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 8 (3): 360-365, 2009
ISSN 1812-5638

Artificial Immune-Chaos Hybrid Algorithm for Geometric Constraint Solving

[1]Xue-Yao Gao, [1]Li-Quan Sun and [2]Da-Song Sun
[1]Research Institute of Computer Applied Techniques,
[2]Computer Center,
Harbin University of Science and Technology, Harbin, China

Abstract: Geometric constraint solving can be transformed into optimization problem which is non-linear and multi-variable. Geometric constraint solving based on artificial immune algorithm and improved chaos search strategy is proposed in this study. The local optimal solutions obtained by artificial immune algorithm are used as the heuristic information and the global best solution is searched by improved chaos search strategy in the neighborhood of local optimal solutions. In order to enhance precision and searching speed, chaos search area is controlled in the neighborhood of local optimal solutions by reducing search area of variables. This algorithm differs from current optimization methods in that it gets the global best solution by excluding bad solutions. Experiment results show that the proposed method is better than artificial immune algorithm and can deal with geometric constraint solving efficiently.

Key words: Antibody, antigen, chaos search, optimization, geometric constraint

INTRODUCTION

One constraint describes a relation that should be satisfied between geometric elements in CAD and CAID. Once a user defines a series of relations, the system will satisfy the constraints by selecting proper state after parameters are modified. The process is named geometric constraint solving (Gao and Jiang, 2004).

Chen *et al*. (2000) applied bipartite graph into optimization process, in which self-adaptive adjustment is realized and constraint solving quality is improved. Kramer (1992) used freedom degree analysis to process geometric constraint problems which are linkage produced. Qiao *et al*. (2002) presented an algorithm based on graph, sparse matrix and freedom degree analysis, which reduced scale of constraint solving by decomposing geometric constraint system mostly. A method based on Gröbner basis was given to solve constraint problem (Kondo, 1992). Joan-Arinyo *et al*. (2003) applied genetic algorithm into constraint solving, but this method can only solve small scale problems. Genetic simulated annealing algorithm was applied to constraint solving (Liu *et al*., 2003), but the search is processed blindly. Cao *et al*. (2007) introduced Newton-Genetic algorithm into constraint solving.

Artificial Immune Algorithm (AIA) is population-based heuristic optimization algorithm, which is diverse, inherently distributed, capable of automatically recognizing antigens and maintaining memory cells (Dasgupta *et al*., 2003). Artificial immune algorithm can rapidly confine the solutions in a small area, but it can't precisely converge to the optimal solution. Chaos search algorithm is non-sensitive to initial value and searches rapidly (Li and Jiang, 1997). It can get rid of local extreme and converge to the optimal solution (Tavazoei and Haeri, 2007).

In this study, an improved chaos search strategy and an Artificial Immune-Chaos Hybrid Algorithm (AICHA) are presented and AICHA utilizes advantages of artificial immune algorithm and improved chaos search strategy. Then we apply AICHA to geometric constraint solving. Experiment results show that the proposed method can work well in geometric constraint solving.

ARTIFICIAL IMMUNE ALGORITHM

The artificial immune algorithm is proposed on the basis of definition and theory of biology immune system (Timmis *et al*., 2008). The biology immune system can defend foreign antigens by antibodies which are auto-generated. After antibodies combine with antigens, antigens will be destroyed by a series of response. Antibodies can activate and restrain each other and this response is on the basis of antibody concentration. The

Corresponding Author: Xue-yao Gao, Research Institute of Computer Applied Techniques,
Harbin University of Science and Technology, Harbin, China

higher the concentration, the more restrained the antibodies and vice versa. In artificial immune algorithm, the antigen is the object function to be solved and antibody is the solution. Affinity between antibody and antigen reflects the total combination intensity between antibody and antigen, which indicates the matching degree between the available solution and objective function. Similarity between antibodies ensures the diversity of available solutions. Anticipant propagate rate is calculated to restrain redundant solutions. The selected available solutions are stored in the memory cells, which restrain more similar solutions and ensure converging to the global best solution rapidly.

For an antibody colony that contains N antibodies, $x_i = (x_{i1}, x_{i2},..., x_{iD})$, i = 1, 2, ..., N, several related definitions are given as follows:

Similarity between antibodies: The similarity ax_{ij} between antibody $x_i = (x_{i1}, x_{i2},..., x_{iD})$ and $x_j = (x_{j1}, x_{j2},..., x_{jD})$ is:

$$ax_{ij} = \frac{1}{1+H_{ij}} \tag{1}$$

where, H_{ij} is combination intensity between antibody x_i and x_j, which is Euclidean distance in this study.

