

A New Method for Fastening the Convergence of Immune Algorithms Using an Adaptive Mutation Approach

Mohammed Abo-Zahhad¹, Sabah M. Ahmed¹, Nabil Sabor¹, Ahmad F. Al-Ajlouni²

¹Electrical and Electronics Engineering Department, Faculty of Engineering, Assiut University, Assiut, Egypt; ²Communication Engineering Department, Hijjawi Faculty for Engineering Technology, Yarmouk University, Irbid, Jordan.
Email: zahhad@yahoo.com

Received December 28th, 2010; revised November 29th, 2011; accepted December 12th, 2011

ABSTRACT

This paper presents a new adaptive mutation approach for fastening the convergence of immune algorithms (IAs). This method is adopted to realize the twin goals of maintaining diversity in the population and sustaining the convergence capacity of the IA. In this method, the mutation rate (p_m) is adaptively varied depending on the fitness values of the solutions. Solutions of high fitness are protected, while solutions with sub-average fitness are totally disrupted. A solution to the problem of deciding the optimal value of p_m is obtained. Experiments are carried out to compare the proposed approach to traditional one on a set of optimization problems. These are namely: 1) an exponential multi-variable function; 2) a rapidly varying multimodal function and 3) design of a second order 2-D narrow band recursive LPF. Simulation results show that the proposed method efficiently improves IA's performance and prevents it from getting stuck at a local optimum.

Keywords: Adaptive Mutation; Immune Algorithm; Convergence; Traditional Mutation

1. Introduction

Finding an appropriate set of parameter values for evolutionary algorithms (EAs) has been a longstanding major challenge of the evolutionary computation community. Such difficulty has directed researchers' attention towards devising an automated ways of controlling EAs parameters [1]. Immune Algorithm (IA) is one of recently EAs that based on the mechanism of the amalgamation between antigen and antibody in biologic immune system [2]. IAs have been used widely and successfully in many computational intelligence areas including medical data processing, biomedical and satellite image processing, spectrum analysis and design of digital filters [3-6]. The performance of IAs depends on many factors, such as selection schemes and control parameters. In spite of the research carried out up to date, there are no general rules on how the control parameters of IA can be selected. In literature [5-8], the choice these parameters is still left to the user to be determined statically prior to the execution of the IA. Mutation rate (p_m) is considered to be one of the most sensitive control parameters that an immune algorithm works with. This is due to the fact that it increases the diversity in population. The choice of mutation rate is essentially a tradeoff between conservatism and exploration [9].

In this paper we investigate an efficient technique for

multi-modal function optimization using IAs. We recommend the use of adaptive probability of mutation to realize the twin goals of maintaining diversity in the population and sustaining the convergence capacity of the IA. In adaptive mutation approach, the value of p_m is varied depending on the fitness values of the solutions. The fitness values of the solution in relation to an objective function to be optimized. During each generation, the fitness of each solution is evaluated, and solutions are selected for reproduction based on their fitness. Selection embodies the principle of survival of the fittest. Good solutions are selected for reproduction while bad solutions are eliminated. The goodness of a solution is determined from its fitness value. The adaptation probability of mutation provides a solution to the problem of choosing the optimal value of the probability of mutation in IA.

The paper is organized as follows. Section 2 describes definition of the optimization problem. The proposed method for fastening the convergence of IA is described in Section 3. Section 4 presents the three experiments that we have conducted to compare the performance of the adaptive and traditional mutation approaches on IA. Section 5 shows the results and discussion of the experiments. Finally, Section 6 offers some conclusions.

2. Problem Definition

The global optimization problem is considered as [10, 11]:

$$\begin{aligned} & \text{minimize } F(x) = y(x) - Y_d(x) \\ & \text{subject to } l \leq x \leq u \end{aligned} \quad (1)$$

