Journal of Theoretical and Applied Information Technology
30th November 2013. Vol. 57 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

AN APPROACH TO OPTIMIZE WORKFLOW SCHEDULING FOR CLOUD COMPUTING ENVIRONMENT

[1] P. KUMAR and [2] SHEILA ANAND

[1]Associate Professor, Department of Information Technology, Rajalakshmi Engineering College, CHENNAI

[2]Dean (Research) Computer Studies, Rajalakshmi Engineering College, CHENNAI

E-mail: [1]kumar@rajalakshmi.edu.in

ABSTRACT

Cloud computing is a model for enabling convenient, on-demand network access to a shared pool of distributed computing resources. Workflow scheduling is one of the key issues in the management of workflow execution. The scheduling process maps and manages the execution of the inter-dependent tasks on the distributed resources. Scheduling of workflows is a challenging task especially when many workflows are considered. Scheduling algorithms are required to implement the workflow scheduling strategies and also for automating the process of scheduling. Proper scheduling can have significant impact on the performance of the system and the user's service requests have to be provisioned with minimal management effort or service provider interaction. This paper presents workflow scheduling for cloud environment based on Artificial Bee Colony algorithm. The objective is to achieve optimization of server utilization and overall computation time to arrive at an efficient scheduling of tasks and services. The proposed work was implemented and tested and the results are presented.

Keywords: *Workflow, Workflow Scheduling, Cloud, Optimization, ABC*

1. INTRODUCTION

Computing is being transformed to a model consisting of services that are commoditized and delivered in a manner similar to traditional utilities such as water, electricity, gas, and telephony. In such a model, users can access services based on their requirements without regard to where the services are hosted or how they are delivered. Several computing paradigms have promised to deliver this utility computing vision and these include cluster computing, Grid computing, and more recently Cloud computing [1].

Cloud computing architecture typically consists of a front end and a back end connected by Internet or Intranet networks [2]. The front end comprises of client devices which can be thin client, fat client or mobile devices. The clients need some interface and applications for accessing the cloud computing system. The back end consists of the various servers and data storage systems. A central server is typically used for administering the cloud system which includes monitoring the overall traffic and fulfilling the client demands in an efficient manner.

Cloud computing is a model for enabling convenient, on-demand network access to the shared pool of distributed computing resources that can be rapidly provisioned and released with minimal management effort or service provider interaction [3]. Cloud computing uses the theme of virtualization: of data storage, of local networks as well as software [4, 5] to meet the demands of the client. Cloud computing is a new form of distributed computing that delivers infrastructure, platform, software and applications as services. Cloud generally provides three levels of services: Infrastructure as a Service (IaaS), Platform as a Service (PaaS) and Software as a Service (SaaS).

Clouds can be classified as public, private and hybrid [6]. When the cloud is made available for the general customer on a pay-per-use basis, then it is termed as a public cloud. When organizations develop their own applications and run their own internal infrastructure then is known as private cloud as access is limited to users within the organization. Hybrid cloud is obtained by the integration and consolidation of public and private cloud. A community cloud is shared among several organizations and can be maintained by the participating organizations or by a third-party

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

managed service provider. The participant organizations can realize the benefits of a public cloud with their added level of services, performance requirements, security and policy compliance [3].

Cloud computing can provide full scalability, reliability, high performance and relatively low cost solution as compared to dedicated infrastructures. Location of data storage and application execution is a critical issue that needs to be addressed. Many organizations have made the transition to support access to organizational data from mobile devices to improve workflow management and gain other operational efficiencies and productivity benefits. Data maintained and processed in a public cloud may present less of a risk to an organization with a mobile workforce than having that data dispersed on portable computers, embedded devices, or removable media out in the field, where theft and loss routinely occur [7].

Applications hosted and executed using clouds are often composed from a set of services which form a workflow. Workflow processing requires tasks to be executed based on their control or data dependencies. As workflow scheduling is a well-known NP-complete problem [8], many heuristic and meta-heuristics methods have been proposed for distributed systems like grids [2]. The concept of running workflow instances on public cloud processing platforms is however still in its infancy [6]. For a utility service like cloud computing, pricing is dependent on the level of Quality of Service (QoS) offered. Typically service providers charge higher prices for higher QoS. Users may not always need to complete workflows earlier than they require and instead, may prefer to use cheaper services with lower QoS that are sufficient to meet their requirements [10]. Workflow scheduling algorithms have been modified and extended for Cloud workflow management. However, scheduling workflows based on users' QoS requirements, such as deadline and budget, has been given very little attention in these existing Cloud workflow management systems [7, 9].

