A Novel Genetic Algorithm Based on Immunity and its Application

Fengqiang Zhao [1,2], Guangqiang Li[1], Jialu Du[1], Chen Guo[1], Hongying Hu[2], Ajith Abraham[3, 4]
[1]School of Information Science and Technology, Dalian Maritime University, Dalian 116026, China
[2]School of Electromechanical & Information Engineering, Dalian Nationalities University, Dalian 116600, China
[3]Machine Intelligence Research Labs, Scientific Network for Innovation and Research Excellence, WA, USA
[4]IT For Innovations, VSB - Technical University of Ostrava, Czech Republic
E-mail: {fqzhao2002, gqlimail}@163.com, ajith.abraham@ieee.org

Abstract—In this paper, a novel genetic algorithm based on immunity (GABI) on the basis of parallel genetic algorithms (PGA) is proposed in order to overcome some defects of them, such as premature and slow convergence rate. The global performance of the algorithm is improved by introducing immunity theory into PGA. This is revealed in the following two aspects. One is that the immune selection based on proposed adjustable geometric-progression rank-based selection can prevent the algorithm from premature. The other is that convergence rate can be accelerate by individual migration strategy between subpopulations based on immune memory mechanism. In this algorithm, the idea of multiple subpopulations evolution based on improved adaptive crossover and mutation is adopted. To be hybridized with the Powell method can further improve local searching performance of the algorithm. An example of layout design shows that GABI is feasible and effective.

Keywords - genetic algorithms; immunity; hybrid methods; parallel computing; layout

I. INTRODUCTION

Genetic algorithms are typical swarm intelligence techniques based on the mechanics of natural selection and natural genetics, which combines artificial survival of the fittest concept with genetic operations abstracted from nature [1-2]. Due to strong flexibility and robustness, genetic algorithms have been widely used in the solution of combinatorial optimization, production scheduling, machine learning, optimal control, image processing and so on. But there still exist some defects of genetic algorithms, such as premature and slow convergence rate. This has hampered further application and development of the algorithms to some extent. To overcome them, a novel genetic algorithm based on immunity (GABI) is proposed on the basis of parallel genetic algorithms (PGA) [3] through the introduction of immune principle, hybrid strategy and other improvement measures. It aims at solving complex engineering optimization problems (e.g. packing and layout design problems) more effectively.

II. GENETIC ALGORITHM BASED ON IMMUNITY

Immunity-based algorithms originated in 1990s have many good characteristics [4-6]. They can embody immune memory, extraction and inoculating efficient antibodies as well as antibody inhibition and promotion mechanism in the biological immune systems. So the genetic algorithms based on immunity can effectively prevent premature, accelerate convergence rate and improve overall performance of algorithms. Traditional immune genetic algorithms are almost all serial algorithms. In this paper, we introduce immune principle into parallel genetic algorithms and put forward following improvements on current immune algorithms.

- We adopt the simple and easy Euclidean distance to calculate affinities between antibodies (i.e. individuals) for convenient to engineering design.
- We present correction formula for calculating individual concentration and the immune selection operator based on proposed adjustable geometric-progression rank-based selection.
- We propose the individual migration strategy according to the immune memory between subpopulations in GABI.

Moreover, some other measures are taken in GABI for the purpose of further improving the proposed algorithm performance, such as multiple subpopulations evolution on the basis of improved adaptive crossover and mutation as well as the hybrid strategy with Powell method [7].

A. Adjustable Geometric-progression Rank-based Selection

In traditional rank-based selection operator of genetic algorithms, a probability assignment table should be preset [1, 3]. But there is no deterministic rule for design of the table. And it is difficult for traditional rank-based model to make the selection probabilities of individuals adaptively changed along with evolution process. We introduce the concept of adjustable geometric-progression rank-based selection. It can overcome the above-stated shortcomings of traditional rank-based selection.

