Power-Efficient Immune Clonal Optimization and Dynamic Load Balancing for Low Energy Consumption and High Efficiency in Green Cloud Computing

Zhuqian Long and Wentian Ji
Hainan College of Software Technology, Qionghai 571400, China
Email: www.abrwhom@gmail.com; chowon.yhw@whu.edu.cn

Abstract —The energy consumption is considered as key factors of green cloud computing to achieve resource allocation. To address the issue of high energy consumption and low efficiency of cloud computing, this paper proposes a power-efficient immune clonal optimization algorithm (PEICO) based on dynamic load balancing strategy and immune clonal selection theory in green cloud computing. The experimental results show that PEICO performs much better than the clonal selection algorithms and differential evolution in terms of the quality of solution and computational cost.

Index Terms—Green cloud computing, global optimization, energy-efficient task scheduling, power-efficient immune clonal optimization algorithm, immune clonal algorithm

I. INTRODUCTION

With the rapid development of cloud computing, the servers' scale of cloud data center is constantly expanding every year, which causes huge power consumption [1]. Furthermore unreasonable scheduling policies lead to energy waste, making the cloud data center operating costs continually increase. Energy efficiency has become prominent contradiction in green cloud computing. Cloud computing is the Internet as the carrier to provide infrastructure, platform, software services by virtualization technology. Cloud computing makes it easier to trade and data storage, and less time consuming task for the end user. Green cloud computing is composed of a series of interconnected and virtualization computers, the virtualization of computer dynamically provides one or more unified computing and storage resources. Therefore, the use of green data center is a typical application of green communication in green cloud computing.

Green cloud computing can not only improve the rate of cloud computing infrastructure, but also can minimize the energy consumption [2]. Due to the heterogeneity of the resource nodes in cloud computing environment, the load between the nodes is imbalance. The load of cloud computing resource nodes is imbalance, which is easy to cause the communication delay between the nodes and the more energy consumption during the process of task scheduling. Load balancing affects the utilization of resource nodes, and the energy consumption determines the operating cost of the data center [3]. Therefore, how to design an efficient cooperative task scheduling algorithm in green cloud computing has become an urgent need to solve problem. Fig. 1 shows the green cloud computing system model.

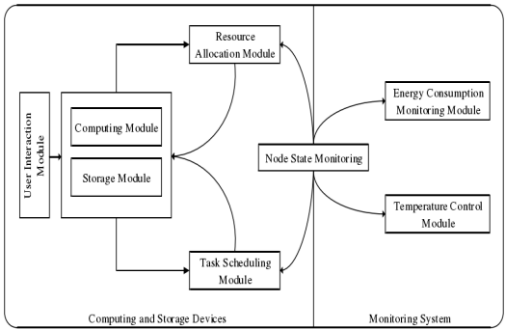

Fig. 1. Green cloud computing system model

Since human society appears in nature, man invented a lot of techniques, methods and tools by simulating the structure, function and behavior of organism in nature, which used to solve practical problems in social life. Many of adaptive optimization phenomena constantly give revelation in nature: organisms and natural ecosystems through their evolution in humans seem to make a lot of highly complex optimization problem has been the perfect solution [4]. In the biological sciences field, people have been carried out extensive and in-depth study on the evolutionary, genetic, immune and other natural phenomena. Although artificial immune algorithm has its own characteristics and advantages, there are some drawbacks in the process of practical application, such as poor stability, data redundancy, and the limited capacity of local search. Differential Evolution (DE) algorithm is a new evolutionary computation technique, which is a stochastic model for simulation biological evolution through iterative so that individuals adapt to the environment is preserved. DE suits for solving some of the use of conventional mathematical programming methods can not solve the complex environment of the optimization problem.

Manuscript received January 13, 2016; revised June 21, 2016.
This work was supported by the Natural Science Foundation of Hainan Province under Grant No. 20156230 and 614242, the Colleges and Universities Science Research Project of Hainan Province under Grant No. Hjkj2013-56.
Corresponding author email: chowon.yhw@whu.edu.cn.
doi:10.12720/jcm.11.6.558-563

The main purposes of this paper include the following three aspects: (1) to reduce the energy consumption of data centers; (2) to make the system resource node load balance; (3) to improve the utilization rate of resource node. Aiming at the problem of energy consumption of data center in green cloud computing, this paper put forward the green cloud computing system structure, and designs a power-efficient immune clonal optimization algorithm for cooperative task scheduling based on dynamic load balancing strategy and immune clonal selection theory in green cloud computing. PEICO has the advantages of Clonal Selection Algorithm (CSA) and DE. Its basic idea is to introduce clonal selection theory to DE, improving the search pattern of algorithm and enhancing the convergence rate of algorithm. It can ensure the ability of global search and local search and enhance the performances of the algorithm.

