Hindawi Publishing Corporation
Journal of Control Science and Engineering
Volume 2013, Article ID 462706, 5 pages
http://dx.doi.org/10.1155/2013/462706

Research Article

An Improved Differential Evolution Algorithm Based on Adaptive Parameter

Zhehuang Huang[1,2] and Yidong Chen[2,3]

[1] *School of Mathematical Sciences, Huaqiao University, Quanzhou 362021, China*
[2] *Cognitive Science Department, Xiamen University, Xiamen 361005, China*
[3] *Fujian Key Laboratory of the Brain-Like Intelligent Systems, Xiamen 361005, China*

Correspondence should be addressed to Yidong Chen; ydchen_xm@163.com

Received 2 July 2013; Revised 19 August 2013; Accepted 20 August 2013

Academic Editor: Xiaomei Qi

The differential evolution (DE) algorithm is a heuristic global optimization technique based on population which is easy to understand, simple to implement, reliable, and fast. The evolutionary parameters directly influence the performance of differential evolution algorithm. The adjustment of control parameters is a global behavior and has no general research theory to control the parameters in the evolution process at present. In this paper, we propose an adaptive parameter adjustment method which can dynamically adjust control parameters according to the evolution stage. The experiments on high dimensional function optimization showed that the improved algorithm has more powerful global exploration ability and faster convergence speed.

1. Introduction

In recent years, intelligent optimization algorithms [1] are considered as practical tools for nonlinear optimization problems. Differential evolution algorithm [2, 3] is a novel evolutionary algorithm on the basis of genetic algorithms first introduced by Storn and Price in 1997. The algorithm is a bionic intelligent algorithm by simulation of natural biological evolution mechanism. Its main idea is to generate a temporary individual based on individual differences within populations and then randomly restructure population evolutionary. The algorithm has better global convergence and robustness, very suitable for solving a variety of numerical optimization problems, quickly making the algorithm a hot topic in the current optimization field.

Because it is simple in principle and robust, DE has been applied successfully to all kinds of optimization problems such as constrained global optimization [4], image classification [5], neural network [6], linear array [7], monopoles antenna [8], images segmentation [9], and other areas [10–14].

However, DE algorithm can easily fall into local optimal solution in the course of the treatment of the multipeak and the large search space function optimization problems. In order to improve the optimization performance of the DE, many scholars have proposed many control parameters methods [15, 16]. Although all the methods can improve the standard DE performance to some extent, they still cannot get satisfactory results for some of the functions. In this paper, we propose an adaptive parameter adjustment method according to the evolution stage.

This paper is organized as follows. Related work is described in Section 2. In Section 3 the background of DE is presented. The improved algorithm is presented in Section 4. In Section 5 some experimental tests, results, and conclusions are given. Section 6 concludes the paper.

2. Related Work

The DE algorithm has a few parameters. These parameters have a great impact on the performance of the algorithm, such as the quality of the optimal value and convergence rate. There is still no good way to determine the parameters. In order to deal with this problem, researchers have made some attempts. Gamperle et al. [17] reported that it is more difficult than expected to choose the control parameters of DE. Liu and Lampinen [18] reported that the performance of DE algorithm is sensitive to the values of the parameters. Different test functions should have different parameter settings.

At present, the parameter settings are mainly three ways:

(1) determined parameter setting method: the method is mainly set by experience, for example, keeping fixed value throughout the entire evolutionary process;

(2) adaptive parameter setting: some heuristic rules are used to modify the parameter values accordingly to the current state;

(3) self-adaptive parameter setting: the idea that "evolution of the evolution" is used to implement the self-adaptive parameter setting.

Liu and Lampinen [19] proposed a fuzzy adaptive parameter setting method which can change the parameters dynamically. The experiment shows the convergence much faster than the traditional DE algorithm when adapting F and CR. In [20], self-adapting control parameters in DE are proposed; the results show the improved algorithm is better than, or at least comparable to, the standard DE algorithm.

