A Hybrid Immune PSO for Constrained Optimization Problems

Aijia Ouyang
College of Mathematics and Information Engineering, Jiaxing University, Jiaxing 314001,China
[1] ouyangaijia@163.com

***Abstract*—Precise Algorithms combining evolutionary algorithms and constraint-handling techniques have shown to be effective to solve constrained optimization problems during the past decade. This paper presents a hybrid immune PSO (HIA-PSO) algorithm with a feasibility-based rule which is employed in this paper to handle constraints in solving global nonlinear constrained optimization problems,and Nelder-Mead simplex search method is used to improve the performance of local search in the algorithm. Simulation results indicate that HIA-PSO approach is an efficient method to improve the performance of immune PSO (IA-PSO) in searching ability to global optimum. The proposed HIA-PSO approach performed consistently well on the studies of Benchmark functions, with better results than previously published solutions for these problems.**

Keywords*— *Immune algorithm;PSO; simplex search method;Constrained-optimization; Feasibility-based rule

I. INTRODUCTION

A constrained optimization problem is formally defined as,

$$\text{Minimize } f(x)$$

$$\text{Subject to } g_j(x) \geq 0,\ j=1,2,\cdots,J \quad (1)$$

$$h_k(x) = 0,\ k = 1,2,\cdots,K \quad (2)$$

$$l_i \leq x_i \leq u_i,\ i = 1,2,\cdots,N \quad (3)$$

Where $f(x)$ is the objective function, $g_j(x)$ is the jth inequality constraint, $h_k(x)$ is the equality constraint and $x_i \in [l_i, u_i]$.

Traditional mathematical programming methods such as the Lagrange multiplier methods [1] usually require the derivative information of the objective function and constraints. Besides, the obtained solution often tends to be a local optimum unless the search space is convex. In recent years, evolutionary algorithms (EAs) have attracted much attention for a variety of optimization problems due to their superior advantages.

Recently, a new type of heuristic algorithm, the so called immunity algorithm (IA) [1] has been developed to solve complex optimization problems. The immune network theory is proposed by Jerne . The immune system is the human body's defense system against bacteria, viruses and other disease-causing organisms. Because the IA has the self organizing function to maintain the antibodies diversity, it can overcome the premature phenomenon in the process of looking for the optimal solution and assure fast convergence to the global optimal solution. It has been used successfully in all kinds of optimization problems [2].

The particle swarm optimization (PSO) [3] is a new evolutionary computation technique and has been introduced in various optimization problems in recent years .This algorithm combines the social psychology principles in socio-cognition human agents and evolutionary computations. It is initially motivated by the behavior of organisms, such as a fish school and bird flock. PSO has a flexible and well-balanced mechanism to enhance the global and local exploration abilities. The algorithm begins with randomly generating an initial population, which is composed of a number of candidate solutions. It possesses a constructive cooperation and sharing information relatively between particles of the population [4].

In this paper, the performance of HIA-PSO when applied to constrained problems is investigated. HIA-PSO uses an adaptive penalty function to handle the constraints. HIA-PSO is compared against other approaches proposed in the literature on 13 Benchmark functions.

The rest of the paper is organized as follows: Section II provides an overview of immune algorithm (IA), particle swarm optimization. The self-adaptive penalty function and HIA-PSO are presented in Section III. Results of the experiments are presented and discussed in Section IV. Finally, Section V concludes the paper.

II. IMMUNE ALGORITHM and PARTICLE SWARM OPTIMIZATION

A. Overview of immune algorithm[5]

The immune algorithm (IA) has the desirable characteristics as an optimization tool and offers significant advantages over the traditional methods. They are inherently robust and have been shown to efficiently search the large solution space containing discrete and continuous parameters and non-linear constraints, without being trapped in local minima [6]. The IA may be used to solve a combinatorial optimization problem. In the IA, antigen represents the problem to be solved. An antibody set is generated where each member represents a candidate solution. Also, affinity is the fit of an antibody to the antigen. In the IA, the role of antibody lies in eliminating the antigen, while the lymphocyte helps to produce the antibody. In the immune system, there are two kind of lymphocyte: T and B; where