Affinity between antibody and antigen: Affinity between antibody and antigen represents degree of antibody recognizing antigen. The affinity ax_i between antibody $x_i = (x_{i1}, x_{i2},..., x_{iD})$ and antigen is:

$$ax_i = \frac{1}{1+opt_i} \tag{2}$$

where, opt_i is fitness, namely optimal value of object function, i = 1, 2,, ..., N, $ax_i \in (0, 1)$, which relates affinity ax_i with optimal value of object function. The larger is the ax_i, the more antibody and antigen are suited.

Concentration of antibody: The concentration c_i of antibody is calculated as:

$$c_i = \frac{1}{N}\sum_{j=1}^{N} ac_{ij} \tag{3}$$

Where:

$$ac_{ij} = \begin{cases} 1 & ax_{ij} \geq Tacl \\ 0 & \text{otherwise} \end{cases}$$

where, Tacl is a threshold which is set beforehand.

Anticipant propagate rate of antibody: Anticipant propagate rate e_i relates with affinity ax_i and concentration c_i of antibody:

$$e_i = \frac{ax_i}{c_i} \tag{4}$$

Selection probability of antibody: Selection probability p_{si} represents probability to select and clone of antibody x_i, which is calculated as:

$$p_{si} = \frac{e_i}{\sum_{i=1}^{N} e_i} \tag{5}$$

From Eq. 5, selection probability relates not only to fitness but also to concentration of antibody.

CHAOS SEARCH STRATEGY

Basic chaos search strategy: Here logic self mapping function is used to produce chaos variables:

$$z_{n+1} = 1 - 2 \times z_n^2,\ n = 0, 1, 2, \ldots,\ -1 < z_n < 1 \tag{6}$$

If the target function $f(x_i)$ is continuous, object problem to be optimized is:

$$\min f(x_i),\ x_i \in [a_i, b_i],\ i = 1, 2, ..., D \tag{7}$$

The basic process of chaos search strategy can be described as follows:

Step 1: Let k = 0, create D different chaos variables z_i (k) randomly and z_i (k) $\neq$ 0, i = 1, 2, .., D. k is the iterative symbol of chaos parameters. Let z_i^* denote the current best chaos variable, f* is the current best solution and initialized as a biggish number.

Step 2: Map chaos variable z_i (k) to optimization variable area and is signed as x_i (k):

$$x_i(k) = \frac{(b_i - a_i)}{2} z_i(k) + \frac{(b_i + a_i)}{2} \tag{8}$$

Step 3: Calculate $f(x_i(k))$ and if $f(x_i(k)) < f^*$, $f^* = f(x_i(k))$, $z_i^* = z_i$ (k)

Step 4: Let k = k+1, $z_i (k) = 1\text{-}2\,(z_i (k))^2$

Repeat from step 2 to step 4 until f* keeps unchanged in certain steps or iterative time reaches the given one. z_i^* is the best chaos variable and f* is the best solution.

Improved chaos search strategy: In order to avoid searching blindly and enhance search speed, we reduce search area of variable around the current optimal solution, according to Eq. 9 and 10:

$$a_i' = z_i^* - C(b_i - a_i) \tag{9}$$

$$b_i' = z_i^* + C(b_i - a_i) \tag{10}$$

where, C∈(0, 0.5) is adjustment coefficient. The rest of particle search areas remain invariably. Therefore, from step 4, process of chaos search is improved as:

Step 4: Let k = k+1 and reduce search area of current best solution according to Eq. 9 and 10. If $a_i' < a_i$, a_i is set to a_i' and if $b_i' > b_i$, b_i is set to b_i'

Step 5: Revert optimization variable x_i (k) to chaos one:

$$z_i(k) = \frac{2}{b_i - a_i}(x_i(k) - \frac{b_i + a_i}{2})$$

and create chaos series according to Eq. 6.

Step 6: Map z_i (k) to optimization variable area and calculate $f(x_i(k))$. If $f^* < f(x_i(k))$, turn to step 4, else revert search area of z_i^*: $a_i' = z_i^* - a_i$, $b_i' = z_i^* + b_i$ and $f^* = f(x_i(k))$, $z_i^* = z_i$ (k).