There is one independent parameter in this operator, dominance coefficient λ. It denotes the ratio of the maximal individual selection probability $P_{\max}$ to the minimal one $P_{\min}$ within a generation, i.e. $P_{\max}=\lambda P_{\min}$. It numerically shows the superiority that the better individuals are reproduced into the next generation during selection operation and it is changeable along with algorithm evolution. In the early stage, lesser λ can maintain population diversity and prevent the algorithm from premature; while in the late stage, greater λ can benefit accelerating convergence. Let $\lambda = f(K)$, K and f denote the generation number and an increasing function respectively. We adopt linear increasing function here. Assume that $\lambda_{\max}$ and $\lambda_{\min}$ denote the maximum and minimum of dominance coefficient respectively, then

$$\lambda = \frac{(K-1)(\lambda_{\max} - \lambda_{\min})}{K_{\max} - 1} + \lambda_{\min} \tag{1}$$

where $K_{\max}$ is the maximal generation number set in an algorithm. Our experiments show that $\lambda_{\max}$ and $\lambda_{\min}$ may be chosen in the interval [6, 15] and [1.5, 5] respectively.

To calculate the selection probability of every individual, first of all, we should arrange all the individuals within a population in descending order based on their fitness values. Let Ind_i represent the ith individual within a population as well as F_i and P_i represent its fitness and selection probability respectively. There exist Ind_i (i=1,2, …, M) and $F_i > F_{i+1}$(i=1,2, , …,M-1). M is the population size. Suppose that the selection probability values of all the individuals form a geometric progression with common ratio q, $0<q<1$. Assume that the first term P_1 of this geometric progression is a as well as its general item P_n is aq^{n-1}. Obviously, the sum of all the individual selection probability is 1, i.e. subtotal of this geometric progression

$$S_M = \sum_{i=0}^{M-1} aq^i = a + aq + \cdots + aq^{M-1} = \frac{a - aq^M}{1-q} = 1 \tag{2}$$

As above stated, according to the concept of dominance coefficient, there exists $P_{\max} = \lambda P_{\min}$. Here $P_{\max} = P_1 = a$, $P_{\min} = P_M = aq^{M-1}$, then

$$a = \lambda a q^{M-1} \tag{3}$$

So we get

$$q = \lambda^{\frac{1}{1-M}} \tag{4}$$

Substituting above formula into formula (2), it is easy to find that

$$a = \left(1 - \lambda^{\frac{1}{1-M}}\right) \Big/ \left(1 - \lambda^{\frac{M}{1-M}}\right) \tag{5}$$

Therefore we obtain

$$p_i = \left[\left(1 - \lambda^{\frac{1}{1-M}}\right) \Big/ \left(1 - \lambda^{\frac{M}{1-M}}\right)\right] \left(\lambda^{\frac{1}{1-M}}\right)^{i-1} \quad i = 1,2,\ldots,M \tag{6}$$

In the process of selection, we firstly reproduce the best individual of current generation and put its copy into next generation directly based on elitist model. Then figure out selection probabilities of all individuals according to formula (6). Finally generate the remaining M-1 individuals of next generation by fitness proportional model. Compared with traditional rank-based selection, the advantage of proposed selection operator is that it can conveniently change the selection probabilities of individuals by changing dominance coefficient and is more adaptive to the algorithm run.

B. Multiple Subpopulations Evolution and Individual Migration Strategy

1) Improved adaptive crossover and mutation

To prevent genetic algorithms from premature effectively as well as protect superior individuals from untimely destruction, Srinivas and Patnaik [8] proposed the concept of adaptive crossover and mutation. But according to these operators, crossover and mutation rate of the best individual among a population are both zero. It may lead to rather slow evolution in the early stage. To avoid its occurrence, it's better to let the individuals possess due crossover and mutation rates, whose fitness values are equal or approximate to the maximal fitness. Therefore, based on [8], improved adaptive crossover rate P_c and mutation rate P_m are presented, see (7) and (8).

$$P_c = \begin{cases} k_1 \exp\left[\dfrac{(F_{\max} - F')}{F_{\max} - F_{avg}}\left(\ln k_3 - \ln k_1\right)\right], & F' \geq F_{avg} \\ k_3, & F' < F_{avg} \end{cases} \tag{7}$$

$$P_m = \begin{cases} k_2 \exp\left[\dfrac{(F_{\max} - F)}{F_{\max} - F_{avg}}\left(\ln k_4 - \ln k_2\right)\right], & F \geq F_{avg} \\ k_4, & F < F_{avg} \end{cases} \tag{8}$$

where $F_{\max}$ and F_{avg} denote the maximal and average fitness of current population. F' denotes the greater fitness of the two individuals that take part in crossover operation. F denotes the fitness of the individual that take part in mutation operation. k_1, k_2, k_3, k_4 are constants. And there exist $0<k_1, k_2, k_3, k_4 \leq 1.0$, $k_1<k_3$, $k_2<k_4$.

