method by using new adaptive methods to mutation and combination with other intelligent algorithms in this area. Moreover, in our next study the environment of problem will be 3D, for instant in UAVs or submarines applications.

## REFERENCES

- [1] Pei-Yan Zhang, Tian-Sheng Lü, and Li-Bo Song. "Soccer robot path planning based on the artificial potential field approach with simulated annealing." (*Robotica* (Cambridge University Press) 22 (2004): 563–566).
- [2] Li L, Ye T, and Tan T. "Present state and future development of mobile robot technology research." *Robot* 24, no. 5 (2002): 475-480.
- [3] Boschion V, and Pruski A. "Grid modeling of robot cells:a memory efficient approach." (*Intelligent and Robotic Systems* 8, no. 2 (1993): 201-223).
- [4] J. Doynne Farmer, Norman H Packard, and Alan S Perelson. "The immune system, adaptation, and machine learning." (*Physica D: Nonlinear Phenomena* (Elsevier) 22, no. 1-3 (October-November 1986): 187-204).
- [5] L.N de Castro, and F.J Von Zuben. "Learning and optimization using the clonal selection principle." (*IEEE Trans. on Evolutionary Computing* 6, no. 3 (jan 2002): 239-251).
- [6] Johnny Kelsey, and Jon Timmis. "Immune Inspired Somatic Contiguous Hypermutation for Function Optimization." (*Lecture Notes in Computer Science* (Springer) 2723 (2003): 6-26).
- [7] S.A Panimadai Ramaswamy, G.K Venayagamoorthy, and S.N Balakrishnan. "Optimal control of class of nonlinear plants using artificial immune systems: application of the clonal selection algorithm." (*IEEE International Symposium on Intelligent Control*. Singapore, 1-3 October 2007. 249-254).
- [8] Richard Neapolitan, and Kumarss Naimipour. *Foundations of Algorithms Using C++ Pseudocode*. (Third Edition. Sudbury, Massachusetts: Jones and Bartlett Publishers, 2004).
- [9] Maki K, HABIB, and ASAMA Hajime. "Efficient method to generate collision free paths for autonomous mobile robot based on new free space structuring approach." *International Workshop on Intelligent*. Osaka: IEEE, 1991. 563-567.
- [10] Hongwei Mo. *Handbook of Research on Artificial Immune Systems and Natural Computing:Applying Complex Adaptive Technologies*. (Medical Information Science Reference (an imprint of IGI Global), 2009).
- [11] Dipankar Dasgupta, and Luis Fernando Niño. *Immunological computation : theory and applications*. (CRC Press, 2009).
- [12] L.N de Castro, and F.J Von Zuben. "An Evolutionary Immune Network for Data Clustering." *IEEE Brazilian Symposium on Artificial Neural Networks*. Rio de Janeiro: IEEE, 2000. 84-89.
- [13] S. Forrest, Perelson S.A., L. Allen, and R. Cherukun. "Self-Nonself Discrimination in a Computer." (*Proceeding of 1994 IEEE Symposium on Research in Security and Privacy*. Los Alamos, CA: IEEE Computer Society Press).
- [14] Yanling, Hao, Shen Zhifeng, and Yuxin Zhao. "Path Planning for Aircraft Based on MAKLINK Graph Theory and Multi Colony Ant Algorithm." *International Joint Conference on Computational Sciences and Optimization*. IEEE, 2009. 232-235.
- [15] TAN Guan-Zheng, HE Huan, and SLOMAN Aaron. "Ant colony system algorithm for real-time globally optimal path planning of mobile robot." (*Acta Automatica Sinica* (Sciencedirect) 33, no. 3 (March 2007): 279-285).
- [16] Along Yang, Lixin gGao, and Yanping Luo. "An Improved Ant Colony System Algorithm for Optimal Path Planning Problem of Mobile Robots." (*Second International Conference on Computer Modeling and Simulation*. IEEE-computer society, 2010. 526-530).