3. Introduction to DE

Compared to other evolutionary algorithms, DE reserves population-based global search strategy and uses a simple mutation operation of the differential and one-on-one competition, so it can reduce the genetic complexity of the operation. At the same time, the specific memory capacity of DE enables it to dynamically track the current search to adjust their search strategy with a strong global convergence and robustness. So it is suitable for solving some of the complex environments of the optimization problem. Basic operations such as selection, crossover, and mutation are the basis of the difference algorithm.

In an iterative process, the population of each generation G contains N individuals. Suppose that the individual i of generation G is represented as

$$X_{iG} = \left(x_{iG}^{1}, x_{iG}^{2}, \ldots, x_{iG}^{D}\right), \quad i = 1, 2, \ldots. \tag{1}$$

3.1. Mutation Operation. An individual can be generated by the following formula:

$$X_{r_1,G+1} = X_{r_1,G+1} + F * \left(X_{r_2,G} - X_{r_3,G}\right). \tag{2}$$

Here r_1, r_2, and r_3 are random numbers generated within the interval $[1, N]$ and variation factor F is a real number of the interval $[0, 2]$; it controls the amplification degree of the differential variable $X_{r_2,G} - X_{r_3,G}$.

3.2. Crossover Operation. In difference algorithm, the cross-operation is introduced to the diversity of the new population. According to the crossover strategy, the old and new individual exchange part of the code to form a new individual. New individuals can be represented as follow:

$$X_{i,G+1} = \left(x_{1i,G+1}, x_{2i,G+1}, \ldots, x_{Di,G+1}\right), \quad i = 1, 2, \ldots, \tag{3}$$

where

$$x_{ji,G+1} = \begin{cases} V_{ji,G+1}, \\ \qquad \text{if}\left(\text{rand}b\left(j\right) \leq \text{CR}\right) \ \text{or} \ \left(j = mbr\left(i\right)\right), \\ V_{ji,G+1}, \\ \qquad \text{if}\left(\text{rand}b\left(j\right) > \text{CR}\right) \ \text{and} \ \left(j \neq mbr\left(i\right)\right) \end{cases} \quad \left(j = 1, 2, \ldots, D\right), \tag{4}$$

where rand $b(j)$ is uniformly distributed in the interval $[0, 1]$ and CR is crossover probability in the interval $[0, 1]$. $mbr\ (i)$ means a random integer between $[0, D]$.

3.3. Selection Operation. Selection operation is greedy strategy; the candidate individual is generated from mutation and crossover operation competition with target individual:

$$x_{i,G+1} = \begin{cases} U_{i,G}, & \text{if } \left(f\left(U_{i,G}\right) > f\left(x_{i,G+1}\right)\right), \\ x_{i,G+1}, & \text{if } \left(f\left(U_{i,G}\right) \leq f\left(x_{i,G+1}\right)\right), \end{cases} \tag{5}$$

where f is the fitness function.

The basic differential evolution (DE) algorithm is shown as Algorithm 1.

Algorithm 1 (the differential evolution algorithm). (1) Initialize the number of population NP, the maximum number of evolution Maxinter, the scale factor and cross-factor.

(2) Initialize the population pop.

(3) Follow the DE/rand/1/bin policy enforcement options, and produce a new generation of individual:

(a) mutation operation;

(b) crossover operation;

(c) selection operation.

(4) Until the termination criterion is met.

The flow chart of differential evolution algorithm is shown in Figure 1.

4. The Adaptive Control Parameter Adjustment Method (ADE)

From standard DE algorithm, it is known that scale factor F and cross-factor CR will not only affect convergence speed of the algorithm, but may also lead to the occurrence of premature phenomenon. In this paper, we propose an adaptive adjustment method according to the evolution stage.

We use a sine function (1/4 cycle) with value of $(-1, 0)$ and a cosine function (1/4 cycle) with value of $(0, 1)$. The image of the two functions shows slower change at the beginning and in the end, with rapid changes and gradual increase in the middle. It is very suitable for setting F value and CR value.