2) Multiple subpopulations evolution

Simulating the varied and colorful biological communities in nature, we adopt the idea of multiple subpopulations evolution [9] and classify all the subpopulations of proposed algorithm into four classes according to their crossover and mutation rates (P_c and P_m). Suppose that there is only one subpopulation within every class, named α, β, γ and δ subpopulation respectively. Their parametric features are shown in Table 1.

TABLE 1. PARAMETRIC FEATURES OF FOUR CLASSES OF SUB-POPULATIONS

Subpopulation	**Class α**	**Class β**	**Class γ**	**Class δ**
Crossover rate	k_1=0.8	k_1=0.5	k_1=0.2	k_1=0.1
	k_3=1.0	k_3=0.8	k_3=0.5	k_3=0.2
Mutation rate	k_2=0.3	k_2=0.2	k_2=0.1	k_2=0.05
	k_4=0.4	k_4=0.3	k_4=0.2	k_4=0.1
Initial fitness	Minimal	Medium	Greater	Maximal

According to their properties of initial fitness as well as crossover and mutation rates, we can see that the fitness values of initial individuals of class α subpopulation are the minimal among those of subpopulations of the four classes. But this subpopulation has the highest P_c and P_m, so it is easier for it to explore the new parts of solution space and enhance the possibility of discovering global optima. As well as, it can guard against premature convergence. The initial individuals of class γ subpopulation are with relatively greater fitness values. Because this subpopulation has relatively lower P_c and P_m, it is easier for it to keep the stability of individuals. The function of class γ subpopulation is mainly to consolidate local search. Class β subpopulation is a transitional subpopulation. And class δ subpopulation is also called memory subpopulation for it corresponds to memory cells in immune systems. It is made up of the initial individuals with the maximal fitness values among those of subpopulations of the four classes. In the process of evolution, this subpopulation saves the superior individuals obtained by subpopulations of the above-mentioned other three classes. At the same time, class δ subpopulation is also evolving itself. But its P_c and P_m are the lowest. The function of class δ subpopulation is to simulate the immune memory function and keep the stability and diversity of the superior individuals.

After random initialization, GABI arranges all the generated initial individuals according to their fitness values. The initial individuals with the maximal fitness values are allocated to class δ subpopulation; the initial individuals with relatively greater fitness values are allocated to class γ subpopulation; the initial individuals with the minimal fitness values are allocated to class α subpopulation; the rest of initial individuals are allocated to class β subpopulation.

3) Individual migration strategy based on immune memory between subpopulations

At intervals of given migration cycle, GABI copies the current best individuals in class α, β and γ subpopulations and remembers (saves) them into class δ subpopulation, then update this memory subpopulation (eliminate the inferior individuals from it) and keep the same subpopulation size. Meanwhile, simulating inoculation, GABI selects some individuals from the memory subpopulation and make them migrate to class α, β and γ subpopulations respectively. The migration individuals will replace the inferior individuals of the three subpopulations respectively as well. This migration strategy can accelerate the convergence rate of the algorithm.

In addition, we set a generation control parameter, denoted by K_m. When generation number K is multiples of K_m, GABI merges all the subpopulations together and arrange all the individuals according to their fitness values. Then GABI reallocates individuals to every subpopulation respectively according to their fitness values.

