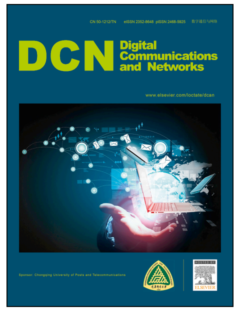
Accepted Manuscript



An efficient IoT cloud energy consumption based on genetic algorithm

Farnoosh Farhadian, Mohammad Mansour Riahi Kashani, Javad Rezazadeh, Reza Farahbakhsh, Kumbesan Sandrasegaran

PII: S2352-8648(17)30167-0

DOI: https://doi.org/10.1016/j.dcan.2019.01.001

Reference: DCAN 150

To appear in: *Digital Communications and Networks*

Received Date: 13 May 2017

Revised Date: 23 December 2018

Accepted Date: 4 January 2019

Please cite this article as: F. Farhadian, M.M.R. Kashani, J. Rezazadeh, R. Farahbakhsh, K. Sandrasegaran, An efficient IoT cloud energy consumption based on genetic algorithm, *Digital Communications and Networks* (2019), doi: https://doi.org/10.1016/j.dcan.2019.01.001.

Digital Communications and Networks(DCN)

HOSTED BY
ELSEVIER
Available online at www.sciencedirect.com
ScienceDirect
journal homepage: www.elsevier.com/locate/dcan

An Efficient IoT Cloud Energy Consumption Based on Genetic Algorithm

Farnoosh Farhadian[a], Mohammad Mansour Riahi Kashani*[a], Javad Rezazadeh[ac], Reza Farahbakhsh[b], Kumbesan Sandrasegaran[C]

[a]Dept. of Electrical and Computer Engineering, Islamic Azad University, North Tehran Branch, Tehran, Iran.
[b]Institut Mines Telecom and University Paris Saclay, Telecom SudParis, CNRS Lab UMR5157, 91011 Evry, France.
[c]School of Electrical and Data Engineering, Faculty of Engineering and IT, University of Technology Sydney, Australia.

Abstract

Cloud computing is a promising technology which enables processing a great amount of data. In a large scale IoT infrastructure, optimum allocation of virtual machines to the physical hosts leads to reduce energy consumption of data centers. Moreover, it may prevent pollution of the environment and improve the efficiency. The comprehensive study of this research shows that employing the various methods of virtualization, proper combination of these methods and applying evolutionary strategies is promising to develop new algorithms. The developed novel algorithms could improve the energy efficiency of cloud computing. In this study, an effort is made to provide an optimum virtual machines allocation to physical hosts with employing imperialistic competitive (IC) and genetic algorithms (GA). Toward that end, the crossover operation from GA is mixed with IC. Cloudsim software is used for simulation and Netbean software for implementation. Considerable improvement on energy consumption is achieved through the proposed method.

KEYWORDS: Cloud Computing, Resources Allocation, Energy Consumption, Genetic Algorithm, IoT.

1. Introduction

Internet of Things (IoT) is one of the leading technologies in ubiquitous computing and the key solution for providing the smart environments [1]. With the appearance of virtualization technology, the usage of resource computing starts being used in the form of virtual machines. There are many advantage in this technology where resource efficiency improvement, cost reduction and the convenience of servers management are just a few examples. In all IoT applications, different types of data must be collected and analyzed, depend on the type of data, different techniques must be applied and make prediction [2]. This phenomenon enables computation in a form of cloud computing which has always been a dream. Prof. Buyya, a well-known researcher in the field of cloud computing, believes that Cloud is a parallel and distributed computing system consisting of a collection of interconnected and virtualized computers that are dynamically provisioned and presented as one or more unified computing resources based on service-level agreements (SLA) established through negotiation between the service provider and consumers [3].

Cloud computing is a model for enabling convenient, on-demand network access to shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be

*Asst.Prof. Mohammad Mansour Riahi Kashani (Corresponding author) is an Assistant Professor in Islamic Azad University North-Tehran Branch (email:m_riahi_kashani@iau-tnb.ac.ir).