978-1-4244-6439-5/10/$26.00 ©2010 IEEE

each of them has its own function. The T lymphocytes develop in bone marrow and travel to thymus to mature. The B lymphocytes develop and mature within the bone marrow. The main purpose of the immune system is to recognize all cells within the body and categorize those cells as self or non-self. Self or non-self antigens are those cells that originally belong to the organism and are harmless to its functioning. The disease-causing elements are known as non-self. Both B-cells and T-cells have receptors that are responsible for recognizing antigenic patterns by different functions. The attraction between an antigen and a receptor cell (or degree of binding) is known as affinity. To handle the infection successfully and effectively, both B-cells and T-cells may be required. After successful recognition, cells capable of binding with non-self antigens are cloned. In the IA, the elements of the population undergo mutations resulting in a subpopulation of cells that are slightly different. In the first step, N antibodies are generated randomly and evaluated using a suitable fitness function. While the fitness function of all antibodies is known, new population is generated through three steps: replacement, cloning and hyper mutation.

B. Particle swarm optimization

Particle Swarm Optimization (PSO) was introduced in 1995 by Eberhart and Kennedy as a numerical optimization algorithm [3].Based on the simulation of simplified social models such as a bird flock and fish school, PSO is first introduced in as a novel evolutionary computation technique with the mechanism of individual improvement, population cooperation and competition[7]. The main ideas behind the development of PSO stem from the fields of social psychology and evolutionary computation[8]. In PSO, multiple candidate individuals called particles coexist and collaborate simultaneously, where the position of each particle denotes a decision vector for the original problem. The trajectory of each particle in the search space is dynamically adjusted by updating the velocity of each particle, according to its own flying experience as well as the experience of neighboring particles (built through tracking and memorizing the best position encountered). Therefore, PSO combines the local search technique (by the particle's own experience) and the global search method (by the neighboring experience) to well balance the exploration and exploitation and finally achieves the global optimum. The core operations of PSO are two updating equations of the velocity and position for each particle. In detail, suppose that the position and the velocity of the ith particle in the dimensional search space are represented as $X_i = [x_{i,1}, x_{i,2}, \ldots, x_{i,d}]^T$ and $V_i = [v_{i,1}, v_{i,2}, \ldots, v_{i,d}]^T$ respe ctively. Each particle's own best historical position (pbest) is denoted by $P_i = [p_{i,1}, p_{i,2}, \ldots, p_{i,d}]^T$, and the best historical position that the entire swarm has passed (gbest) is denoted by $P_g = [p_{g,1}, p_{g,2}, \ldots, p_{g,d}]T$. The new velocity of each particle is calculated as follows:

$$v_{i,j}(k+1) = \omega v_{i,j}(k) + c_1 r_1 (p_{i,j} - x_{i,j}(k)) + c_2 r_2 (p_{g,j} - x_{i,j}(k)) \ , \quad j=1,2,\cdots,d \tag{4}$$

where c_1 and c_2 are two positive constants called acceleration coefficients, ω is called the inertia factor, ω is the key factors to affect the convergence of PSO, an improved formula of ω is introduced in the paper so as to improve the global convergence[9], The parameter ω is modified by the formula (5)

$$\omega(t+1) = 4.0 \times \omega(t) \times (1 - \omega(t)), \quad \omega(t) \in (0,1) \tag{5}$$

r_1 and r_2 are two independent random numbers uniformly distributed in the range of $[0,1]$. After the velocity is updated, the new position of each particle for the next generation is determined according to the following equation:

$$x_{i,j}(k+1) = x_{i,j}(k) + v_{i,j}(k+1), \ j=1,2,\cdots,d \tag{6}$$

C. Nelder-Mead simplex search method

The simplex search method was firstly proposed by Spendley, Hext, and Himsworth in1962 and refined in1965 by Nelder and Mead [10].A Nelder–Mead simplex optimization algorithm is used in order to optimize the objective function. The simplex algorithm does not need a derivative; only a numerical evaluation of the objective function is required [10]. For example,in three dimensional space, simplex is a tetrahedron determined by four points (vertices) and their interconnecting line segments. At every point the objective function is evaluated. The point with the highest numerical value of all four points is perpendicularly mirrored against the opposite plain segment. This is called a reflection. The reflection can be accompanied with an expansion to take larger steps or with a contraction to shrink the simplex where an optimization valley floor is reached. The optimization procedure continues until the termination criteria are met. The termination criterion is usually the maximum number of reflections with contractions or a tolerance for optimization variables. The algorithm can be implemented in N dimensions, where simplex is a hypercube with N +1 vertex points [11].