The main contributions of this paper include:(1) A brief review about the advantages and disadvantages of various existing task scheduling algorithms in green cloud computing are presented. (2) An effective comprehensive energy efficiency model is proposed. (3) A power-efficient immune clonal optimization algorithm for task scheduling is proposed.

The rest of this paper is organized as follows: A brief survey is given in Section 2. The comprehensive energy efficiency model is presented in Section 3. The power-efficient immune clonal optimization algorithm for task scheduling is proposed in Section 4. Section 5 describes simulation and analysis of results, followed by the conclusions in Section 6.

II. Related Works

In this section, we focus our discussion on the prior research on energy consumption and task scheduling algorithms. In recent years, there have been some studies devoted to the new energy-efficient techniques and heuristic intelligent optimization algorithms that are suitable for task scheduling in green cloud computing.

In order to deal with the criterion of makespan minimization for the HFS (Hybrid Flow Shop) scheduling problems, O. Engin proposed a generic systematic procedure which was based on a multi-step experimental design approach for determining the optimum system parameters of AIS [5]. J. C. Chen proposed a hybrid immune multi-objective optimization algorithm (HIMO) based on clonal selection principle [6]. In HIMO, a hybrid mutation operator was proposed with the combination of Gaussian and polynomial mutations. For the problem of indeterminate direction of local search, lacking of efficient regulation mechanism between local search and global search and regenerating new antibodies randomly in the original optimization version of artificial immune network, Q. Z. Xu *et al*. proposed a novel predication based immune network to solve multimodal function optimization more efficiently, accurately and reliably [7]. In order to build a general computational framework by simulating immune response process, M.G. Gong introduced a model for population-based artificial immune systems, termed as PAIS, and applied it to numerical optimization problems [8]. In order to maintain a diverse repertoire of antibodies, K. C. Tan et al. proposed an evolutionary artificial immune system for multi-objective optimization which combines the global search ability of evolutionary algorithms and immune learning of artificial immune systems was proposed [9].

D. X. Zou proposed a Modify Differential Evolution algorithm (MDE) to solve unconstrained optimization problems [10]. Min–max problems were considered difficult to solve, specially constrained min–max optimization problems. A. S. Segundo proposed a novel differential evolution approach consisting of three populations with a scheme of copying individuals for solving constrained min–max problems [11]. To reduce the computational time for high-dimensional problems, Hui Wang et al. presented a parallel differential evolution based on Graphics Processing Units (GPUs) [12].

In order to compare the performance of DE and PSO in solving min-max constrained optimization problems, Mahmud I. studied considers two well-known variants of PSO and DE [13]. X. Z. Gao *et al*. proposed a novel optimization scheme CSA–DE based on the fusion of the clonal selection algorithm and differential evolution [14].

In order to reduce energy consumption, many scholars put forward many effective solutions. Gong L. *et al*. proposed green energy saving strategy for cloud computing platform [15]. From the angle of system resource allocation analysis of how to reduce the amount of energy, Guazzone M. *et al*. proposed a framework of automatic management of cloud infrastructure resources [16]. Jones *et al*. proposed a task scheduling model and algorithm with a bandwidth centric [17], which does not consider the load balancing of system, leading to the task distribution imbalance. Lu Xiaoxia *et al*. introduced ecological difference equation based on the ecological dynamics, dynamic adjustment the number of tasks in resource nodes, and proposed a main task scheduling algorithm based on the establishment of a predator-prey model [18]. Chen Yanpei measured the energy consumption of data center based on the framework of MapReduce and HDFS, and achieved the purpose of efficient utilization of energy through the optimization of system configuration parameters [19]. Through processing the idle node to achieve energy saving, Harnik *et al*. achieved the purpose of saving energy through closing a large number of nodes in idle periods [20]. Y. Kessaci *et al*. presented an energy-aware multi-start local search algorithm (EMLS) [21]. A parallel-machine scheduling involving both task processing and resource allocation was studied by using an Improved Differential Evolution Algorithm (IDEA) [22].