[1]Farnoosh Farhadian is a Master of Science, Software Engineering in Islamic Azad University North-Tehran Branch. (email:FranooshFarhadian@gmail.com).

[2]Asst.Prof. Javad Rezazadeh is an Assistant Professor in Islamic Azad University North-Tehran Branch and Research Scientist at University of Technology Sydney (email:rezazadeh@ieee.org).

[3]Dr. Reza Farahbakhsh is a senior researcher in Institut Mines Telecom based in France (email:reza.farahbakhsh@it-sudparis.eu).

[4]Assoc.Prof. Kumbesan Sandrasegaran is an Associate Professor in Faculty of Engineering and Information Technology, University of Technology, Sydney (UTS), Australia. (email:Kumbesan.Sandrasegaran@uts.edu.au).

rapidly provisioned and released with minimal management effort or service provider interaction [4]. In large scale IoT infrastructure, cloud is playing an important role and many middleware technologies have been designed for cloud of things [5] and provide many new services in smart cities scenarios [6].

Considering the proposed definition of cloud computing, some essential cloud characteristics includes broad network access, rapid elasticity, on-demand self-service, measured service, and resource pooling [7]. There are two crucial problems in cloud computing, energy consumption and SLA violation. Cloud data centers contain thousands of computing nodes which consume great amount of electrical energy. According to a report published by IDC in 2011, worldwide spending on power and cooling was estimated around $60 million. It is deduced that energy consumption had a rapid growth (around 20 times greater) over the last 15 years. The reason for this extremely high energy consumption is not just the quantity of computing resources and the power inefficiency of hardware, but rather lies in the inefficient usage of these resources [8]. Further studies shows that using different virtual methods, acceptable combination of these methods and using heuristic strategies could help to propose a new algorithms. The proposed algorithm causes acceptable results in the field of optimization energy consumption in cloud computing.

Considering all the mentioned previous studies, in this study the allocation of virtual machines to physical hosts has been improved by enhancing the imperialism competitive algorithm [9] and genetic algorithm [10]. To evaluate the performance of the proposed method, we have used Cloudsim simulator [11, 12, 13] and implemented the proposed idea and compare the results with other available solutions. Our results show we are advancing the previous methods and our solution is showing better results and improving resource allocation. This improvement was around 20% in comparison with PABFD algorithm.
The main outcomes of this research are as follow:

1.We firstly provide a summary of the current solutions on reducing energy consumption.

2. The proposed algorithm in this study named ICGA is novel in compare to other solutions.

3.Our evaluation results show around 20% improvement in compare to other similar solutions.

The rest of the paper is organized as follows: the existing methods proposed for energy efficiency of cloud computing is summarized in Section 2. The proposed algorithm for energy efficiency is presented in Section 3. Section 4 provides simulation, evaluation and results out of our study and finally, simulation and performance evaluation is presented in Section 5.

2. Literature Review

There has been a large body of research on cloud computing focused on improving its energy efficiency. In this section, some of the highlighted methods are briefly reviewed. The relevant studies to this research can be categorized in 2 groups: the first one aim to enhance the energy consuming like Gmach, Srikantaiah, Beloglazoy et al., Nathuji et al , Kusic et al., Verma et al and the second group look at the named algorithm like Madhusudhan et al, Portaluri et al., Asemi et al. Abarghoei et al. All works got good results but we trying to find a way to solve the problem with less energy consumption in regards to green clouds. Gmach et al.,(2009) studied minimizing power usage dynamically using virtualization in enterprise environments. They employed a trace-based workload placement controller and combined it with a proactive reassign controller [9].