C. Antibody Concentration and Immune Selection

1) Antibody affinity and antibody concentration

Here Antibodies are exactly individuals. They have the same concept and all represent solutions of a given problem. Antibody affinity ay_{vw} defined as follows indicates similar extent between antibody v and antibody w.

$$ay_{vw} = 1/[1 + H(2)] \tag{9}$$

The range of ay_{vw} is within $(0,1]$. If the value ay_{vw} is higher then the antibody v is more similar with antibody w. At present, H (2) in formula (9) is mostly calculated by average information entropy based on antibody v and w. In fact, as above stated, antibody affinity denotes similar extent between antibodies. In other words, H (2) represents the distance between two antibodies. It can be calculated by average information entropy and also can be calculated by other methods, if two conditions are satisfied. One is H (2) ≥ 0, and H (2) $=0$ indicates that the genes of two antibodies are exactly the same. The other is that greater differences between genes of two antibodies can lead to greater value of H (2). In order to simplify the calculation and be easy for engineering realization, we adopt Euclidean distance to calculate affinities. Let antibody $\boldsymbol{v}=(v_1, v_2,\ldots, v_n)$ and antibody $\boldsymbol{w}=(w_1, w_2,\ldots,w_n)$, then

$$H(2) = \sqrt{\sum_{i=1}^{n}(v_i - w_i)^2} \tag{10}$$

Suppose that M denotes the population size. Concentration c_v of antibody v in its population is usually defined as follows presently.

$$c_v = \frac{1}{M}\sum_{w=1}^{M} ay_{vw} \tag{11}$$

Obviously there exists $c_v \in (0,1]$. The concept of antibody concentration is applied to the following immune selection. In order to avoid oscillation during the later period of proposed algorithm and facilitates algorithm convergence, antibody concentration c_v should tend to 1 ultimately along with increase in the value of generation number K. Therefore, we present a correction for (11) as follows.

$$c_v = \left(\frac{1}{M}\sum_{w=1}^{M} ay_{vw}\right)^{(1-\frac{K}{K_{\max}})\cdot\beta} \tag{12}$$

where β is a system parameter and usually set β=0.5.

2) Immune selection

The procedure for immune selection of GABI can be described below.

- Calculate the fitness value of every antibody (individual) in the population, i.e. F_v, v=1,2,…, M;
- According to the above formula (12), calculate the concentration of every antibody in the population, i.e. c_v, v=1,2,…, M;
- Calculate the adjusted fitness F'_v of every antibody in the population and there exist $F'_v = F_v/c_v$, v=1,2,…, M;
- Generate the next population based on adjusted fitness values by above proposed adjustable geometric-progression rank-based selection.

Compared with traditional selection operators, the above immune selection can reflect self regulation function of antibody inhibition and promotion in immune systems. Namely the antibodies with greater fitness values and lower concentration will be promoted and their survival probabilities become larger. On the contrary, the antibodies with lower fitness values and higher concentration will be inhibited and their survival probabilities become smaller. Consequently proposed immune selection can effectively maintain population diversity and prevent GABI from premature convergence.

D. Hybrid Strategy

To further improve local search ability of the algorithm, it is necessary to apply hybrid strategy. Taking the matching problem into consideration, we hybridize Powell method with proposed algorithm. Powell method possesses relatively fast local convergence rate and doesn't involve derivative information. Allowing for the problem of computational efficiency, the hybrid algorithm should give full play to the global search ability of genetic algorithm in the early stage, while to the local search ability of Powell method in the late stage. Therefore in GABI, we set parameter K_P and select N_P individuals as initial points to search C_P turns by Powell method at intervals of K_P generations. To enhance the local search ability of GABI in the late stage and accelerate convergence rate, N_P and C_P are set in direct proportion to generation number K in the proposed algorithm.

E. The Procedure of Proposed Algorithm

Flow chart of the proposed genetic algorithm based on immunity (GABI) is shown in Fig. 1.