This paper presents an optimization algorithm for work flow scheduling to achieve optimization of server utilization and overall computation time. The rest of the paper is organized as, section 2 contains a literature survey about scheduling in cloud network, section 3 describes about the work flow scheduling based on ABC algorithm, section 4 discuss the experimental results of the proposed approach and section 5 shows the conclusion of the research.

2. RELATED WORKS

As described in [1], a cloud workflow system can be regarded as a type of platform service that facilitates the automation of distributed large-scale e-business and e-science applications in the cloud [4, 5]. A workflow is typically described by a Directed Acyclic Graph (DAG) in which each computational task is represented by a node, and each data or control dependency between tasks is represented by a directed edge between the corresponding nodes [8]. As a cloud service itself, a cloud workflow system belongs to a specific service provider and under the management of its cloud resource managers. Workflow scheduling has a vital role to play in workflow management [8]. Workflow scheduling discovers resources and allocates tasks on suitable resources to meet the users' requirements. Scheduling of workflows is a challenging task especially when many workflows are considered. The management and scheduling of resources in cloud environment is a complex task [3].

Scheduling algorithms are required to implement the workflow scheduling strategies and also for automating the process of scheduling [1].

Considerable research work has been carried out in the area of scheduling in traditional distributed systems, like Grids. However, there has been relatively little work with respect to scheduling in Cloud computing environments. Some of the significant work is presented in this section.

Li Wenhao et al [3] have proposed a workflow system framework for Community Cloud and a task scheduling methodology called the Unified Scheduling Strategy. Tests were performed for different workloads to schedule each process to relevant users according to priority. The scheduling method can be enhanced for supporting process-driven fast collaboration. Bhaskar Prasad Rimal et al [10] have proposed a framework for scientific workflow systems for multi-tenant clouds. Scientific workflows were implemented using BPEL standard language. They have explored two methods: semantic based workflow and policy-based workflow. They have considered authorization, authentication, SLA and dynamic resource allocation. This system was found to be equally suitable when tested with different scenario, such as, large scale setup, like Open Cirrus research cloud hub. The major problem is to isolate the data and maintain the security.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Riktesh Srivastava [11] has proposed FAIR scheduler for scheduling jobs for multiple users. FAIR is used by the master node to distribute waiting tasks to computing nodes (slaves) in response to the status messages it receives from the nodes. FAIR scheduler is seen to improve response time of small jobs and guarantees resources for large jobs. However, the scheduler does not consider other aspects of scheduling such as performance optimization, data localization and QoS.

BaominXu *et al.* [12] have proposed their job scheduling based on Berger model. In the job scheduling process, the algorithm establishes dual fairness constraints. The first constraint was used to classify user tasks by QoS preferences, and establish the general expectation function in accordance with the classification of tasks to restrain the fairness of the resources in selection process. The second constraint was used to define resource fairness justice function to judge the fairness of the resources allocation. The experimental results showed that the algorithm was able to effectively execute the user tasks based on Quality of service. Other aspects of scheduling, such as, optimization of server utilization and security has not been considered. Yujia Ge, Guiyi Wei [13] has proposed a scheduler where scheduling decision is made by evaluating the entire group of tasks in the job queue. Genetic algorithm was applied for server optimization for scheduling in cloud computing environment. The simulation results showed that their scheduler showed better performance when compared to First-in-First-out (FIFO) and Delay Scheduling to achieve a better balanced load across all the nodes in the cloud. The authors have suggested that as future work, developing techniques for optimum balance of computation time and efficiency to arrive at better optimization. V. Nelson and V. Uma [14] have introduced an Inter-cloud Resource Provisioning System (IRPS) which enables the fulfillment of customer requirements by providing additional resources to the cloud system participating in a federated environment. Their proposed work dealt with the interoperability problem between cloud providers in a federated cloud environment. The tasks were allocated these resources with the use of a semantic scheduler and an inference engine. When a cloud system was found to run short of resources, it was supplemented by other cloud systems in the federated environment so as to fulfil the customer requirements.