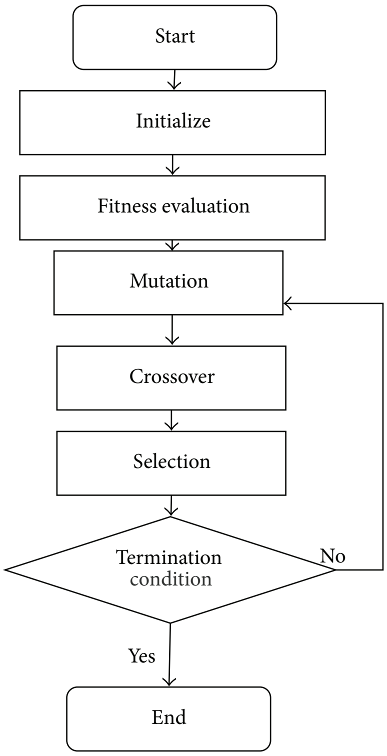


Figure 1: Flow chart of difference evaluation algorithm.

Table 1: Functions used to test the effects of ADE.

Function	Function expression
Sphere function	$f_1(x) = \sum_{t=1}^{n} x_t^2$
Rastrigin function	$f_2(x) = \sum_{t=1}^{n} (x_t^2 - 10\cos(2\pi x_t) + 10)$
Griewank function	$f_3(x) = \frac{1}{4000}\sum_{t=1}^{n}(x_t - 100)^2 - \prod_{t=1}^{n}\cos\left(\frac{x_t - 100}{\sqrt{t}}\right) + 1$
Ackley function	$f_4(x) = 20 + e - 20e^{-0.2\sqrt{(\sum_{t=1}^{n} x_t{}^2)/n}} - e^{(\sum_{t=1}^{n}\cos(2\pi x_t))/n}$
Shaffer's function	$f_5(x) = 0.5 + \frac{\left(\sin\sqrt{x_1^2 + x_2^2}\right)^2 - 0.5}{\left(1 + 0.001\left(x_1^2 + x_2^2\right)\right)^2}$

Table 2: The performances of DE and ADE.

Function	DE		ADE	
	Optimal	Time (s)	Optimal	Time (s)
Sphere	0.039	0.42	0.019	0.31
Rastrigin	19.07	0.43	4.75	0.36
Griewank	0.43	0.45	0.33	0.28
Ackey	1.16	0.45	1.31	0.27
Shaffer	0.00973	0.43	0.00973	0.35

The early stage and the late stage of scale factor F and cross-factor CR are relatively small, with relatively fast increase in the middle, just to meet the global search of PE

$$F = \begin{cases} \alpha + (1-\alpha) \times \sin\left(\frac{\pi t}{\text{MAXITER}} - \frac{\pi}{2}\right), \\ \qquad \text{if}\left(t \leq \frac{\text{MAXITER}}{2}\right), \\ \alpha - (1-\alpha) \times \cos\left(\frac{\pi}{2} - \frac{\pi t}{\text{MAXITER}}\right), \\ \qquad \text{otherwise}, \end{cases} \tag{6}$$

$$\text{CR} = \begin{cases} \beta + (1-\beta) \times \sin\left(\frac{\pi t}{\text{MAXITER}} - \frac{\pi}{2}\right), \\ \qquad \text{if}\left(t \leq \frac{\text{MAXITER}}{2}\right), \\ \beta - (1-\beta) \times \cos\left(\frac{\pi}{2} - \frac{\pi t}{\text{MAXITER}}\right), \\ \qquad \text{otherwise}, \end{cases} \tag{7}$$

where α and β are constants; for example, we can set $\alpha = 0.8$, and $\beta = 0.75$ in the experiment. MAXITER is the maximum number of iterations, and t is the current number of iterations.

The procedure for implementing the APE is given by the following steps.