III. Comprehensive Energy Efficiency Model

Load balancing is a kind of effective balance work as load mechanism [23], [24]. According to the computing performance level of resource node, the tasks will be

allocation by load balancing to different resource nodes on execution [25].

At present, there are two ways to reduce energy consumption in green cloud computing: (1) by dynamically adjusting the voltage or frequency of the resource nodes to save energy. (2) Turn off unneeded resource nodes to achieve energy saving.

Definition 1. The number of instructions per unit time that resources can execute called computing performance of resource node.

Definition 2. The computing performance of resource r_i denotes c_i (The unit is MIPS), the total amount of tasks to reach the resource r_i within the unit time t is S_i, load level of resource r_i is given as

$$u_i = \frac{c_i}{S_i} \times 100\% \tag{1}$$

Definition 3. Assume that u_i is the load level of resource r_i, PL_i is the corresponding power consumption value of the resource under different loads u_i. The comprehensive power consumption P_i of the resource r_i is given as

$$P_i = \alpha_1 \times PL_{u_1} + \alpha_2 \times PL_{u_2} + \alpha_3 \times PL_{u_3} + \alpha_4 \times PL_{u_4} \tag{2}$$

In which, α_i is the weight ratio coefficient, $\alpha_1 + \alpha_2 + \alpha_3 + \alpha_4 = 1$. $\alpha_i = 0\%$ indicates that the resource has no traffic flow, but it is in the link state.

Definition 4. Assume that T_i is the sum of resources throughput under the configuration model, P_i is the comprehensive power consumption of the resource r_i in different models, A_i is the application weight in different models. Then, the comprehensive energy efficiency of the system is defined as:

$$CEE = \sum_{i=1}^{n} A_i \times \frac{P_i}{T_i} \tag{3}$$

Definition 5. Assume that $t(m,n)$ is the total task execution time , LB is the system task balance factor. Then, the affinity function $aff(a_i)$ can be described as:

$$aff(a_i) = \frac{LB}{t(m,n)} \tag{4}$$

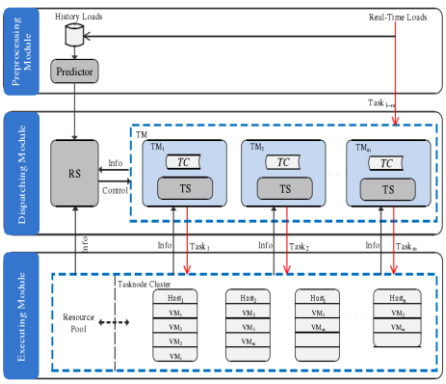

Fig. 2. System resource allocation model

Through the dynamic load balancing strategy, system resource allocation model is shown in Fig. 2.

IV. Power-Efficient Immune Clonal Optimization Algorithm for Task Scheduling

Based on the above work, the main operations of our algorithm for task scheduling can be summarized as follows:

Algorithm 1: Power-Efficient Immune Clonal Optimization Algorithm (PEICO)

Input: population size S, mutation probabilities P_m, maximum evolution generation G_m

Begin

Step 1: Randomly initialize the antibody population $A(0) = \{a_1(0), a_2(0), ..., a_n(0)\} \in I^n$, $k = 0$.

Step 2: Calculate the affinity of initial population $A(k)$ according to objective function.

Step 3: Select half of the antibodies with larger affinity to $A_r(k)$, and denote the other antibodies as $A_l(k)$.

Step 4: Clone $A_r(k)$ to generate the population $B(k)$, and the clonal number is proportional to their affinities.

Step 5: Perform differential crossover for the population $A_r(k)$ to generate the population $C(k)$ as follow:

$$u_{j,i,G+1} = \begin{cases} v_{j,i,G+1} & rand_j \le t \\ x_{j,i,G} & otherwise \end{cases} \tag{5}$$

where t is the crossover constant that takes values based on a random variable $rand_j \in [0,1]$.

Step 6: Perform differential mutation for the population $A_l(k)$ to generate the population $D(k)$ as follow:

$$V_{i,g} = x_{1,g} + \delta(x_{2,g} - x_{3,g}) \tag{6}$$

where δ is a scaling factor which controls amplification of the differential evolution.