Srikantaiah et al., (2009) have studied the problem of request scheduling for multi-tire web-applications in virtualized heterogeneous systems to minimize energy consumption, while meeting performance requirements [10]. In another study, Beloglazoy et al., (2012) proposed several heuristics for dynamic consolidation of Virtual Machines (VMs). They split the problem of dynamic VM consolidation into four parts [8]: (1) Host over loading detection (2) VM selection from the overloading hosts (3) Host under loading detection and lastly (4) Finding a new host and sets VMs for migration to the determined target host. Researchers evaluates the combination of the described methods using Cloud-sim simulator [11] and consequently, Local Regression (LR) for host overloading detection and Minimum Migration Time (MMT) for Virtual Machine selection had optimum results. They employed PABFD algorithm which was presented by Beloglazov et al., (2012) [8] in order to determine a new host for VMs. In PABFD, all the VMs are sorted out in the decreasing order of their current CPU utilization and allocate each VM to a host that provides the least increase of the power consumption caused by the allocation. Local Regression is formed based on Loess method which was introduced by Cleveland et al.,(1979). This is a combination of k- nearest- neighbor oriented multivariate regression methods.

The Minimum Migration Time (MMT) policy migrates a VM that requires the minimum time to complete a migration relatively to the other VMs allocated to the host. The migration time is estimated as the amount of RAM utilized by the VM divided by the spare network bandwidth available for the host. Both RAM and network band width is assumed to be fixed and specified at the time of VMs building. However, it is completely invalid in real world. In Beloglazoy et al. research, a novel and efficient method is proposed for resource allocation with decreasing the number VMs migration. Madhusudhan et al., (2013) proposed a genetic algorithm for placing new virtual machines. The researchers compared their algorithm with thrMmt. It is observed from the results that energy consumption is minimized compared to energy resulted from thrMmt [12]. There are two main approaches for power management in server in server hardware: DVFS (Dynamic Voltage and Frequently Scaling) and dynamic power shutdown. DVFS reduces power consumption by lowering down operating frequency and/or voltage dynamic power shutdown saves power during idle times by powering down as much as possible. Nathuji et al (2007) explored how to integrate power management mechanisms and policies with the virtualization technologies being actively deployed in these environments. The goals of the proposed Virtual Power approach to on-line power management are to support the isolated and independent operation assumed by guest virtual machines (VMs) running on virtualized platforms and to make it possible to control and globally coordinate the effects of the diverse power management policies applied by these VMs to virtualized resources [13].

Power delivery, electricity consumption, and heat management are becoming key challenges in data center environments. In this case Raghavendra et al (2008) first proposed and validated a power management solution that coordinates different individual approaches and second, using unified architecture as the base, they performed a detailed quantitative sensitivity analysis and draw conclusions about the impact of different architectures, implementations, workloads, and system design choices [14].

Kusic et al (2008) implemented and validated a dynamic resource provisioning framework for virtualized server environments wherein the provisioning problem is posed as one of sequential optimization under uncertainty and solved using a look-ahead control scheme. The proposed approach accounts for the switching costs incurred while provisioning virtual machines and explicitly encodes the corresponding risk in the optimization problem [15].

Verma et al (2008) investigated the design, implementation, and evaluation of a power-aware application placement controller in the context of an environment with heterogeneous virtualized server clusters. First, they presented multiple ways to capture the cost-aware application placement problem that may be applied to various settings. In the second part they presented the pMapper architecture and placement algorithms to solve one practical formulation of the problem: minimizing power subject to a fixed performance requirement [16]. Researchers in (Portaluri et al., 2014) developed an approach based on genetic algorithms with a power efficient resource allocation. It successfully enables to minimize system power consumption and operational expenses. Moreover, a reasonable trade-off is achieved among task completion time and power consumption.

Asemi et al., (2015) focused on this sun-problem and aimed to understand where should virtual machines migrate. They proposed a method based on the imperialist competitive algorithm for optimal allocation of virtual machines to physical hosts. In this method, an array refers to one solution (one country). Each index in this array points to each virtual machine and its value point to the physical host which virtual machine places. For each virtual machine (each index of array), an available list of hosts is provided and index value is selected randomly from a list has been linked to each index. Then, Imperialist competitive algorithm searches the best solution according to algorithms steps. The researchers considered two different scenarios, the first one which tried to simulate a small datacenter with 100 physical hosts and the second one, included a data center with 800 virtual machines. This approach in both scenarios has caused the energy consumption in physical hosts with 31.25% reduction in comparison to similar algorithms such as GA or PSO [18].