In order to have an efficient and cost effective job scheduling, application schedulers have to consider different aspects and policies that can vary according to their objective or function. These aspects could be: to minimize total execution time, minimize total cost to execute, balance the load on resources used while meeting the deadline constraints of the application.

3. PROPOSED WORKFLOW SCHEDULING ALGORITHM

Cloud Computing helps user applications to dynamically provision computing resources at specified locations. Workflow scheduling plays a vital role in the workflow management. Proper scheduling can have significant impact on the performance and utilization of the system [9].

The proposed work aims to achieve optimization of server utilization and overall computation time to arrive at an efficient scheduling of tasks and services. The proposed work applies the Artificial Bee Colony (ABC) [15] algorithm to schedule work flows for cloud computing. ABC is an optimization algorithm that is based on the intelligent behavior of honey bee swarm. To the best of our knowledge, ABC algorithm has not been used for job scheduling in cloud or grid environment. ABC is known to provide optimum solutions to complex problems and has significant advantages when compared with random search and enumeration techniques.

The application of the algorithm is explained by considering a sample scenario. It is assumed that there are three servers (S_1, S_2, S_3) and three processes or jobs (p_1, p_2, p_3). The servers associated with the process are given by:

$$S_1 : [p1]$$
$$S_2 : [p2]$$
$$S_3 : [p3]$$

The user requests or processes are executed by the servers. Process assignment must take into consideration that two processes cannot be scheduled for the same cloud server at the same time. U_0 denotes the list of jobs requested by user1 and U_1 and U_2 the list of jobs requested by user2 and user3 respectively and can be represented as shown below:

$$U_0 \rightarrow [p2, p1, p3]$$
$$U_1 \rightarrow [p1, p3, p2]$$
$$U_2 \rightarrow [p2, p1, p3]$$

User requests are stored in the process queue, which is defined as R and can be represented as:

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

$$R = [U_0, U_1, U_2]$$

The above expression is the preliminary service request to the servers from the users. The next step is to provision the services to the users according to their availability. Each job is associated with a quantum value which indicates the time quanta required to complete the process. Quantum values in the range of 1 to 3 were considered for the defined processes which are represented as:

$$p1 = [quatum : 3]$$
$$p2 = [quatum : 2]$$
$$p3 = [quatum : 1]$$

The initial population is the set of processes or tasks that have to be scheduled. The jobs have to be scheduled taking into consideration the time quantum of each process and the list of services requested by the users. The initial population expressed in terms of their time quantum is given as:

U_0 - [2, 1, 3]
U_1 - [1, 3, 2]
U_2 - [2, 1, 3]

Before applying the ABC algorithm, the fitness computation of the initial population is first performed. The services are assigned to the users as per their requests and the tasks have to be completed in the order in which it is requested by the users. The user requests are assigned to the appropriate servers based on which server is capable of performing which process. In the given scenario, for example, server s_1 is capable of executing process p1. User request U_0 is first processed. The first service requested is p2 which is assigned to S_2. The first request in U_1 is p1 which is assigned to S_1. The first request of U_2 is p2. However, S_2 is busy with request of U_0, hence U_2 request has to wait till p2 is completed, which requires 2 time quanta. The services requested are assigned in the order of user requests and is continued till all the three user requests are satisfied as shown in Table 1.

Table 1: Fitness Computation for Initial Population

Servers / Time Quanta	**S_1**	**S_2**	**S_3**
1	U_1	U_0	Null
2	U_1	U_0	Null
3	U_1	U_2	Null
4	U_0	U_2	U_1
5	U_0	U_1	Null
6	U_0	U_1	Null
7	U_2	Null	U_0
8	U_2	Null	Null
9	U_2	Null	Null
10	Null	Null	U_2

It can be noted that 10 iterations are required to complete all the user requests. However, the server utilization is not optimized as the servers are idle for different periods of time as seen from Table 1.

The workflow scheduling is now applied to the initial population and carried out in three phases as i) scheduling multiple process on time, ii) Improving the cyclic process and iii) random phase which is applied when a best solution is not obtained in the previous phase.