where, $F(x)$ is the objective function, $y(x)$ is the optimizing function and $Y_d(x)$ is the desired function. The vector $x = (x_1, x_2, \dots, x_i, \dots, x_N)$ is a variable vector and $l = (l_1, l_2, \dots, l_i, \dots, l_N)$ and $u = (u_1, u_2, \dots, u_i, \dots, u_N)$ define the feasible solution space. Since the IAs mimic the antigen-antibody reaction of the immune system in mammals. The antigen and the antibody in the IA are equivalent to the objective function and the feasible solution for a conventional optimization method, respectively. The antibodies (chromosomes) population is generated either by using binary coding representation or real coding representation. In the binary coding representation, each variable (gene) is encoded as a binary string and the resulting strings are concatenated to form single antibody [12]. However, in the real coding representation, each antibody is encoded as a vector of floating point numbers (*i.e.* $x_i = l_i + \beta(u_i - l_i)$, where $\beta \in [0, 1]$ is a random value), with the same length as the vector of decision variables. This representation is accurate and efficient because it is closest to the real design space, and the string length represents the number of design variables. The hyper-mutation mechanism of the clonal proliferation and mutation mechanism play a critical role in creating diverse antibodies based on minimization of the difference between the optimizing and desired functions. The diversity is measured between the antibodies, and it is increased to prevent local optimization during the optimal search [6].

In optimizing functions, it is important that the IA should be able to converge to the optimal value in as few generations as possible. The convergence speed heavily depends on its main control parameters: population size, replication rate, mutation rate, clonal rate and hyper-mutation rate [13]. Mutation rate is considered to be one of the most sensitive control parameters that an immune algorithm works with, because it increases the diversity in population to prevent the solution to get stuck at a local optimal value during the optimization search. So, the adaptive mutation approach is considered here.

3. Adaptive Mutation Approach

Mutation alters one or more genes (unknown variables) depending on the mutation rate. For a given antibody $X = (x_1, x_2, \dots, x_i, \dots, x_k, \dots, x_N)$, if the gene x_i is selected for mutation depending on mutation rate p_m and the other x_k is randomly selected to join in, the resulting offspring antibody becomes $X' = (x_1, x_2, \dots, x'_i, \dots, x_k, \dots, x_N)$,

where the new gene x'_i is $x'_i = (1 - \beta)x_i + \beta x_k$ and β is selected randomly in the range $[0, 1]$. The choice of p_m is essentially a tradeoff between conservatism and exploration [9]. It has been well established in IA literature [8] that when the value of p_m is selected nearby the range of $[0.02, 0.05]$, the time spent on searching is relatively short and the searching process is stable as well as the search efficiency is better [13]. In the proposed approach, we aim at achieving this trade-off between conservatism and exploration in a different manner, by varying the p_m adaptively in response to the fitness values of the solutions; p_m is increased when the population tends to get stuck at a local optimum and is decreased when the population is scattered in the solution space. So, it is essential to be able to identify whether the IA is converging to an optimum solutions or it diverges. The average of fitness value (f_{av}) of the population in relation to maximum fitness value (f_{max}) of the population is used to detect the conversance of IA. $f_{max} - f_{av}$ is likely to be less for a population that has converged to an optimum solution than that for a population scattered in the solution space. Therefore the difference in the maximum and average fitness values ($f_{max} - f_{av}$) is used as a yardstick for detecting the convergence of IA. The value of p_m is varied depending on the value of $f_{max} - f_{av}$ to prevent premature convergence of the IA to a local optimum. Since p_m has to be increased when the IA converge to a local optimum, p_m will have to be varied inversely with $f_{max} - f_{av}$. The expression that we have chosen for p_m is of the form:

$$p_m = \frac{k_1}{f_{max} - f_{av}}, \quad 0 \leq k_1 \leq 1 \quad (2)$$

From Equation (2), it can be noticed that p_m do not depend on the fitness value of any particular solution, and have the same value for all the solutions of the population. Consequently, solutions with high fitness values as well as solutions with low fitness values are subjected to the same levels of mutation. When a population converges to a globally optimal solution (or even a locally optimal solution), p_m increases and may cause the disruption of the near-optimal solutions. To overcome this problem and to preserve good solutions of the population p_m should have lower value for high fitness solutions and higher value for low fitness solutions. While the high fitness solutions aid in the convergence of the IA, the low fitness solutions prevent the IA from getting stuck at a local optimum. The value of p_m should depend not only on $f_{max} - f_{av}$, but also on the fitness value f of the solution. The closer f is to f_{max} , the smaller p_m should be, *i.e.*, p_m should vary directly as $f_{max} - f$. So, the expression in Equation (2) is modified to become:

$$p_m = k_1 \frac{f_{max} - f}{f_{max} - f_{av}}, \quad 0 \leq k_1 \leq 1 \quad (3)$$