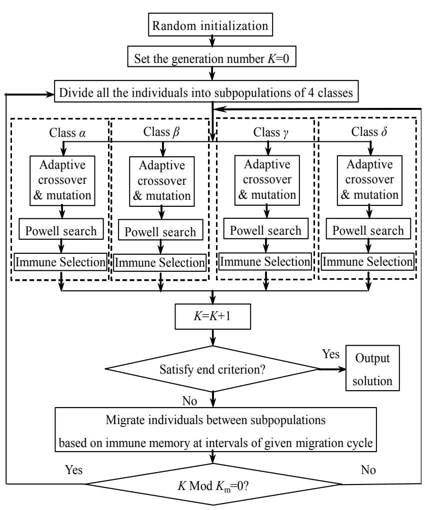

Figure1. Flow chart of the proposed GABI

III. NUMERICAL EXAMPLE

The engineering background of this example is the packing and layout design of printed circuit boards (PCB) and plant equipments. Assume that there are n objects named $A_1, A_2, \ldots, A_n$ and the weight between A_i and A_j is w_{ij}, i, j=1,2,…, n. Try to locate each object such that the value of expression $S+\lambda_w C$ of a layout scheme is as small as possible and the constraints of no interference between any two objects are satisfied. Here S is the area of enveloping rectangle of a layout scheme. λ_w is a weight factor and C is the sum of the products of d_{ij} multiplied by w_{ij}, i.e.

$$C=\sum_{i=1}^{n-1}\sum_{j=i+1}^{n} d_{ij} w_{ij} \tag{13}$$

where d_{ij} is the distance between object A_i and A_j.

Suppose that (x_i, y_i) is the coordinates of the center of the object A_i. The mathematical model for this problem is

$$\begin{aligned} &\text{Find } \boldsymbol{X}=(x_i, y_i)^{\mathrm{T}},\ i\in\{1,2,\ldots,n\} \\ &\min\ f(\boldsymbol{X})=S+\lambda_w C \\ &\text{s.t. } \operatorname{int}A_i \cap \operatorname{int}A_j=\varnothing \quad i\neq j,\quad i,j\in\{1,2,\ldots,n\} \end{aligned} \tag{14}$$

where intA_i presents the interior of object A_i.

Quoted from [10], 15 circular objects are contained in this example. Let λ_w=1. The radii of objects are $r_1=r_3=r_{10}$=12 mm, $r_2=r_4$=3 mm, $r_5=r_{13}=r_{14}$=9 mm, $r_6=r_{12}=r_{15}$=10 mm, r_7=7 mm, r_8=8 mm, r_9=4 mm, r_{11}=6mm. The weight matrix is

$$\boldsymbol{W}=\begin{pmatrix}
0 & 0 & 0 & 98 & 98 & 0 & 81 & 0 & 92 & 93 & 45 & 61 & 99 & 84 & 27 \\
0 & 0 & 34 & 0 & 0 & 0 & 93 & 44 & 0 & 0 & 33 & 60 & 0 & 0 & 56 \\
0 & 34 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 85 & 0 & 65 & 39 & 0 & 50 \\
98 & 0 & 0 & 0 & 91 & 50 & 5 & 24 & 73 & 0 & 4 & 0 & 0 & 31 & 23 \\
98 & 0 & 0 & 91 & 0 & 37 & 0 & 16 & 78 & 95 & 0 & 0 & 73 & 32 & 0 \\
0 & 0 & 0 & 50 & 37 & 0 & 0 & 35 & 0 & 31 & 0 & 0 & 0 & 48 & 0 \\
81 & 93 & 0 & 5 & 0 & 0 & 0 & 94 & 33 & 34 & 26 & 61 & 0 & 87 & 87 \\
0 & 44 & 0 & 24 & 16 & 35 & 94 & 0 & 91 & 0 & 0 & 0 & 59 & 39 & 0 \\
92 & 0 & 0 & 73 & 78 & 0 & 33 & 91 & 0 & 0 & 30 & 0 & 0 & 0 & 0 \\
93 & 0 & 85 & 0 & 95 & 31 & 34 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
45 & 33 & 0 & 4 & 0 & 0 & 26 & 0 & 30 & 0 & 0 & 0 & 21 & 35 & 2 \\
61 & 60 & 65 & 0 & 0 & 0 & 61 & 0 & 0 & 0 & 0 & 0 & 56 & 0 & 43 \\
99 & 0 & 39 & 0 & 73 & 0 & 0 & 59 & 0 & 0 & 21 & 56 & 0 & 1 & 0 \\
84 & 0 & 0 & 31 & 32 & 48 & 87 & 39 & 0 & 0 & 35 & 0 & 1 & 0 & 0 \\
27 & 56 & 50 & 23 & 0 & 0 & 87 & 0 & 0 & 0 & 2 & 43 & 0 & 0 & 0
\end{pmatrix} \tag{15}$$