III A HYBRID IMMUNE PSO FOR CONSTRAINED OPTIMIZATION PROBLEMS

A. constrained handling method

Due to the simplicity and ease of implementation, the penalty function method has been considered as the most popular technique to handle constraints. A penalty function can be formulated as follows[7]:

$$F(x) = f(x) + \sum_{j=1}^{N} w_j \times G_j \tag{7}$$

Where $f(x)$ is the objective function; G_j denotes the constraint violation for the jth constraint, which is

employed as the penalty term; and w_j denotes the penalty factor.

Since the objective function and the constraint violation are simultaneously considered in the penalty function, the performance of this kind of approach is significantly affected by the penalty factor. However, the suitable penalty factors are usually difficult to determine and problem-dependent.A feasibility-based rule is employed in this paper to handle constraints, which is described as follows[7]:

(1) Any feasible solution is preferred to any infeasible solution.

(2) Between two feasible solutions, the one having better objective function value is preferred.

(3) Between two infeasible solutions, the one having smaller constraint violation is preferred.

Based on the above criteria, objective function and constraint violation information are considered separately. Consequently, penalty factors are not used at all. Moreover, in the first and the third cases the search tends to the feasible region rather than infeasible region, and in the second case the search tends to the feasible region with good solutions. In brief, such a simple rule aims at obtaining good feasible solutions. There is no need to design the additional fitness function in this paper by incorporating the rule into PSO.

B. A Self-adaptive Immune PSO for Constrained Problems

Actually, IA-PSO, which is proposed by involving the immune information processing mechanism into the original particle swarm optimal algorithm, is a kind of heuristic random algorithm with a stronger ability to find the globally excellent result. In an immune particle swarm system, the problem to be solved is regarded as an antigen, and every antibody represents one solution of the problem[13]. At the same time, every antibody is also a particle of the particle swarm. The affinity between the antigen and the antibody, which is weighed by the fitness of particle, reflects the extent of satisfaction of the object function and the constraints. In addition, the affinity between antibody and antibody reflects the difference among particles, namely the diversity of the particle swarm. However, during the renewal process of the particle swarm, it is preferable to preserve the particle with higher fitness. If this kind of particles becomes over concentrated, it is hard to keep the diversity of the swarm, which may plunge the algorithm into partially extreme optimization. So that, in contrast with PSO, on one hand, IA-PSO keeps the particles with different hierarchical fitness maintaining a certain concentration by using immune memory and a self-adjusting mechanism to guarantee the diversity of the particles; on the other hand, IA-PSO introduces immune vaccination, which can improve the ability of searching, and guide the process of evolution. In order to realize the mathematical description of the objective function (1), let D denote the searching space and N denote the total amount in the particle swarm.

To improve the performance of IA-PSO, the simplex method is incorporated into IA-PSO. One major advantage of selecting the simplex method is that only one iteration of a derivative-free local search is performed on each iteration of HIA-PSO. As a result, only a small computation cost is added into the original IA-PSO. The execution of the simplex method not only improves IA-PSO performance on each iteration, but also helps to find a better solution. In HIA-PSO, each particle in IA-PSO is regarded as a point of the simplex. On each iteration of HIA-PSO, the worst particle is replaced by a new particle generated by one iteration of the N–M simplex method. Then, all particles are again updated by IA-PSO. The IA-PSO and N–M simplex methods are performed iteratively. This way, on each iteration of HIA-PSO, the worst particle is replaced by a new particle, which is further enhanced by subsequent IA-PSO operations.

The process of HIA-PSO algorithm is given as follows:

Step1: Initializing the particle swarm..

Step2: Nelder-Mead Simplex Algorithm is used to optimize the points what IA-PSO has searched.

Step3: Calculating a particle's fitness.

Step4: Forming the next new generation particle swarm.

Step5: Immune memory and self-adjustment: First, check whether all the new particles generated meet the constraints; Second, let $M \in Z^+$ denote the number of new particles generated on the basis of the new generation of the swarm, where $M < N$ .Calculate the consistency of every particle. The function to calculate the consistency of the particle is

$$D(x_i) = \frac{1}{\sum_{j=1}^{N+M} \left| f(x_i) - f(x_j) \right|} (j = 1,2,\cdots,N+M) \quad (8)$$

The selecting function on the basis of particle consistency probability can be written as

$$P(x_i) = \frac{\frac{1}{D(x_i)}}{\sum_{i=1}^{N+M} \frac{1}{D(x_i)}} (i = 1,2,\cdots,N+M) \quad (9)$$

Step6: Immune vaccination: There are three main parts: picking-up vaccine, vaccination, and immune selection.