Step 7: Compute individual affinity after differential mutation. If the affinity of individual after mutation is larger than the old one, then substitute the old one with it.

Step 8: Obtain the following generation population $A(k+1) = B(k) \cup C(k) \cup D(k)$.

Step 9: $k = k+1$; If ending conditions are satisfied, the PEICO algorithm automatically ends; Otherwise, go to step 2 until the proposed iterations are completed .

End

Output: the individual with minimal objective function value

V. Experimental Results

In this section, in order to verify the validity of the proposed PEICO in this paper, the benchmark functions

and CloudSim are used to as a platform and tool for experimental testing. CloudSim is a function library developed on the discrete event simulation package SimJava. CloudSim component tool is open source, and provides a virtual engine. The PEICO is compared with several typical tasks scheduling algorithms, including HIMO [6] and IDEA [22]. The configuration information of computing nodes is shown in Table I.

TABLE I: THE CONFIGURATION INFORMATION OF COMPUTING NODES

Node Type	CPU	Memory	VM(Virtual Machine) Information
1	Intel Core i5 4200h 3.4G Hz	4GB	1* VM Type 1+1* VM Type 2
2	Intel Core i7 4790k 4G Hz	4GB	2* VM Type 2
3	Intel Core 2 T6400 2G Hz	4GB	1* VM Type 1+1* VM Type 2
4	Intel Core 2 T9600 2.8G Hz	4GB	1* VM Type 1+1* VM Type 2
5	Intel Core i3 4130 3.4G Hz	4GB	2* VM Type 2
6	Intel Core i5 5200u 2.7G Hz	4GB	1* VM Type 1+1* VM Type 2

Two kinds of contrast experiments are carried out in this paper :(1) the optimal solution is compared for the there combinatorial optimization functions. The three combinatorial optimization functions are shown in Table II. (2) The response time and energy consumption are compared in CloudSim platform. The optimal solutions for the three task scheduling algorithms under the conditions of running 100 times are shown in Fig. 3- Fig. 5. Their maximum evolution generation is set to 1000.

TABLE II: THE THREE COMBINATORIAL OPTIMIZATION FUNCTIONS

Functions	Global minimum	Convergence point
$f_1(x)=\sum_{i=1}^{3}x_i^2$ $(-5.12\le x\le 5.12)$	0	$x^*=(0,0,0)$
$f(x)=\frac{1}{4000}\sum x-\prod\cos(\frac{x}{\sqrt{i}})+1$ $(-600\le x\le 600)$	0	$x^*=(0,0,0)$
$f(x)=\sum\sqrt{\lvert x\rvert}+\prod\lvert x\rvert$ $(-10\le x_i\le 10)$	0	$x^*=(0,0,0)$

It can be seen from Fig. 3-Fig. 5 that the whole performance of PEICO is superior to HIMO and IDEA. The simulation results indicate that PEICO can significantly enhance the quality of the solutions obtained, and reduce the time taken to reach the solutions. In particular, simulation results on numerical optimization problems demonstrate that the proposed algorithm achieves an improved success rate and final solution with less computational effort.

Apparently, compared with the original HIMO and IDEA, our PEICO can acquire much better optimization results within the same numbers of iterations. Experimental results demonstrate effectively the PEICO has a superior nonlinear function optimization performance over the IDEA, and the affinity of antibody use differential mutation to achieve the balance between the convergence speed and optimum quality. So it can efficiently search in a diverse set of local optimum to find better solutions.