Note that p_m is zero for the solution with the maximum fitness. Also $p_m = k_1$ for a solution with $f = f_{av}$. For solutions with sub-average fitness values (*i.e.* $f < f_{av}$), p_m might assume value larger than 1.0. To prevent the overshooting of p_m beyond 1.0, we also have the following constraints,

$$p_m = k_2, \quad \text{if } f < f_{av} \tag{4}$$

where, $0 \leq k_2 \leq 1$

For convenience, the expression for p_m is given as:

$$p_m = \begin{cases} k_1 (f_{\max} - f) / (f_{\max} - f_{av}), & \text{if } f \geq f_{av} \\ k_2, & \text{if } f < f_{av} \end{cases} \tag{5}$$

The values of k_1 and k_2 are in the range $[0,1]$ and are selected to prevent the IA from getting stuck at a local optimum. For solutions with sub-average fitness the search space is searched for the region containing the global optimum solution. Such solutions need to be completely disrupted, and for this purpose the value of k_2 is selected to be 0.5. But, the value of k_1 is selected by $1/L$ as state in [14], where L is the encoding string length.

4. Experiments

This section presents the three experiments that we have conducted to compare the performance of the adaptive and traditional mutation approaches on IA. For this purpose we have employed three test problems with varying complexities.

Experiment 1: This experiment considers the optimization of the exponential function shown in **Figure 1** and described by the following Equation [13]:

$$y(x) = \sum_{i=0}^9 a_i x^i \tag{6}$$

with the following desired specified values $Y_d(x)$ at $x = [0, 1, 2, 3 \dots, 20]$.

$$Y_d(x) = [0.01, -0.01, -3.83, -4.79, 758.33, 9.0021 \times 10^3, 5.7237 \times 10^4, 5.7237 \times 10^4, 9.2998 \times 10^5, 2.8368 \times 10^6, 7.6281 \times 10^6, 1.8563 \times 10^7, 4.165 \times 10^7, 8.7358 \times 10^7, 1.7309 \times 10^8, 3.2667 \times 10^8, 5.9104 \times 10^8, 1.0306 \times 10^9, 1.7397 \times 10^9, 2.8528 \times 10^9, 4.5587 \times 10^9]$$

Experiment 2: This experiment considers the rapidly varying multimodal function of two variables described by Equation (7). This function is symmetric about the origin with the barrier height between adjacent minima approaching zero as the global optimum is approached and is considered in [15] to test the performance of adaptive genetic algorithm. **Figure 2** illustrates the function $y(x_1, x_2)$ in bi- and uni-dimensional plots for

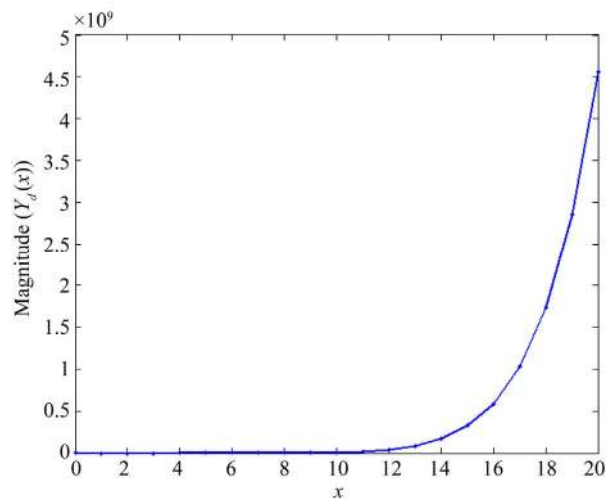


Figure 1 Desired specifications of the function $y(x)$ (experiment 1).