Step7: Judging whether the process of evolution attains the terminal term: On attaining the terminal term, the program will stop; otherwise, it returns to step3, to continue the search..

IV SIMULATION RESULT

The parameters setting of IA-PSO and HIA-PSO are fixed to: populatin size $N = 200$, acceleration factor $c_1 = c_2 = 1.49$,inertia weight factor $\omega(1) = 0.51$,the maximum number of iteration $MAXIT = 1000$.

In order to evaluate the performance of the algorithm HIA-PSO, 13 common Benchmark functions taken from [14]

are used in the experiments. These test functions contain the characteristics which are representative of what could be considered "difficult" constrained optimization problems for an evolutionary algorithm.

ISR is one of current most competitive approaches for constrained optimization, and it can be observed from Table 1 that it has found out the optimum in each run on all functions except g03, g05 and g12. From the results of HIA-PSO in Table 1, HIA-PSO is better than ISR in best, mean, worst values and std.dev. on g03,g05 and g12, and converges to the optimum of g13 in all 100 runs. It can be seen our approach HIA-PSO has distinct superiority over all the functions except g04, g08 and g13 in best, mean, worst values and std.dev. Only in HIA-PSO method, can g05 be searched the optimum 5126.498. The robustness of HIA-PSO is slightly weaker than IA-PSO but much stronger than other five algorithms on g08. The whole performance of HIA-PSO is similar to HIA-PSO, SMES, AI and PSO. We can see that our approach HIA-PSO has absolute superiority over all the aspects of performance except for robustness on g13. The performance of HIA-PSO is better than IA-PSO on all the functions except for g13. We can conclude that the improvement of our approach HIA-PSO is very valuable.