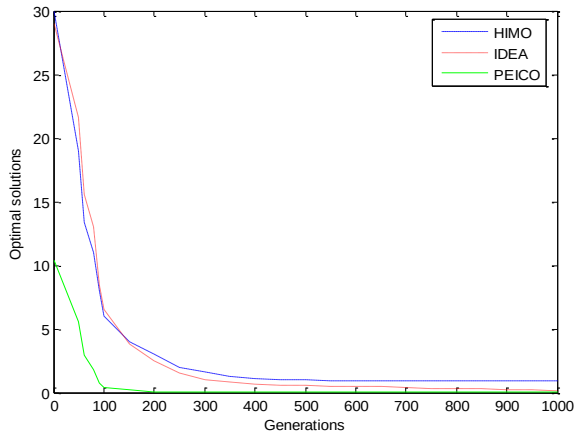

Fig. 3. Comparison of optimal solutions for the three algorithms in $f(x)$

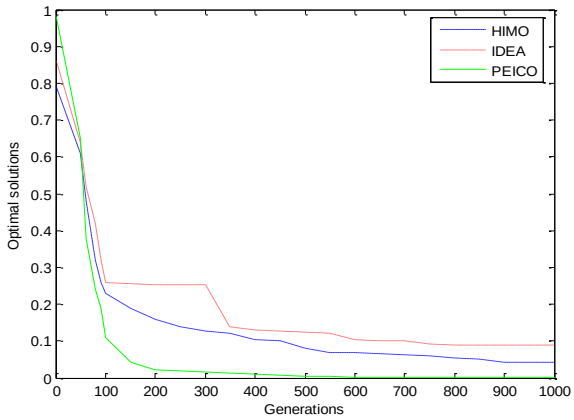

Fig. 4. Comparison of optimal solutions for the three algorithms in $f_2(x)$

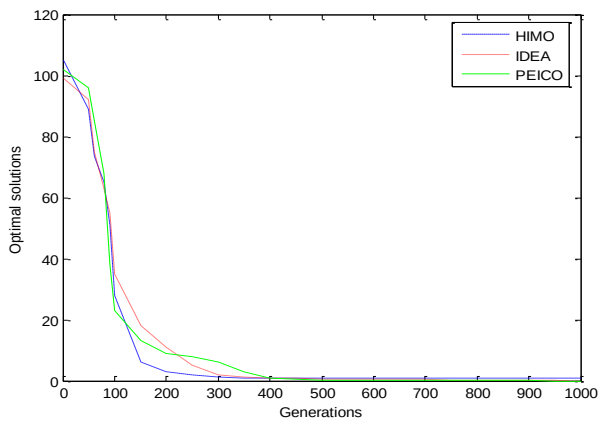

Fig. 5. Comparison of optimal solutions for the three algorithms in $f_3(x)$

We compare the response time of using the same cloud computing environments in Fig. 6. In most cases, the

response time of the PEICO is the least. In particular, PEICO has faster response speed when the number of tasks is significantly increased. PEICO retains global search strategies based on population, uses real-coded and simple differential mutation operation, which reduces the complexity of genetic operation. Meanwhile, PEICO has a unique ability to dynamical track the current search so as to adjust their search strategies, and has a strong global convergence and robustness.

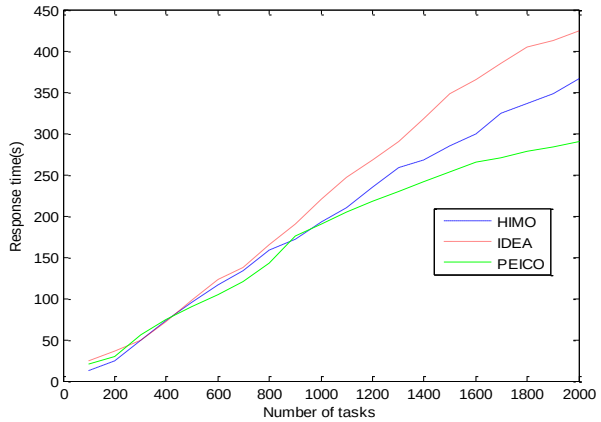

Fig. 6. Comparison of response time for the three algorithms when the number of VM is 100

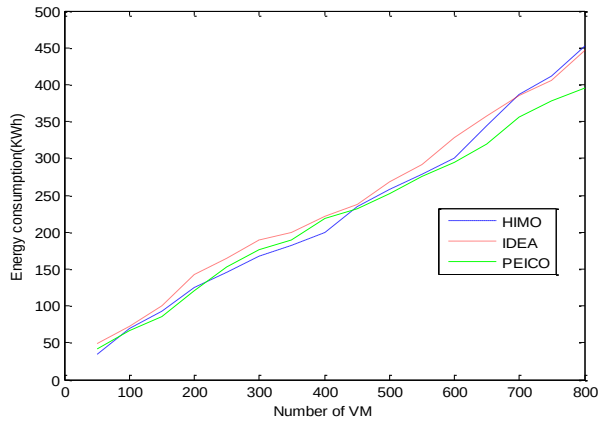

Fig. 7. Comparison of energy consumption for the three algorithms in different number of VM