Repeat from step 4 to 6 until f* keeps unchanged in certain steps or iterative time reaches the given one. z_i^* is the best chaos variable and f* is the best solution.

ARTIFICIAL IMMUNE-CHAOS HYBRID ALGORITHM (AICHA)

Proliferation of immunocyte is one of chaos phenomenon, we propose to combine improved chaos search strategy with artificial immune algorithm. The initial antibodies are generated by chaos series and are distributed symmetrically in solution space, which can avoid redundancy of stochastic series. The approximative global optimal solutions are searched by artificial immune algorithm. Then, the neighborhood of the approximative solutions is searched by improved chaos search strategy, to obtain the global precise solution.

On the basis of above definitions and analysis, steps of AICHA are described as follows:

Step 1: Initialize population number N, memory cell number M, immune selection threshold T, cross probability p_c, mutation probability p_m and terminate condition S.

Step 2: Initialize a population of antibodies: Initialize z_n in Eq. 6 by D different variables in area (-1, 1) excluding 0, which is denoted as $z_n = (z_{n1}, z_{n2}, .., z_{nD})$. z_n is mapped to optimization variable x_n according to Eq. 8 and denoted as $x_n = (x_{n1}, x_{n2}, .., x_{nD})$, which represents an antibody. Initialize antibody colony which contains N+M antibodies according to this method.

Step 3: Calculate similarity ax_{ij} between antibodies according to Eq. 1 and calculate affinity ax_i between antibodies and antigen according to Eq. 2.

Step 4: Calculate anticipant propagate rate: Calculate concentration c_i of antibody according to Eq. 3 and calculate anticipant propagate rate e_i according to Eq. 4.

Step 5: Generate parent colony: Rank the initial colony in descending order according to e_i and select the first N antibodies to generate parent colony. Choose and store the first M antibodies as memory cells.

Step 6: Choose the preferable antibody individuals for chaos search: Choose the first 20% memory cells $x_i = (x_{i1}, x_{i2}, .., x_{iD})$ for chaos search. These antibody cells are firstly reverted to chaos variables:

$$z_i = \frac{2}{b_i - a_i}(x_i - \frac{b_i + a_i}{2})$$

and are iterated once according to Eq. 6. Then, the optimal solution is searched by improved chaos search strategy which is proposed above.

Step 7: If the terminate condition is satisfied, stop the algorithm, otherwise turn to step 8.

Step 8: Generate a new colony and turn to step 3. Parent colony is firstly selected, crossed and mutated to generate N solutions. Then, a new colony is generated, which contains the N solutions and M individuals of memory cells. Here, selection probability p_{si} of each antibody is calculated according to Eq. 5 when parent colony is selected. Antibody colony is selected and copied by proportion select method according to selection probability p_{si}. When antibody colony is crossed and mutated, cross probability p_c∈(0, 1) and mutation probability p_m∈(0, 1) are used.

GEOMETRIC CONSTRAINT SOLVING BASED ON AICHA

The geometric constraint can be formalized as(E, C). $E = (e_1, e_2, .., e_n)$ expresses geometric elements, such as point, line, circle and etc. $C = (c_1, c_2, .., c_n)$, c_i is a set of constraints between geometric elements. Usually one constraint is represented by an algebraic equation (Ge *et al*., 2000), which can be expressed as follows:

$$\begin{cases} f_1(x_0, x_1, x_2, \ldots, x_n) = 0 \\ \ldots \\ f_m(x_0, x_1, x_2, \ldots, x_n) = 0 \end{cases} \tag{11}$$

where, $X = (x_0, x_1, .., e_n)^T$, x_i are parameters of geometric elements. For example, planar point can be expressed as (x_1, x_2). The process of constraint solving is to get a solution X to satisfy Eq. 11.

$$F(X) = \sum_{1}^{m} |f_i| \tag{12}$$

Apparently if X can satisfy F(X) = 0, then X can satisfy Eq. 11. So, the constraint problem can be transformed into a multi-variable optimization function and we only need to solve X when F(X) is minimal.