Algorithm 2 (the improved differential evolution algorithm). (1) Initialize the number of population NP, the maximum number of evolution Maxinter, scale factor F and cross-factor CR.

(2) Initialize the population pop.

(3) Update the scaling factor F of each individual according to the above formula (6).

(4) Update the cross-factor CR of each individual according to the above formula (7).

(5) Perform the following behavior: Mutation, Crossover and Selection, and produce a new generation of individuals.

(6) Until the termination criterion is met.

5. Experimental Results

A set of unconstrained real-valued benchmark functions shown in Table 1 was used to investigate the effect of the improved algorithm.

The results are shown in Table 2. Each point is made from average values of over 10 repetitions. We set scale factor $F = 0.6$ and cross-factor $\text{CR} = 0.5$ for the standard PE algorithm and dynamically adjust F and CR according to the evolution stage for the ADE algorithm.

From Table 2, we can see that no algorithm performs better than the others for all five functions, but on average, the ADE is better than DE algorithm.

For Sphere function, Rastrigin function, and Griewank function, ADE algorithm can effectively improve the accuracy such that the optimal value obtained is much closer to the theoretical one compared with the standard DE algorithm. Ackley function is a multimodal function; from the results of iteration, the accuracy of the improved algorithm is not as that of good as the standard DE algorithm,

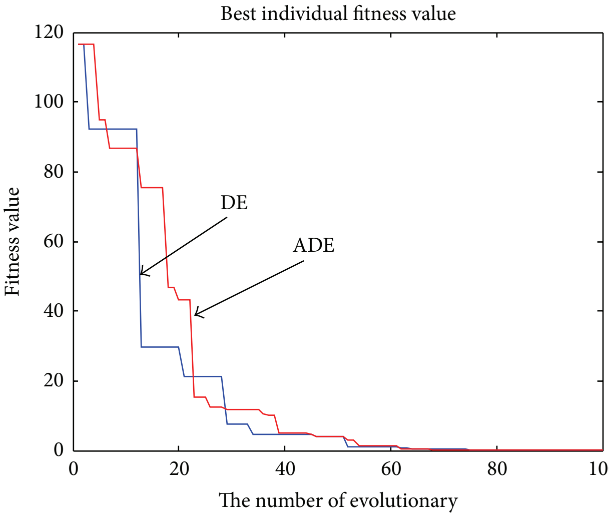


Figure 2: Sphere function.

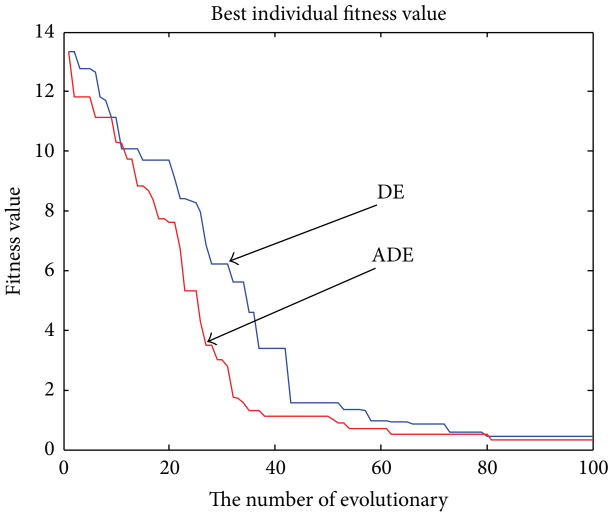


Figure 4: Griewank function.