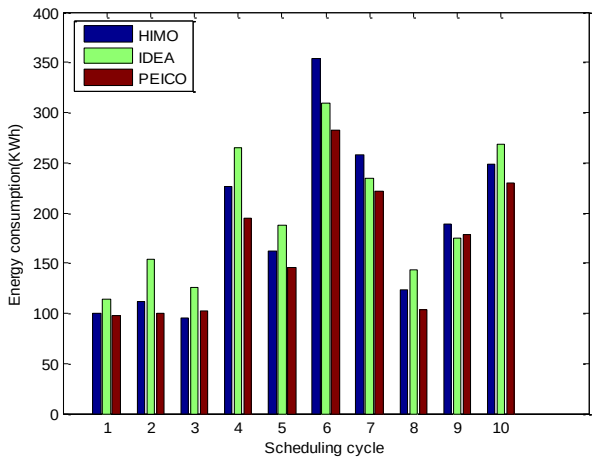

Fig. 8. Comparison of energy consumption for the three algorithms in different scheduling cycles

Fig. 7-Fig. 8 show the energy consumption for the three algorithms in different number of VM and scheduling cycles, respectively. It can be observed that the proposed PEICO uses less energy consumption, compared with HIMO and IDEA in most scheduling cycles. PEICO can effectively balance load of resource node and reduce energy consumption. Through the above experimental results, PEICO can ensure the ability of global search and local search and improve convergence performance. PEICO can enhance the diversity of the population, avoid the premature convergence phenomenon, and have high accuracy of solution.

VI. Conclusions

For the problem of energy consumption in green cloud computing data centers, this paper puts forward the theory of green cloud computing and designs a power-efficient immune clonal optimization algorithm for task scheduling. Based on the green cloud system architecture, the energy is considered as system resources to achieve resource allocation. The results show that the proposed algorithm has the most optimal ability of the reducing energy consumption of data center. Obviously, the proposed PEICO can be applied to any other combinatorial and numerical optimization problem using suitable representations and variable operators.

Acknowledgment

This research work was supported by the Natural Science Foundation of Hainan Province under Grant No. 20156230 and 614242, the Colleges and Universities Science Research Project of Hainan Province under Grant No. Hjkj2013-56. The authors wish to thank the anonymous reviewers who helped to improve the quality of the paper.

References

[1] W. Shu, W. Wang, and Y. Wang, "A novel energy-efficient resource allocation algorithm based on immune clonal optimization for green cloud computing," *EURASIP Journal on Wireless Communications and Networking*, pp. 1-9, April 2014.

[2] C. F. Lai, S. Zeadally, J. Shen, and Y. X. Lai, "A cloud-integrated appliance recognition approach over internet of things, " *Journal of Internet Technology*, vol. 16, no. 7, pp. 1157-1168, Dec. 2015.

[3] Y. Gao, J. Duan, and W. Shu, "A novel ant optimization algorithm for task scheduling and resource allocation in cloud computing environment," *Journal of Internet Technology*, vol. 16, no. 7, pp. 1329-1338, Dec. 2015.

[4] X. Fan and C. Yuan, "An improved lower bound for bayesian network structure learning," in *Proc. 29th AAAI Conference on Artificial Intelligence*, 2015, pp. 2439-2445.

[5] O. Engin and A. Döyen, "A new approach to solve hybrid flow shop scheduling problems by artificial immune system," *Future Generation Computer Systems*, vol. 30, pp. 1083–1095, June 2014.

[6] J. Chen, Q. Lin, and Z. Ji, "A hybrid immune multiobjective optimization algorithm," *European Journal of Operational Research*, vol. 204, pp. 294-302, July 2010.

[7] Q. Xu, L. Wang, and J. Si, "Predication based immune network for multimodal function optimization," *Engineering Applications of Artificial Intelligence*, vol. 23, pp. 495–504, June 2010.

[8] M. Gong, L. Jiao, and X. Zhang, "A population-based artificial immune system for numerical optimization," *Neurocomputing*, vol. 72, pp. 149-161, Nov. 2008.

[9] K. C. Tan, C. K. Goh, A. A. Mamun, and E. Z. Ei, "An evolutionary artificial immune system for multi-objective optimization," *European Journal of Operational Research*, vol. 187, pp. 371–392, Jan. 2008.

[10] D. Zou, J. Wu, L. Gao, and S. Li, "A modified differential evolution algorithm for unconstrained optimization problems," *Neurocomputing*, vol. 120, pp. 469-481, May 2013.