In this study, geometric constraint solving problem is transformed into optimization problem according to Eq. 12 firstly. Then, solutions of the transformed non-linear and multi-variable function are searched by AICHA proposed above. The inputs of the algorithm are constraint equations in the form of Eq. 11 and outputs are solutions $X = (x_0, x_1, .., e_n)^T$ of the transformed function in the form of Eq. 12. For planar geometric constraint, the outputs are coordinate of points.

RESULTS AND DISCUSSION

An original design is shown in Fig. 1, in which parameters and constraint conditions are labeled. When some dimensions and angles are modified, we solve this problem by AIA and AICHA respectively and output coordinates of P_1, P_2, P_3, O_2 and experiment results are shown in Table 1. Experiment results solved by AICHA are shown in Fig. 2.

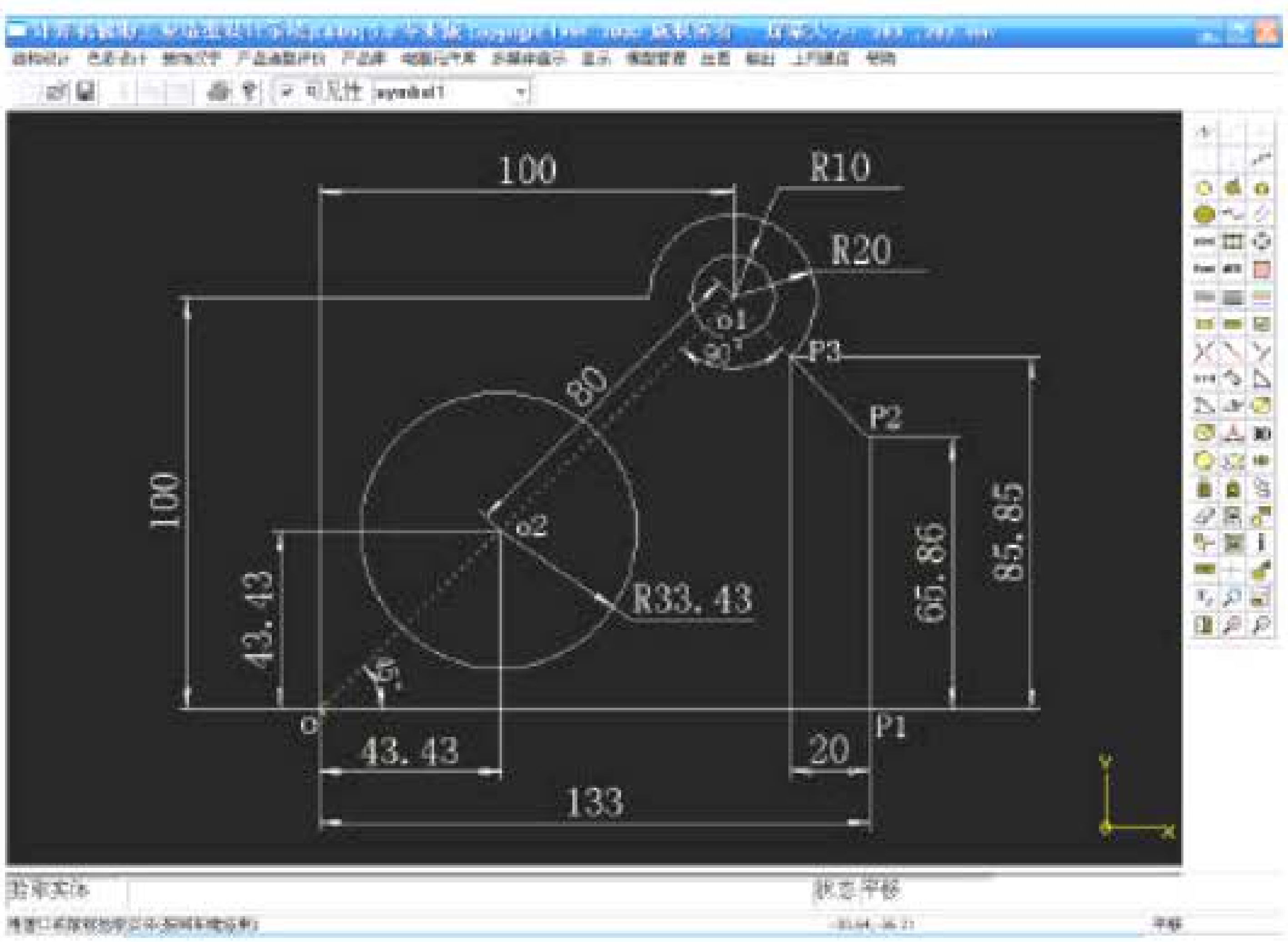

Fig. 1: Original design

Table 1: Compared experiments of AIA and AICHA

	Actual value	Algorithm	Best optimal value	Average optimal value
P_1	(151.733118, 0)	AIA	(151.716652, 0)	(151.712383, 0)
		AICHA	(151.733054, 0)	(151.733039, 0)
P_2	(151.733118, 36.194718)	AIA	(151.716652, 36.218853)	(151.712383, 36.235187)
		AICHA	(151.733039, 36.201258)	(151.733039, 36.201258)
P_3	(131.733118, 63.803275)	AIA	(131.716652, 63.831592)	(131.712383, 63.845766)
		AICHA	(131.733039, 63.804329)	(131.733039, 63.804329)
O_2	(47.113446, 27.200961)	AIA	(47.076763, 27.230116)	(46.882157, 27.247431)
		AICHA	(47.109664, 27.201512)	(47.109664, 27.201512)