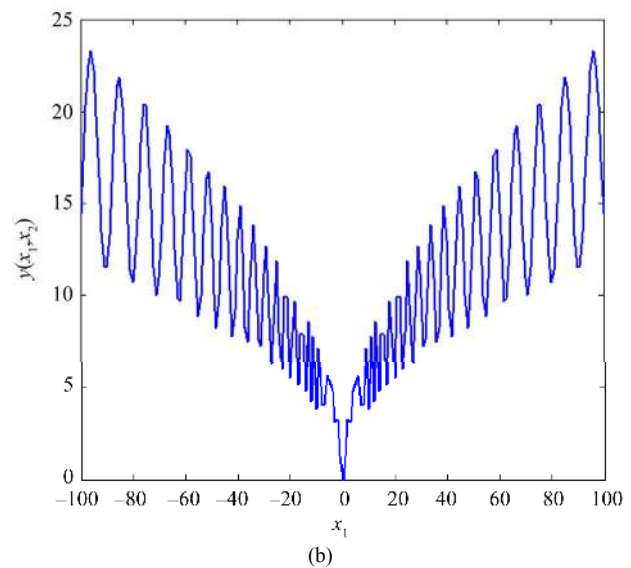
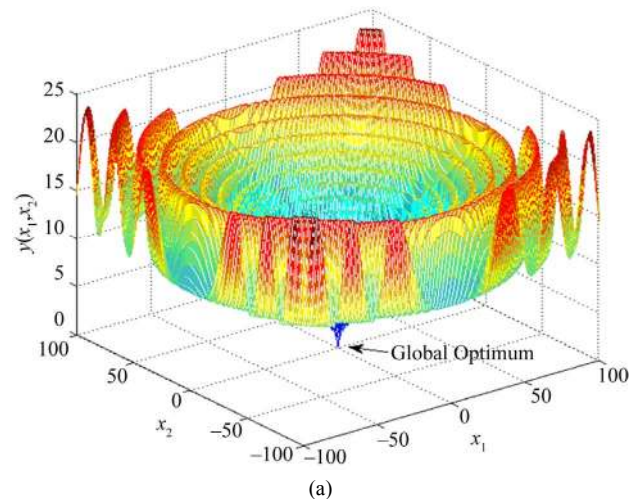


Figure 2. The function $y(x_1, x_2)$ (experiment 2). (a) Bi-dimensional plot; (b) Uni-dimensional plot.

$-100 \leq x_1 \leq 100$ and $-100 \leq x_2 \leq 100$.

$$y(x_1, x_2) = (x_1^2 + x_2^2)^{0.25} \left[\sin^2 \left(50(x_1^2 + x_2^2)^{0.1} \right) + 1 \right] \quad (7)$$

Experiment 3: This experiment considers the design of a second order 2-D narrow-band recursive LPF with magnitude and group delay specifications. The specified magnitude $M_d(\omega_1, \omega_2)$ is shown in **Figure 3** [7,13]. Namely, it is given by Equation (8) with the additional constant group delay $\tau_{d_1} = \tau_{d_2} = 5$ over the passband $\sqrt{\omega_1^2 + \omega_2^2} \leq 0.1\pi$ and the design space is $[-3 \ 3]$. To solve this problem, the frequency samples are taken at $|\omega_i/\pi| = 0, 0.02, 0.04, \dots, 0.2, 0.4, \dots, 1$ in the ranges $-\pi \leq \omega_1 \leq \pi$, and $-\pi \leq \omega_2 \leq \pi$.

$$M_d(\omega_1, \omega_2) = \begin{cases} 1.0, & \text{for } \sqrt{\omega_1^2 + \omega_2^2} \leq 0.08\pi \\ 0.5, & \text{for } 0.08\pi < \sqrt{\omega_1^2 + \omega_2^2} \leq 0.12\pi \\ 0.0, & \text{for } \sqrt{\omega_1^2 + \omega_2^2} > 0.12\pi \end{cases} \quad (8)$$

5. Results and Discussion

For the three experiments [16] considered, the program implementation for each chosen approach is run 30 times. The population size is set to 100 for all experiments and the values of mutation rate for all experiments are given in **Table 1**. **Table 2** gives the average number of generations required by each approach for attaining a solution, the number of instances (out of 30 trials) for which the IA have gotten stuck at a local optimum, and the maximum number of generations. **Figures 4, 5 and 6** show the objective value of the three experiments respectively that obtained using adaptive and traditional muta-

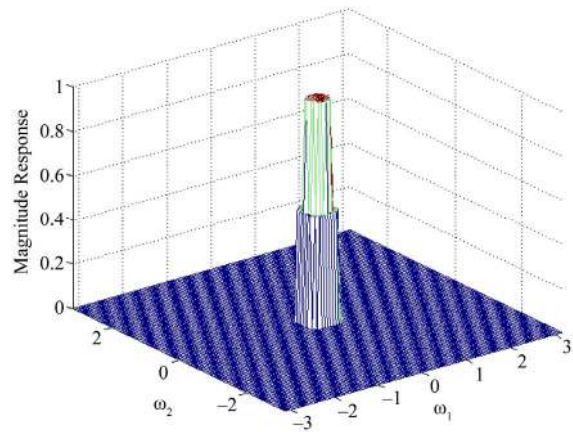


Figure 3. Desired amplitude response $|M_d(\omega_1, \omega_2)|$ of the 2-D narrow-band LPF (experiment 3).