3.1. Scheduling Multiple Process on Time

3.1 In order to obtain an effective scheduling procedure, the employed bee phase of ABC algorithm is first applied. A position update process is performed to improve the chances of getting a better processing order of the tasks to be scheduled. The position update process assumes that the tasks requested by a user can be performed in any random order. However, only one process request of a user would be scheduled at any given time. The positions are updated randomly with the help of a uniform distribution. Then the fitness of each service is evaluated by a fitness function. If the fitness of the service has not improved, then new solution is searched for iteratively. The phase is completed when there is no significant improvement in the fitness function.

Each services x in R can be defined using coordinates i and j as:

$$x_{ij} = position(p_i) : i = 1,2,...,n;\ j = 1,2,....,m \quad (1)$$

f(pi) is defined as the difference in computation time of the particular service to total computation time of all the services present in that user's request and is given as:

$$f(p_i) = \left[\sum_{j=0}^{n} time(p_j)\right] - time(p_i) \ldots\ldots\ldots(2)$$

The fitness of each service is evaluated by a fitness function.

$$fit(p_i) = \begin{cases} \dfrac{1}{1+f(p_i)}, if\ f(p_i) \geq 0 \\ 1 + abs(f(p_i)_i), if\ f(p_i) \leq 0 \end{cases} \ldots\ldots\ldots(3)$$

The fitness values of each services p_i are calculated using the fitness function. For the given example R, total computation time of all the services present in that user's request is [3+2+1] = 6. Fitness value calculated for the three user processes are given in Table 2.

Journal of Theoretical and Applied Information Technology
30th November 2013. Vol. 57 No.3

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

Table 2: Fitness Values of Services

Process	Process time	$f(p_i)$	$fit(p_i)$
p_1	3	6 – 3 = 3	0.25
p_2	2	6 – 2 = 4	0.20
p_3	1	6 – 1 = 5	0.16

Hence the initial population R in terms of fitness value becomes:

[2, 1, 3] as [0.2, 0.25,0 .16]
[1, 3, 2] as [0.25, 0.16, 0.2]
[2, 1, 3] as [0.2, 0.25, 0.16]

A position update process is then performed to get a better processing order of the services that improves the server optimization. The formula used for process updated is given in equation3.

$$v_{i,j} = x_{i,j} + \Phi_{i,j}(x_{i,j} - x_{k,j}) \cdots\cdots\cdots\cdots (4)$$

where, $v_{i,\,j}$ represents the updated position, $x_{i,\,j}$ is the current position and the value $\Phi_{i,j}$ is a random number between (-1 and 1).

The updated position that is obtained is given as:

Cycle 1: [[2, 1, 3],[3, 2, 1], [2, 1, 3]]
Cycle 2: [[1, 2, 3],[1, 3, 2], [2, 3, 1]]
Cycle 3: [[2, 1, 3],[3, 1, 2], [2, 1, 3]]

The fitness value is once again calculated for the updated position and the value obtained was 15. This process is repeated and the fitness value obtained for cycles 2 and 3 were 9 and 15. Since there is no improvement in the fitness value, the scheduling is forwarded to the next phase to check for further improvement.

3.2 Improving the Cyclic Process

In this phase, the services are selected based on the probability values of the particular service. The services with best probability values are updated and the best fitness is selected as the best solution. The probability value specifies the relevance of the particular service for the scheduling process. The probability values are calculated in respect to fitness values of the previous phase as:

$$probability(p_i) = \frac{fit(p_i)}{\sum_{i=0}^{d} f(p_i)} \cdots\cdots\cdots (5)$$

where, probability (p_i) represents the probability value of the service p_i and the variable d represent the dimension of the process table. The probability of all the services is calculated based on the probability equation (5). The fitness values and probabilities for the considered example is given in Table 3.