[11] G. A. S. Segundo, R. A. Krohling, and R. C. Cosme, "A differential evolution approach for solving constrained min–max optimization problems," *Expert Systems with Applications*, vol. 39, pp. 13440–13450, May 2012.

[12] H. Wang, S. Rahnamayan, and Z. Wu, "Parallel differential evolution with self-adapting control parameters and generalized opposition-based learning for solving high-dimensional optimization problems," *Journal Parallel Distributed Computing*, vol. 73, pp. 62–73, May 2013.

[13] M. Iwan, R. Akmeliawati, T. Faisal, and H. M. A. Al-Assadi, "Performance comparison of differential evolution and particle swarm optimization in constrained optimization," *Precede Engineering*, vol. 41, pp. 1323–1328, May 2012.

[14] X. Z. Gao, X. Wang, and S. J. Ovaska, "Fusion of clonal selection algorithm and differential evolution method in training cascade–correlation neural network," *Neurocomputing*, vol. 72, pp. 2483–2490, June 2009.

[15] Y. W. Ma, W. T. Cho, J. L. Chen, Y. M. Huang, and R. Zhu, "RFID-based mobility for seamless personal communication system in cloud computing," *Telecommunication Systems*, vol. 58, no. 3, pp. 233-241, Mar. 2015.

[16] M. Guazzone, C. Anglano, and M. Canonico, "Exploiting VM migration for the automated power and performance management of green cloud computing systems," *Energy Efficient Data Centers*, vol. 3, pp. 81-92, June 2012.

[17] X. Liu, R. Zhu, B. Jalaian, and Y. Sun, "Dynamic spectrum access algorithm based on game theory in cognitive radio networks," *Mobile Networks and Applications*, vol. 20, no. 6, pp. 817-827, Dec. 2015.

[18] L. Xiaoxia and Z. Zhong, " Research on Optimization task scheduling algorithm in cloud computing," *Computing Technology and Automation*, vol. 30, no. 4, pp. 108-110, April 2011.

[19] C. Yanpei, L. Keys, and R. H. Katz, "Towards energy efficient map-reduce," Berkeley: EECS Department, University of California, 2009.

[20] D. Harnik, D. Naor, and I. Segall, "Low power mode in cloud storage systems, " in *Proc. 23rd IEEE International Symposium on Parallel and Distributed Processing*, Rome, Italy, 2009, pp. 1-8.

[21] Y. Kessaci, N. Melab, T. El-Ghazali, " A multi-start local search heuristic for an energy efficient VMs assignment on top of the OpenNebula cloud manager," *Future Generation Computer System*, vol. 29, no. 1,pp. 1-20, 2013.

[22] J. T. Tsai, J. C. Fang, and J. H. Chou, "Optimized task scheduling and resource allocation on cloud computing environment using improved differential evolution algorithm," *Computers & Operations Research*, vol. 40, no. 2, pp. 3045–3055, Feb. 2013.

[23] L. Zhang, J. Ma, and Y. Wang, *et al*., "Toward green cloud computing: an attribute clustering based collaborative filtering method for virtual machine migration," *Information Technology Journal*, vol. 12, no. 23, pp. 7275-7279, May 2013.

[24] B. Mondal, K. Dasgupta, and P. Dutta, "Load balancing in cloud computing using stochastic hill climbing-a soft computing approach," *Precede Technology*, vol. 4, no. 5, pp. 783-789, May 2012.

[25] X. Fan, C. Yuan, and B. Malone, "Tightening bounds for Bayesian network structure learning," in *Proc. 28th AAAI Conference on Artificial Intelligence*, 2014, pp. 2439-2445.

Zhuqian Long was born in Hainan Province, China, in 1983. He received the Bachelor's degree from the Central China Normal University, in 2004, and the Master's degree from the Hainan University, in 2013. He is currently an associate professor in the Department of Software Engineering, Hainan College of Software. He has published 8 journal papers and about 4 conference papers. His research interests include cloud computing, green communication, and software engineering.

Wen-Tian Ji was born in Gansu Province, China, in 1979.He received the Bachelor's degree from the LanZhou University, in 2003, and the Master's degree from the Hainan University, in 2012. He is currently an associate professor in the Department of Software Engineering, Hainan College of Software. His research interests include green cloud computing.