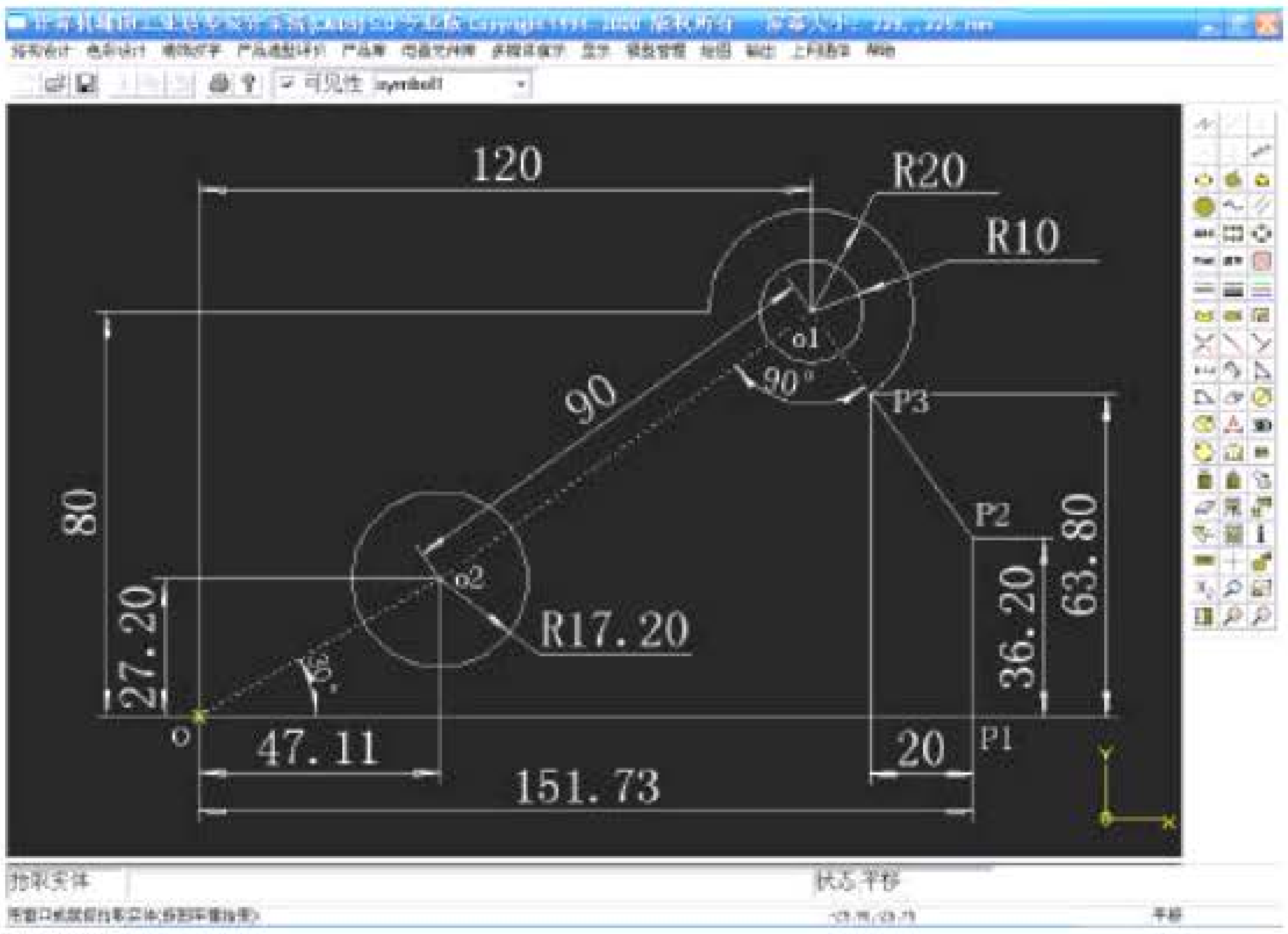

Fig. 2: Modified result by AICHA

The same initial conditions were used for both algorithms. The population number n is set to 100, memory cell number m is set to 20, immune selection threshold T is set to 0.85, cross probability p_c is set to 0.85 and mutation probability p_m is set to 0.05. For AIA and AICHA, the terminate condition is both setting the iterate generations to 60. When the iteration generations reaches 60, the antibody that most fits the antigen is considered as the solution. For each algorithm, 30 trials are carried out respectively, the best optimal value and the average optimal value are shown in Table 1.

From Table 1, the best optimal value and the average optimal value of AICHA are both better than that of AIA. AICHA can solve geometric constraint more efficiently.

CONCLUSION

In this study, an artificial immune-chaos hybrid algorithm for geometric constraint solving is proposed. The proposed method transforms traditional geometric constraint solving into optimization problem to search the solution. This new algorithm integrates advantages of the artificial immune algorithm and improved chaos search strategy. The antibody colony is initialized by chaos series. The global optimal solutions are parallel searched on the basis of approximative solutions which are searched by artificial immune algorithm, these preferable solutions are taken as the parent colony to be selected, cloned, crossed and mutated and to generate a new colony which inherits excellent characteristic of parent colony. Experiment results show that the new method is better than artificial immune algorithm and can solve geometric constraint efficiently.

ACKNOWLEDGMENT

This study was supported by National Nature Science Foundation of China under Grant No. 60173055.

REFERENCES