TABLE I COMPARISON PERFORMANCE OF EACH ALGORITHM

Fun	Sta.	Approaches for constrained optimization						
		ISR[14]	SMES[14]	RDE[14]	AI	PSO	IA-PSO	HIA-PSO
g01	Best	**-15.000**	**-15.000**	**-15.000**	**-15.000**	-15.0001	**-15.000**	**-15.000**
	Mean	**-15.000**	**-15.000**	-14.792	**-15.000**	-13.1260	**-15.000**	**-15.000**
	Worst	**-15.000**	**-15.000**	-12.743	**-15.000**	-9.0218	**-15.000**	**-15.000**
	Std.dev	5.8E-14	**0**	NA	6.7E-13	2.4E-2	3.2E-16	**0**
g02	Best	**0.803619**	0.803601	**0.803619**	0.801212	0.800001	**0.803619**	**0.803619**
	Mean	0.782715	0.785238	0.746236	0.783052	0.765631	0.782020	**0.798120**
	Worst	0.723519	0.751322	0.302179	0.723050	0.698547	0.714562	**0.765412**
	Std.dev	2.2E-02	1.7E-02	NA	2.0E-02	8.3E-01	3.1E-02	**9.3E-03**
g03	Best	1.001	**1.000**	**1.000**	1.008	1.007	1.005	**1.0004**
	Mean	1.001	**1.000**	0.640	0.9651	0.9342	1.0005	**1.0004**
	Worst	1.001	**1.000**	0.029	0.8315	0.6210	1.0005	**1.0004**
	Std.dev	**8.2E-09**	2.1E-04	NA	2.1E-03	4.0E-02	3.2E-07	6.5E-08
g04	Best	**-30665.539**	**-30665.539**	-30665.539	-30665.539	-30665.539	-30665.539	**-30665.539**
	Mean	**-30665.539**	**-30665.539**	-30592.154	-30665.539	-30665.004	-30665.539	**-30665.539**
	Worst	**-30665.539**	**-30665.539**	-29986.214	-30636.465	-30605.139	-30665.539	**-30665.539**
	Std.dev	1.1E-11	**0**	NA	2.3E-05	2.5E-04	5.7E-10	6.4E-11
g05	Best	5126.497	5126.599	5126.497	5126.989	5126.6467	5126.497	**5126.498**
	Mean	5126.497	5174.492	5218.729	5129.430	5495.2389	5126.907	**5126.498**
	Worst	5126.497	5304.167	5502.410	5174.492	6272.7423	5127.599	**5126.498**
	Std.dev	7.2E-13	5.0E+01	NA	7.8E-01	4.0E+02	5.7E-05	**0**
g06	Best	**-6961.814**	**-6961.814**	**-6961.814**	**-6961.814**	-6961.831	**-6961.814**	**-6961.814**
	Mean	**-6961.814**	-6961.284	-6367.575	**-6961.814**	-6961.829	**-6961.814**	**-6961.814**
	Worst	**-6961.814**	-6952.482	-2236.950	**-6961.814**	-6961.011	**-6961.814**	**-6961.814**
	Std.dev	1.9E-12	1.9E+00	NA	5.9E-05	1.2E-02	2.3E-14	**0**
g07	Best	**24.306**	24.327	**24.306**	**24.306**	24.327	**24.306**	**24.306**
	Mean	**24.306**	24.475	104.599	**24.306**	24.700	**24.306**	**24.306**
	Worst	**24.306**	24.843	1120.541	**24.306**	25.297	**24.306**	**24.306**
	Std.dev	6.3E-05	1.3E-01	NA	7.3E-02	5.5E-01	7.1E-06	**7.0E-08**
g08	Best	**0.095825**	**0.095825**	0.095825	0.095825	0.095825	0.095825	**0.095825**
	Mean	**0.095825**	**0.095825**	0.091292	0.095825	0.095825	0.095825	**0.095825**
	Worst	**0.095825**	**0.095825**	0.027188	0.095825	0.095825	0.095825	**0.095825**
	Std.dev	2.7E-07	**0**	NA	9.3E-09	4.4E-05	1.9E-10	6.6E-9
g09	Best	**680.630**	**680.630**	**680.630**	**680.630**	680.630	**680.630**	**680.630**
	Mean	**680.630**	680.643	692.472	**680.630**	680.643	**680.630**	**680.630**
	Worst	**680.630**	680.719	839.78	**680.630**	680.686	**680.630**	**680.630**
	Std.dev	3.2E-13	1.6E-02	NA	2.1E-06	2.0E-03	5.8E-11	**4.7E-19**
g10	Best	**7049.248**	7051.903	**7049.248**	**7049.248**	7089.623	**7049.248**	**7049.248**
	Mean	7049.250	7253.047	8442.66	7049.260	7274.856	7049.250	**7049.248**
	Worst	7049.270	7638.366	15580.37	7049.276	7634.115	7049.263	**7049.249**
	Std.dev	3.2E-03	1.4E+02	NA	4.9E-02	1.0E+02	1.1E-03	**4.9E-10**
g11	Best	**0.75**	**0.75**	**0.75**	**0.75**	0.75	0.7499	**0.7499**
	Mean	**0.75**	**0.75**	0.76	**0.75**	0.76	0.7499	**0.7499**
	Worst	**0.75**	**0.75**	0.87	**0.75**	0.96	0.7499	**0.7499**
	Std.dev	1.1E-16	1.5E-04	NA	5.5E-14	1.5E-02	3.3E-16	**0**
g12	Best	1.000	**1.000**	1.000	1.000	1.000	1.000	**1.000**
	Mean	1.000	**1.000**	1.000	1.000	1.000	1.000	**1.000**

	Worst	1.000	**1.000**	1.000	1.000	1.000	1.000	**1.000**
	Std.dev	1.2E-09	**0**	NA	8.7E-08	5.0E-03	1.1E-11	**0**
g13	Best	**0.053942**	0.053986	0.053866	0.053214	0.053986	**0.053942**	**0.053942**
	Mean	0.06677	0.166385	0.747227	0.061241	0.072564	**0.053942**	**0.053942**
	Worst	0.438803	0.468294	2.259875	0.312546	0.587961	**0.053942**	**0.053942**
	Std.dev	7.0E-02	1.8E-01	NA	9.9E-03	8.6E-01	**1.0E-04**	3.1E-03

Note: Fun means Benchmark functions; Sta. means Statistical Characteristics; NA means no available.

V. CONCLUSIONS

In this paper, new HIA-PSO approach is proposed and applied to solve constrained optimization problems. The possibilities of exploring the AI-PSO efficiency combined with the feasibility-based rule are successfully presented, as illustrated by the studies of Benchmark functions.

In this paper, new combinations of PSO and IA and Nelder-Mead simplex search method are employed in well-studied continuous optimization problems of engineering design. Three case studies are evaluated in this work. Our results indicate that HIA-PSO approach handle such problems efficiently in terms of precision and robustness and, in most cases, they outperform the results presented in the literature. Finally, the simulation results on the Benchmark functions and practical examples demonstrate that our approach is competitive and easy to implement. In the future, the convergent proof of the proposed algorithm should be studied.