Table 1. The values of mutation rate for all experiments.

Experiments	Unknown variables (L)	Mutation rate	
		Adaptive mutation approach	Traditional mutation approach
Experiment 1	10	0.1	0.1
Experiment 2	2	0.5	0.5
Experiment 3	15	0.07	0.07

Table 2. Comparison between adaptive and traditional mutation approaches for IA.

Experiments	Required generations		Stuck		Maximum number of generations
	Adaptive mutation	Traditional mutation	Adaptive mutation	Traditional mutation	
Experiment 1	102.31	146.23	1	7	200
Experiment 2	57.33	65.43	0	8	100
Experiment 3	101.56	116.53	1	5	250

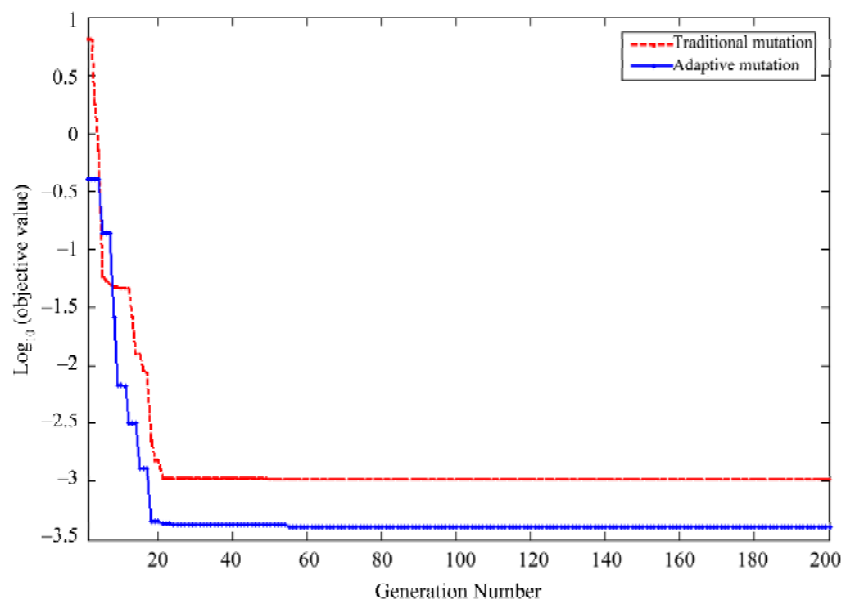


Figure 4. The objective values of experiment 1.