Cao, C.H., B. Zhang and W.H. Li, 2007. Geometric constraint solving based on Newton-genetic hybrid algorithm. J. Syst. Simulat., 19: 3650-3652.

Chen, L.P., B.X. Wang, X.B. Peng and J. Zhou, 2000. An optimal method of bipartite graph matching for under-constrained geometry solving. Chinese J. Comput., 23: 523-530.

Dasgupta, D., Z. Ji and F. Gonzalez, 2003. Artificial immune system (AIS) research in the last five years. Proceedings of the International Conference on Evolutionary Computation, December 8-12, Canberra, Australia, pp: 123-130.

Gao, X.S. and K. Jiang, 2004. Survey on geometric constraint solving. J. Comput. Aided Design Comput. Graph., 16: 385-396.

Ge, J.X., S.C. Chou and X.S. Gao, 1999. Geometric constraint satisfaction using optimization methods. Comput. Aided Design, 31: 867-879.

Joan-Arinyo, R., M.V. Luzon and A. Soto, 2003. Genetic algorithms for root multiselection in constructive geometric constraint solving. Comput. Graph., 27: 51-60.

Kondo, K., 1992. Algebraic method for manipulation of dimensional relationships in geometric models. Comput. Aided Design, 24: 141-147.

Kramer, G.A., 1992. A geometric constraint engine. Artif. Intell., 58: 327-360.

Li, B. and W.S. Jiang, 1997. Chaos optimization method and its applications. Control Theor. Appl., 14: 613-615.

Liu, S.L., M. Tang and J.X. Dong, 2003. Two spatial constraint solving algorithm. J. Comput. Aided Design Comput. Graph., 15: 1021-1029.

Qiao, Y., B.X. Wang and W. Xiang, 2002. Solving 3D geometric constraints based on freedom analysis. J. Comput. Aided Design Comput. Graph., 14: 557-561.

Tavazoei, M.S. and M. Haeri, 2007. An optimization algorithm based on chaotic behavior and fractal nature. J. Comput. Applied Math, 206: 1070-1081.

Timmis, J., A. Hone, T. Stibor and E. Clark, 2008. Theoretical advances in artificial immune systems. Theor. Comput. Sci., 403: 11-32.