ACKNOWLEDGEMENTS

The authors are grateful to the support of the National Natural Science Foundation of China under Grant No.90715029 and No.60603053.

REFERENCES

[1] N.K. Jerne. Towards a network theory of the immune system. J Ann Immunol 1974,125C:373–89.

[2] H.G.Xiong, H.Z.Cheng, H.Y.Li. Optimal reactive power flow incorporating static voltage stability based on multi-objective adaptive immune algorithm. Energy Conversion and Management 2008, 49:1175–1181.

[3] J .Kennedy, R .Eberhart. Particle swarm optimization. In: Proc IEEE int conf neural networks, vol. IV, Perth, Australia; 1995,1942–1948.

[4] W.D.Chang, S.P.Shih. PID controller design of nonlinear systems using an improved particle swarm optimization approach. Commun Nonlinear Sci Numer Simulat (2010), doi:10.1016/j.cnsns.2010.01.005.

[5] V. P.Sakthivel. Artificial immune system for parameter estimation of induction motor. Expert Systems with Applications.(2010), doi:10.1016/j.eswa.2010.02.034.

[6] J. S.Chun, , J. P.Lim, J. S.Yoon.Optimal design of synchronous motor with parameter correction using immune algorithm. IEEE Transactions on Energy Conversion, 1999,14(3):610–615.

[7] Q.He, L.Wang. A hybrid particle swarm optimization with a feasibility-based rule for constrained optimization. Applied Mathematics and Computation 2007,186:1407–1422.

[8] K.E. Parsopoulos, M.N. Vrahatis.Parameter selection and adaptation in Unified Particle Swarm Optimization, Mathematical and Computer Modelling, 2007, 46:198–213.

[9] J. Chuanwen, E. Bompard.A hybrid method of chaotic particle swarm optimization and linear interior for reactive power optimisation, Mathematics and Computers in Simulation 2005, 68 :57–65.

[10] J.A.Nelder, R. A Mead.simplex method for function minimization. Computer Journal 7(4) (1965)308–313.

[11] J. Bonnans, J. Fr'ed'eric, C. Gilbert. Numerical Optimization, Theoretical and Practical Aspects, second ed.Springer-Verlag, Berlin Heidelberg,2006.

[12] C.-C. Fuh .Detecting unstable periodic orbits embedded in chaotic systems using the simplex method.Communications in Nonlinear Science and Numerical Simulation 14 (2009) 1032–1037.

[13] A.Q.Li, L.P.Wang, J.Q.Li. Application of immune algorithm-based particle swarm optimization for optimized load distribution among cascade hydropower stations. Computers and Mathematics with Applications. 2009, 57:1785-1791.

[14] M .Zhang, W.J.Luo, X.F.Wang. Differential evolution with dynamic stochastic selection for constrained optimization. Information Sciences 2008,178 :3043–3074.

[15] Q.He, L.Wang. A efficient co-evolutionary particle swarm optimization for constrained engineering designed problems. Engineering Applications of Artificial Intelligence, 2007,20:89-99.

[16] A.D.Belegundu. A study of mathematical programming methods for structural optimization. Dept.of Civil and Engineering ,Univ.of Iowa,Iowa city,Iowa,1982.

[17] J.S. Arora. Introduction to optimization design.New York:McGraw-Hill,1989.

[18] C.A.C.Colleo. Use of a self-adaptive penalty approach for engineering optimization problems. Computers in Industry, 2000,41:113-117.

[19] C.A.C.Colleo. Montes EM.Constraint-handling in genetic algorithms through the use of dominance-based tournament selection. Advanced Engineering Informatics,2002,16:193-23.

[20] C.A.C.Colleo, R.L.Becerra. Efficient evolutionary optimization through the use of a cultural algorithm.Engineering Optimization, 2004,36:219-236.

[21] K.Deb. A.S.Gene. a robust optimal design technique for mechanical component design.In:Dasgupta D and Michalewicz Z(Eds.), Evolutionary Algorithms in Engineering Applications,Berlin:Springer-Verlag,1997,497-514.

[22] B.K. Kannan, S.N.Kramer. An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its application to mechanical design. Trans.of the ASME,Journal of Mechanical Design.1994,116:318-320.

[23] E.Sandgren. Nonlinear integer and discrete programming in mechanical design.In:Proc.of The ASME Design Technology Conference. 1988,95-105.