To compare the performance of GABI with that of PGA objectively, we adopt GABI and PGA that possesses four subpopulations (same as GABI) to solve this example respectively and subpopulation sizes of both algorithms are identical. Moreover, any relevant contents of the two algorithms, such as encoding scheme, fitness function and migration cycle, that may be identical are selected as the same. The migration strategy of PGA we adopted in this paper is as follows. At intervals of given migration cycle, PGA copies several superior individuals of every subpopulation, sends to another arbitrarily taken subpopulation and replaces the inferior individuals of the subpopulation. All computation is performed on PC with CPU at 2.1GHz and RAM size of 2GB. Both algorithms are calculated 20 times respectively. The best layouts among 20 optimal results by them are in Table 2 and the corresponding best geometric layout patterns are shown in Fig. 2. For the best layout by PGA, S, C and computation time t are 5884.01mm^2, 91235.20 and 25.31s; for the best layout by GABI, S, C and t are 5602.03mm^2, 80627.64and 23.56s. When obtained $S\leq$5884.01mm^2, $C\leq$91235.20 by GABI, it takes 19.93s. So in the sense of best results, to reach the same precision, GABI reduces the cost of time by 21.26% compared with PGA. Table 3 lists relevant average values of

20 optimal results of the example obtained by two algorithms. In Table 3, ΔS and K represent the interference area and elapsed generation number for an optimal result respectively. Table 3 shows that compared with PGA, on an average, GABI reduces S, C and elapsed generation number K by 6.73%, 10.94% and 25.36%, i.e. from 6231.91mm^2 to 5812.43mm^2, from 96744.74 to 86163.08 and from 698 to 521 respectively.

TABLE 2. BEST LAYOUTS OF THE EXAMPLE BY THE TWO ALGORITHMS

No.	The best layout by PGA		The best layout by GABI	
	x_i/mm	y_i/mm	x_i/mm	y_i/mm
1	9.45	21.27	5.25	-3.35
2	-11.88	-9.29	-18.34	2.30
3	-31.27	12.15	-31.85	-22.46
4	-0.63	2.32	23.27	24.31
5	-11.51	20.96	17.62	13.77
6	-16.89	-23.30	36.84	-24.30
7	-22.77	-7.35	-10.04	7.88
8	-13.02	4.03	1.16	17.89
9	-2.08	9.59	28.90	20.22
10	15.17	-1.96	29.20	-3.65
11	-3.46	-6.17	-13.20	20.46
12	35.12	-11.20	-0.74	-24.45
13	19.87	-22.83	-14.26	-11.03
14	1.97	-20.13	18.07	-21.43
15	33.31	10.48	-27.03	11.93

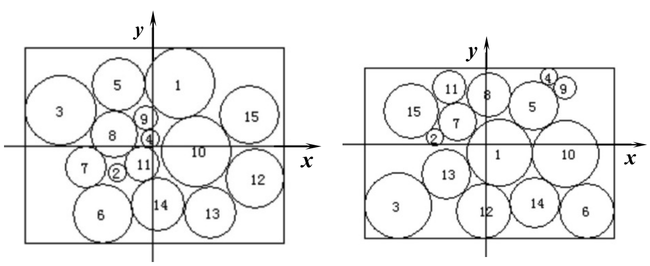

Figure 2. The obtained best layout patterns of the example by PGA (*left*) and GABI (*right*)

TABLE 3. COMPARISON OF AVERAGE VALUES OF 20 OPTIMAL RESULTS OF THE EXAMPLE BY THE TWO ALGORITHMS

Algorithms	S/mm^2	C	ΔS/mm^2	K
PGA	6231.91	96744.74	0	698
GABI	5812.43	86163.08	0	521

IV. CONCLUSIONS

To overcome defects of PGA, we take several measures on it and propose a novel algorithm named GABI. These measures involve introducing immune selection operator based on proposed adjustable geometric-progression rank-based selection, hybrid strategy, adaptive multiple subpopulations evolution and individual migration strategy based on immune memory. The numerical example of layout design shows that GABI is feasible and effective. It is really superior to PGA in accuracy and convergence rate. Because proposed GABI is a universal algorithm, it also can be adopted to solve other complex optimization problems. In the future, we plan to investigate more hybrid methods [11-16] to improve the efficiency of the current approach.