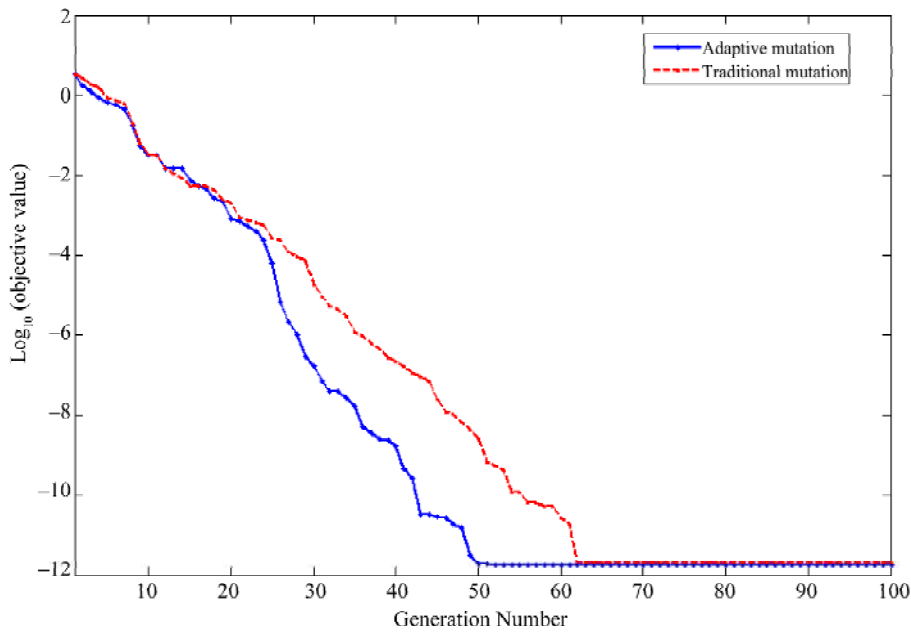


Figure 5. The objective values of experiment 2.

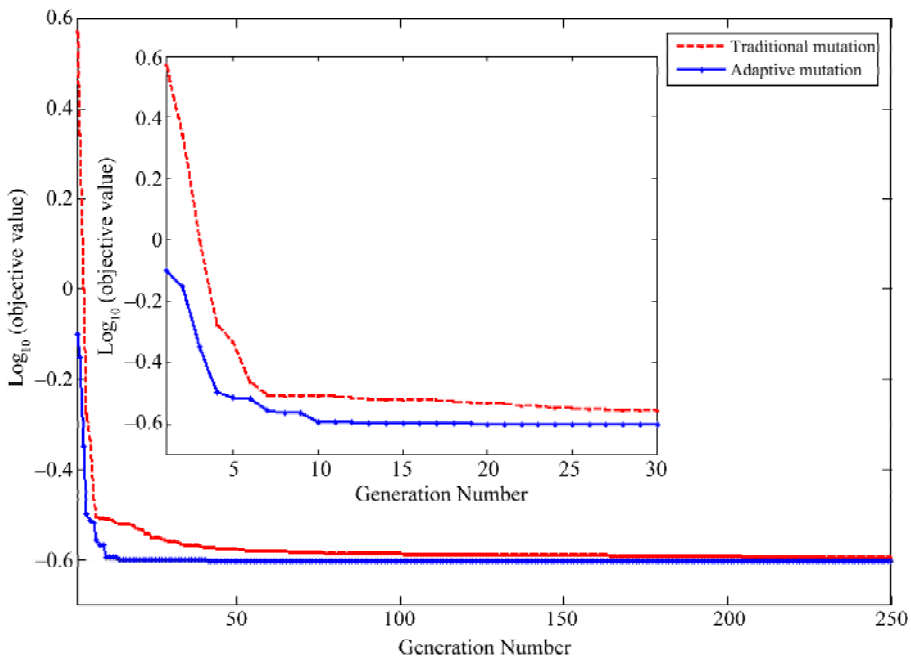


Figure 6. The objective values of experiment 3.

tion approaches. From **Table 2** and **Figures 4, 5** and **6** can be noticed that the adaptive mutation approach improves the performance of the IA and give better results than traditional mutation approach.

6. Conclusion

In this paper, we adopt an efficient adaptive mutation approach to provide a solution to the problem of finding

the optimal value of p_m for immune algorithms. In this approach, the value of p_m is varied depending on the fitness values of the solutions. The proposed approach decreases the value of p_m for high fitness solutions to sustain the convergence capacity of the IA and increases the value of p_m for low fitness solutions to maintain diversity in the population. This method not only improves the convergence of IA, but also prevents the IA from getting stuck at a local optimum. Simulation results show that

the proposed mutation approach efficiently improves IA's performance.