Table 3: Fitness Values of Services

Process	Process time	$f(p_i)$	$fit(p_i)$	*Probability (p_i)*
p_1	3	6 – 3 = 3	0.25	0.08
p_2	2	6 – 2 = 4	0.20	0.05
p_3	1	6 – 1 = 5	0.16	0.032

The initial population now becomes:

[2, 1, 3] as [0.05, 0.08, 0.032]
[1, 3, 2] as [0.08, 0.032, 0.05]
[2, 1, 3] as [0.05, 0.08, 0.032]

The services with the best probability values are used in the position update phase, which is similar to the previous phase.

$$pos_{new}(O_i) = pos(O_i) + \Phi_{i,j}(pos(O_i) - pos(O_k)) \cdots\cdots (6)$$

The fitness value is once again calculated for the updated position and the value obtained was once again 15. This process was repeated and the fitness value obtained for cycles 2 and 3 were 9 and 15. It can be noted that again, there is no improvement in the fitness value and hence the scheduling is forwarded to the next phase to check for further improvement.

3.3 Random phase

In this phase, a service request with least probability is calculated and that service is replaced with a randomly generated service request. This process increases the chances of improving the solution. Then fitness of the solution is calculated and position update process is also applied to find out whether there is any improvement in the obtained solution. The process is repeated till the best solution is obtained. If a best solution is not obtainable, then the process is abandoned and the initial population is taken for scheduling.
For example, the least probability is obtained in the previous step was 0.032 for p3. The initial population is modified as:

[2, 1, 3] as [0.032, 0.08, 0.032]
[1, 3, 2] as [0.08, 0.032, 0.05]

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[2, 1, 3] as [0.05, 0.08, 0.032]

The services requested are assigned [[3,2,1],[2,1,3], 1,2,3]] in this specified order as shown in Table 3.

Table 3: Fitness Computation for Modified Population

Servers / Time Quanta	S_1	S_2	S_3
1	U_2	U_1	U_0
2	U_2	U_1	Null
3	U_2	U_0	U_1
4	U_1	U_0	Null
5	U_1	U_2	Null
6	U_1	U_2	Null
7	U_0	Null	U_2
8	U_0	Null	Null
9	U_0	Null	Null
10	Null	Null	Null

It can be noted that 9 iterations are required to complete all the user requests. Even though the servers are idle for different periods of time, there is still as improvement over the initial process. Hence this solution can be chosen as the order for executing the user requests.

4. IMPLEMENTATION AND RESULTS

The proposed approach was simulated using ***cloudsim*** tool. ***Cloudsim*** provides a generalized and extensible simulation framework that enables seamless 622modelling, simulation, and experimentation of emerging cloud computing infrastructures and application services [11].
The performance of the proposed algorithm was compared with the Genetic Algorithm (GA) [13] based scheduling algorithm given in the related work in section 2. GA starts with an initial set of random solutions called population. Each individual in a population is called a chromosome. The crossover, mutation and selection operations are applied in the procedure till the specified fitness criteria are reached or a specified number of iterations have been completed.
Considering three different services and service requests arranged randomly for a set of 100 users, the fitness value was calculated and the results obtained is given in Table 4.

Table 4: Comparison of Fitness Value

Iteration	I1	I2	I3	I4	I5	I6	I7	I8	I9	I10
ABC	132	107	91	76	60	39	24	21	21	21
GA	28	25	26	26	30	30	29	29	29	29

The results are also graphically represented in Figure 1.

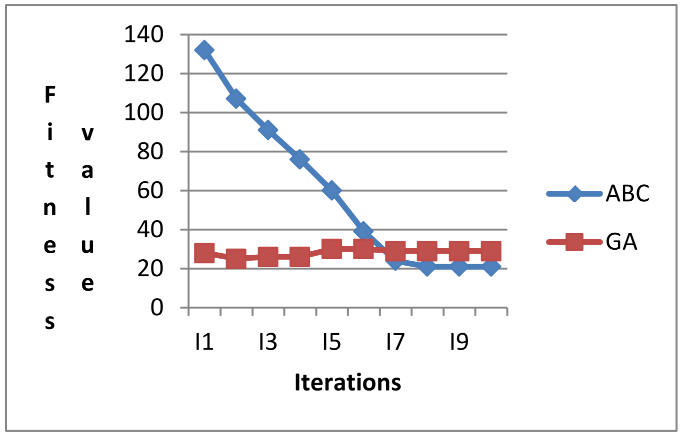

Figure 1: Fitness Value Comparison

It can be seen that GA performs better in the initial iterations. However, the proposed approach reaches a better fitness value of 21 as compared to 29 obtained by GA method. Hence the proposed method may be considered more efficient as compared to the GA based scheduling.