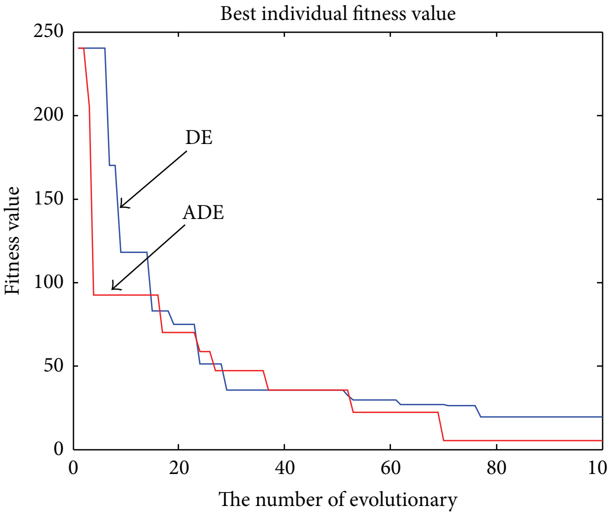


Figure 3: Rastrigin function.

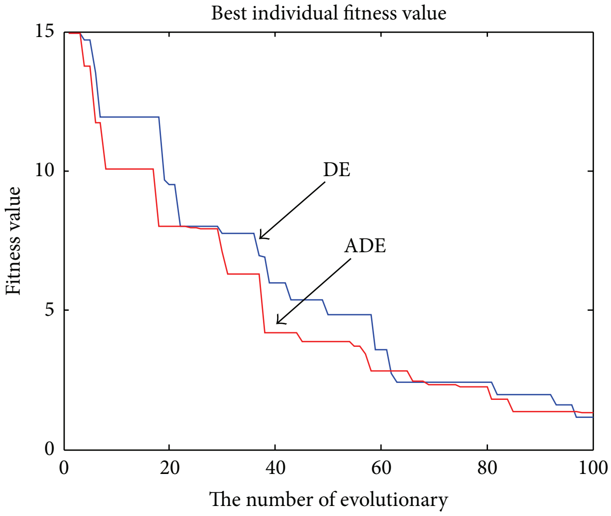


Figure 5: Ackley function.

but the difference is small and acceptable. For Shaffer function, there is no obvious superior algorithm.

For all the five functions, there is a significant improvement as expected on the convergence time. These experimental results show that improving the algorithm can effectively improve the convergence speed with excellent convergence effect.

The comparison of two methods with convergent curves is shown in Figures 2, 3, 4, 5, and 6. The experiment results show the ADE algorithm has better result. Compared with DE, the ADE algorithm has both global search ability and fast convergence speed.

6. Conclusion

The scale factor F and cross-factor CR have a great impact on the performance of the algorithm, such as the quality of the optimal value and convergence rate. There is still no

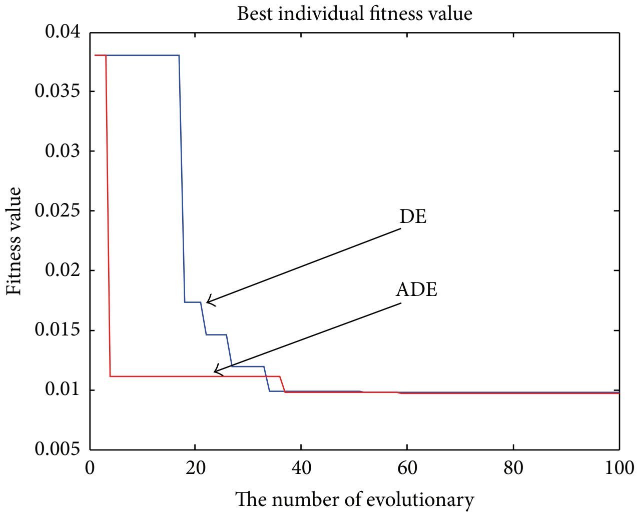


Figure 6: Shaffer's function.

good way to determine the parameters. In this paper, we propose an adaptive parameter adjustment method according to the evolution stage. From before mentioned experiment, we can know the improved algorithm has more powerful global exploration ability and faster convergence speed and can be widely used in other optimization tasks.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant no. 61005052), the Fundamental Research Funds for the Central Universities (Grant no. 2010121068), and the Science and Technology Project of Quanzhou (Grant no. 2012Z91).