REFERENCES

- [1] F. Vafaei and P. C. Nelson, "A Genetic Algorithm That Incorporates an Adaptive Mutation Based on an Evolutionary Model," *International Conference on Machine Learning and Applications*, Miami Beach, 13-15 December 2009, pp. 101-107. [doi:10.1109/ICMLA.2009.101](https://doi.org/10.1109/ICMLA.2009.101)
- [2] W. Zhang and Y. Liu, "Reactive Power Optimization Based on PSO in a practical Power System," *Proceedings of the International Multi-Conference of Engineers and Computer Scientists*, Vol. 2, 2008.
- [3] M. Abo-Zahhad, S. M. Ahmed, N. Sabor and A. F. Al-Ajlouni, "Design of Two-Dimensional Recursive Digital Filters with Specified Magnitude and Group Delay Characteristics Using Taguchi-based Immune Algorithm," *International Journal of Signal and Imaging Systems Engineering*, Vol. 3, No. 3, 2010, pp. 222-235. [doi:10.1504/IJSISE.2010.038018](https://doi.org/10.1504/IJSISE.2010.038018)
- [4] V. Cutello, G. Nicosia, M. Romeo and P. S. Oliveto, "On the Convergence of Immune Algorithm," *IEEE Symposium on Foundations of Computational Intelligence*, Honolulu, 1-5 April 2007, pp. 409-415.
- [5] J. T. Tsai, W. H. Ho, T. K. Liu and J. H. Chou, "Improved Immune Algorithm for Global Numerical Optimization and Job-Shop Scheduling Problems," *Applied Mathematics and Computation*, Vol. 194, No. 2, 2007, pp. 406-424. [doi:10.1016/j.amc.2007.04.038](https://doi.org/10.1016/j.amc.2007.04.038)
- [6] J. T. Tsai and J. H. Chou, "Design of Optimal Digital IIR Filters by Using an Improved Immune Algorithm," *IEEE Transactions on Signal Processing*, Vol. 54, No. 12, 2006, pp. 4582-4596. [doi:10.1109/TSP.2006.881248](https://doi.org/10.1109/TSP.2006.881248)
- [7] J. T. Tsai, W. H. Ho and J. H. Chou, "Design of Two-Dimensional Recursive Filters by Using Taguchi Immune Algorithm," *IET Signal Process*, Vol. 2, No. 2, 2008, pp. 110-117.
- [8] G. Zilong, W. Sun'an and Z. Jian, "A Novel Immune Evolutionary Algorithm Incorporating Chaos Optimization," *Pattern Recognition Letter*, Vol. 27, No. 1, 2006, pp. 2-8. [doi:10.1016/j.patrec.2005.06.014](https://doi.org/10.1016/j.patrec.2005.06.014)
- [9] R. Myers and E. R. Hancock, "Genetic Algorithm Parameter Sets for Line Labelling," *Pattern Recognition Letters*, Vol. 1827, No. 11, 1997, pp. 1363-1371. [doi:10.1016/S0167-8655\(97\)00111-6](https://doi.org/10.1016/S0167-8655(97)00111-6)
- [10] R. Danchick, "Accurate Numerical Partials with Applications to Optimization," *Applied Mathematics and Computation*, Vol. 183, No. 1, 2006, pp. 551-558. [doi:10.1016/j.amc.2006.05.083](https://doi.org/10.1016/j.amc.2006.05.083)
- [11] J. T. Tsai, T. K. Liu and J. H. Chou, "Hybrid Taguchi-Genetic Algorithm for Global Numerical Optimization," *IEEE Transactions on Evolutionary Computation*, Vol. 8, No. 4, 2004, pp. 365-377. [doi:10.1109/TEVC.2004.826895](https://doi.org/10.1109/TEVC.2004.826895)
- [12] Z. Michalewicz, "Genetic Algorithm and Data Structure," 3rd Edition, Springer-Verlag, Berlin, 1996.
- [13] M. Abo-Zahhad, S. M. Ahmed, N. Sabor and A. F. Al-Ajlouni, "The Convergence Speed of Single-And Multi-Objective Immune Algorithm Based Optimization Problems," *Signal Processing: An International Journal (SPIJ)*, Vol. 4, No. 5, 2010, pp. 247-266.
- [14] G. Ochoa, "Setting the Mutation Rate: Scope and Limitations of the 1/L Heuristic," *Proceedings of the Genetic and Evolutionary Computation Conference*, Morgan Kaufmann, 9-13 July 2002, pp. 315-322.
- [15] M. Srinivas and L. M. Patnaik, "Adaptive Probabilities of Crossover and Mutation in Genetic Algorithms," *IEEE Transactions on Systems*, Vol. 24, No. 4, 1994, pp. 656-667.
- [16] M. Abo-Zahhad, S. M. Ahmed, N. Sabor and A. F. Al-Ajlouni, "Digital Filters Design Educational Software Based on Immune, Genetic and Quasi-Newton Line Search Algorithms," *International Journal of Innovation and Learning*, Vol. 9, No. 1, 2011, pp. 35-62. [doi:10.1504/IJIL.2011.037191](https://doi.org/10.1504/IJIL.2011.037191)