ACKNOWLEDGEMENTS

We would like to express our gratitude to the National Natural Science Foundation of China (No. 51079013, No. 61074053), the Fundamental Research Funds for the Central Universities of China (No. DMU2011NQ030, No. DC120101014, No. DC110320), the Applied Basic Research Program of Ministry of Transport of China (No. 2011-329-225-390，No. 2012-329-225-070), and the Higher Education Research Fund of Education Department of Liaoning Province of China (No. LT2010013) for financial support of our work.

REFERENCES

[1] R. L. Haupt and S. E. Haupt, Practical Genetic Algorithms, 2nd ed., NJ: John Wiley & Sons, Inc., 2004.

[2] R. Ghaemi, N. B. Sulaiman, H. Ibrahim and N. Mustapha, “A review: Accuracy optimization in clustering ensembles using genetic algorithms,” Artificial Intelligence Review, Vol. 35, no. 4, pp. 287-318, 2011.

[3] D. S. Knysh and V. M. Kureichik, “Parallel Genetic Algorithms: A survey and Problem State of the Art,” Journal of Computer and Systems Sciences International, Vol. 49, no. 4, pp. 579-589, 2010.

[4] C. J. Shih and T. L. Kuan, “An Immunity Based Hybrid Evolutionary Algorithm for Engineering Optimization,” Tamkang Journal of Science and Engineering, Vol. 9, no. 1, pp. 25-36, 2006.

[5] X. Xu and C. Li, “Research on Immune Genetic Algorithm for Solving the Job-shop Scheduling Problem,” International Journal of Advanced Manufacturing Technology, Vol. 34, no. 7-8, pp. 783-789, 2007.

[6] S. Kumar, P. Asokan and S. Kumanan, “An artificial immune system-based algorithm to solve linear and loop layout problems in flexible manufacturing systems,” International Journal of Product Development, Vol. 10, no. 1-3, pp. 165-179, 2010.

[7] J. Zhang and Y. Zhou, “An artificial glowworm swarm optimization algorithm based on Powell local optimization method,” Pattern Recognition and Artificial Intelligence, 24(5), pp. 680-684, 2011.

[8] M. Srinivas and L. M. Patnaik, “Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms,” IEEE Transactions on Systems, Man and Cybernetics, Vol. 24, no.4, pp. 656-667, 1994.

[9] G. Li, J. Huo and H. Teng, “A parallel hybrid genetic algorithm and its application to layout design,” Computer Engineering, Vol. 29, no.17, pp. 6-8, 2003.

[10] G. Li, “Evolutionary Algorithms and their Application to Engineering Layout Design,” Postdoctoral Research Report, Tongji University, Shanghai, China, 2005.

[11] Y. Chen, A Abraham, B. Yang, Hybrid flexible neural - tree - based intrusion detection systems, International Journal of Intelligent Systems 22 (4), pp. 337-352, 2007.

[12] A. Abraham, N.S. Philip, P. Saratchandran, Modeling Chaotic Behavior of Stock Indices Using Intelligent Paradigms, International Journal of Neural, Parallel & Scientific Computations, USA, Volume 11, Issue (1&2), pp. 143-160, 2003.

[13] H. Liu, A. Abraham, A.E. Hassanien, Scheduling jobs on computational grids using a fuzzy particle swarm optimization algorithm, Future Generation Computer Systems, 26 (8), pp. 1336-1343, 2010

[14] A Abraham, Intelligent systems: Architectures and perspectives, Recent advances in intelligent paradigms and applications, pp. 1-35, 2003.

[15] C Grosan, A Abraham, A new approach for solving nonlinear equations IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, Volume 38, No. 3, pp. 698-714, 2008.

[16] F Xhafa, E Alba, B Dorronsoro, B Duran, A Abraham, Efficient batch job scheduling in grids using cellular memetic algorithms, Metaheuristics for Scheduling in Distributed Computing Environments, pp. 273-299, 2008.