References

[1] M. Clerc and J. Kennedy, "The particle swarm-explosion, stability, and convergence in a multi-dimensional complex space," *IEEE Transactions on Evolutionary Computation*, vol. 6, pp. 58–73, 2002.

[2] R. Storn and K. Price, "Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces," *Journal of Global Optimization*, vol. 11, no. 4, pp. 341–359, 1997.

[3] R. Storn and K. Price, "Differential evolution for multi-objective optimization," *Evolutionary Computation*, vol. 4, pp. 8–12, 2003.

[4] H. K. Kim, J. K. Chong, K. Y. Park, and D. A. Lowther, "Differential evolution strategy for constrained global optimization and application to practical engineering problems," *IEEE Transactions on Magnetics*, vol. 43, no. 4, pp. 1565–1568, 2007.

[5] M. G. H. Omran and A. P. Engelbrecht, "Self-adaptive differential evolution methods for unsupervised image classification," in *Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems*, pp. 1–6, Bangkok, Thailand, June 2006.

[6] H. Dhahri and A. M. Alimi, "The modified differential evolution and the RBF (MDE-RBF) neural network for time series prediction," in *Proceedings of the International Joint Conference on Neural Networks 2006 (IJCNN '06)*, pp. 2938–2943, Vancouver, Canada, July 2006.

[7] S. Yang, Y. B. Gan, and A. Qing, "Sideband suppression in time-modulated linear arrays by the differential evolution algorithm," *IEEE Transactions on Antennas and Propagations Letters*, vol. 1, no. 1, pp. 173–175, 2002.

[8] A. Massa, M. Pastorino, and A. Randazzo, "Optimization of the directivity of a monopulse antenna with a subarray weighting by a hybrid differential evolution method," *IEEE Transactions on Antennas and Propagations Letters*, vol. 5, no. 1, pp. 155–158, 2006.

[9] V. Aslantas and M. Tunckanat, "Differential evolution algorithm for segmentation of wound images," in *Proceedings of the IEEE International Symposium on Intelligent Signal Processing (WISP '07)*, Alcalá de Henares, Spain, October 2007.

[10] L. H. Wu, Y. N. Wang, X. F. Yuan, and S. W. Zhou, "Differential evolution algorithm with adaptive second mutation," *Chinese Journal of Control and Decision*, vol. 21, no. 8, pp. 898–902, 2006.

[11] C. T. Su and C. S. Lee, "Network reconfiguration of distribution systems using improved mixed-integer hybrid differential evolution," *IEEE Transactions on Power Delivery*, vol. 18, no. 3, pp. 1022–1027, 2003.

[12] M. F. Tasgetiren, P. N. Suganthan, T. J. Chua, and A. Al-Hajri, "Differential evolution algorithms for the generalized assignment problem," in *Proceedings of the IEEE Congress on Evolutionary Computation (CEC '09)*, pp. 2606–2613, Trondheim, Norway, May 2009.

[13] W. G. Zhang, H. P. Chen, D. Lu, and H. Shao, "A novel differential evolution algorithm for a single batch-processing machine with non-identical job sizes," in *Proceedings of the 4th International Conference on Natural Computation (ICNC '08)*, pp. 447–451, Jinan, China, October 2008.

[14] T. Sum-Im, G. A. Taylor, M. R. Irvings, and Y. H. Song, "A differential evolution algorithm for multistage transmission expansion planning," in *Proceedings of the 42nd International Universities Power Engineering Conference (UPEC '07)*, pp. 357–364, Brighton, UK, September 2007.

[15] Z. F. Wu, H. K. Huang, B. Yang, and Y. Zhang, "A modified differential evolution algorithm with self-adaptive control parameters," in *Proceedings of the 3rd International Conference on Intelligent System and Knowledge Engineering (ISKE '08)*, pp. 524–527, Xiamen, China, November 2008.

[16] J. Liu and J. Lampinen, "A fuzzy adaptive differential evolution algorithm," *Soft Computing*, vol. 9, no. 6, pp. 448–462, 2005.