5. CONCLUSION AND FUTURE WORK

This paper presents an optimized workflow scheduling for cloud computing based on ABC algorithm. The proposed approach aims to improve the server utilization while arriving at an efficient schedule of services that satisfy all the user's requests. The proposed work was implemented and tested. The proposed work showed an improvement in performance when compared with GA based scheduling. As future work, it is planned to extend the work to include other criteria such as, user given priority for scheduling jobs, inter-dependency between the task requests and also to balance the load on resources used. More extensive testing using large scale setup is also proposed.

REFERENCES

[1] Rajkumar Buyyaa, Chee Shin Yeo, Srikumar Venugopal, James Broberg, Ivona Brandic ,"Cloud Computing and Emerging IT platforms: Vision, Hype, and Reality for Delivering Computing as the 5th Utility," *Journal Future Generation Computer Systems*, Vol. 25 no. 6, 2009.

ISSN: 1992-8645 www.jatit.org E-ISSN: 1817-3195

[2] Liu, K.,“Scheduling Algorithms for Instance Intensive Cloud Workflows”, *Ph.D. Thesis*, Swinburne University of Technology, Australia, 2009.

[3] LI Wenhao, “A Community Cloud Oriented Workflow System Framework and its Scheduling Strategy”, *IEEE 2nd Symposium on Web Society (SWS)*, 2010.

[4] Leavitt N, 2009, “Is Cloud Computing Really Ready for Prime Time?,” *Computer*, Vol. 42, pp. 15-20, 2009.

[5] Weinhardt C, Anandasivam A, Blau B, and Stosser J, ‘Business Models in the Service World’, *IT Professional*, Vol. 11, pp. 28-33, 2009.

[6] John C. Mace, Aad Van Moorsel, Paul Watson, “The Case for Dynamic Security Solutions in Public Cloud Workflow Deployments,” *Proceeding of the IEEE/IFIP 41st International Conference on Dependable Systems and Networks Workshops*, 2011.

[7] Zhangjun, W., Xiao, L., Zhiwei, N., Dong, Y., and Yun, Y.“A Market-Oriented Hierarchical Scheduling Strategy in Cloud Workflow Systems”, *Journal of Supercomputing*, 2011.

[8] Yu, J., and Buyya, R., “Workflow Scheduling Algorithms for Grid Computing,” Xhafa F, Abraham A (eds), *Metaheuristics for scheduling in distributed computing environments*, Springer, Berlin, 2008.

[9] Ke, L., Hai, J., Jinjun, C., Xiao, L., Dong, Y., and Yun, Y, “A Compromised-Time-Cost Scheduling Algorithm in SwinDeW-C for Instance-Intensive Cost-Constrained Workflows on Cloud Computing Platform”, *International Journal of High Performance Computing Applications*, pp.1-16,2010.

[10] Bhaskar Prasad Rimal, Mohamed A. El-Refaey, “A Framework of Scientific Workflow Management Systems for Multi-Tenant Cloud Orchestration Environment,” *Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises*, 2010.

[11] Riktesh Srivastava, “Implementation of Job Scheduling Algorithms in a Cloud Computing Environment for B2C Electronic Commerce Architecture using GI/G/1 Queuing Model”,2008

[12] Baomin Xu, Chunyan Zhao, EnzhaoHu, Bin Hu, "Job Scheduling Algorithm Based on Berger Model in Cloud Environment", *Advances in Engineering Software*, Vol. 42, no. 7, pp. 419 - 425, 2011.

[13] Yujiage, Guiyiwei, “GA – Based Task Scheduler For the Cloud Computing Systems(WISM’10)”, *Proceedings of the International Conference on Web Information Systems and Mining*, Vol. 02, pp181 – 186, 2010.

[14] V. Nelson,V. Uma, “Semantic based Resource Provisioning and Scheduling in Inter-Cloud Environment,” *International Conference on Recent Trends in Information Technology (ICRTIT)*, pp.250-254, 2012.

[15] Dervis Karaboga and Bahriye Basturk, "A Powerful and Efficient Algorithm for Numerical Function Optimization: Artificial Bee Colony (ABC) Algorithm", *Journal of Global Optimization*, Vol. 39, pp. 459-471, 2007.