[17] R. Gamperle, S. D. Muller, and P. Koumoutsakos, "A parameter study for differential evolution," in *Proceedings of the International Conference on Advances in Intelligent Systems, Fuzzy Systems, Evolutionary Computation (WSEAS '02)*, pp. 11–15, Interlaken, Switzerland, February 2002.

[18] J. Liu and J. Lampinen, "On setting the control parameter of the differential evolution method," in *Proceedings of the 8th Internatuional Conference on Soft Computing (MENDEL '02)*, pp. 11–18, Brno, Czech Republic, June 2002.

[19] J. Liu and J. Lampinen, "A fuzzy adaptive differential evolution algorithm," *Soft Computing*, vol. 9, no. 6, pp. 448–462, 2005.

[20] J. Brest, S. Greiner, B. Bošković, M. Mernik, and V. Zumer, "Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems," *IEEE Transactions on Evolutionary Computation*, vol. 10, no. 6, pp. 646–657, 2006.

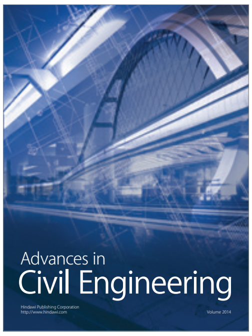
Journal of
Engineering

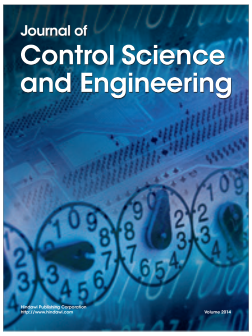
The Scientific World Journal

International Journal of
Rotating Machinery

Journal of
Sensors

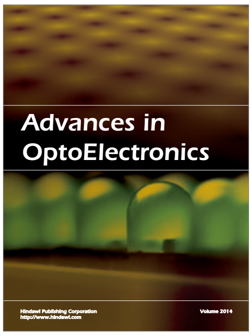
International Journal of
Distributed Sensor Networks


Advances in
Civil Engineering


Journal of
Control Science and Engineering

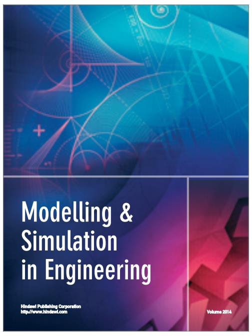
Journal of
Robotics

Journal of
Electrical and Computer Engineering

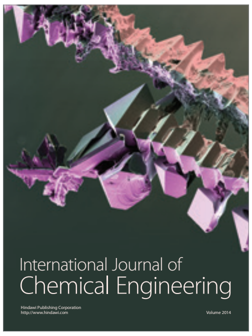

Advances in
OptoElectronics

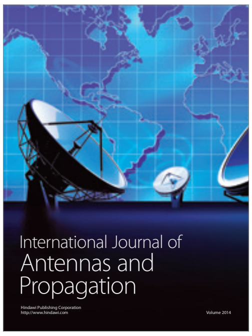
VLSI Design

International Journal of
Navigation and Observation

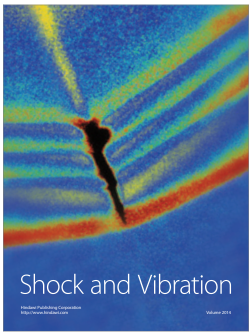

Modelling & Simulation in Engineering

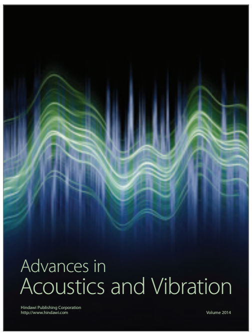
International Journal of
Aerospace Engineering


International Journal of
Chemical Engineering


International Journal of
Antennas and Propagation

Active and Passive Electronic Components


Shock and Vibration


Advances in
Acoustics and Vibration