Abarghoei et al., (2015) proposed an algorithm which is a combination of imperialist competitive and local search optimization algorithms for resource allocation in cloud computing and the scheduling of each user works on existing virtual machines. The algorithm attempts to improve the possible responses by creating an initial empire through applying imperialist competitive algorithm. To avoid premature convergence, imperialist competitive algorithm is combined with a local search algorithm. Quality and efficiency of the proposed algorithm are compared with round-robin, ant colony and genetic algorithms. The results show that the proposed algorithm in terms of time and the responses quality is faster than the ant colony optimization and genetic algorithms [19].

Taleb et al., (2017) proposed an integration of energy aware Wireless sensor Networks (WSNs) in cloud computing [20]. In WSNs, reported data is meaningless if its location is unknown [21, 22, 23], therefore most recent research addressed this problem [24, 25, 26, 27]. The authors in [20] explained that cloud computing is promising technology that can provide a flexible storage and processing infrastructure to process WSN's data in large scale.

In summary of this section, Table 1 shows a comparison of the related works.

3. Proposed Approach

In this section, we discus and investigate imperialistic competitive algorithm and how to apply it by the proposed algorithm, and the role of the genetic algorithm plays in our developed method. The developed algorithm includes five different steps that explain in this section

3.1. Imperialist Competitive Algorithm

Imperialist competitive algorithm is a method in the field of evolutionary computation which tries to find

Table 1: Review of related works

REF	Goal	Method	Result
Ref[13], 2007	Studying power management techniques in virtual systems	Using small part of VM time in order to collect information on the efficiency of resource use monitoring capabilities in virtual machines	1. Provide a new power management technique on a virtual systems scale as scaling software source 2.Reduce power consumption up to 34% using the proposed method combines virtual machines
Ref[14], 2008	Power management issues for a data center	Focus on 5 specific solution for managing all aspects of power management and provide 5power management controllers for them	Reduce power consumption in the volume of work
Ref[15], 2008	Power management and performance issues in virtual computer systems	In order to consider the issue as an optimization problem and solve it using the limited look-ahead control (LLC)	26% reduced energy consumption using managed cluster by LLC in 24h period in compare with a uncontrolled system
Ref[16], 2008	Minimize power consumption in the embedded dynamic applications on virtualized systems	Considering the issue as a continuous optimization problem: in any time frame, location of VM must be optimized to minimize power consumption and maximum performance	1.Provide a two-phase algorithm called Power packing in order to minimize the costs and problem-solving Placement Program 2. 25% more savings in power over the placement of load balancing algorithms and static
Ref[29], 2012	Dynamic combination of Virtual Machines	Introducing 4 host overloading determination method, proposing 3 policies for VMs selection. Cloud Sim utilization for evaluation	Optimum results for: Local Regression (LR) for host overloading detection and Minimum Migration Time (MMT) for Virtual Machine selection
Ref[12], 2013	placing new virtual machines	proposed a genetic algorithm for placing new virtual machines and comparing their algorithm with thrMmt	Minimizing the energy consumption compared to energy consumed by thrMmt
Ref[10], 2009	Minimizing energy of request scheduling for multi-tire web-applications in virtualized heterogeneous systems	Applying a heuristic method for multi-dimensional bin- packing problem of workload combination	Energy consumption per transaction in a U-shape curve. Optimum efficiency point is specified.
Ref[9], 2009	minimizing power usage dynamically using virtualization in enterprise environments	Proposing a combination of trace-based workload placement controller and a proactive reassign controller	Optimum efficiency achieved by simultaneous controllers employment and calling workload placement controller every 4 hours
Our Proposed Method, 2018	Reducing energy consumption and immigration in VM in cloud	Proposing a combination of IC and Genetic algorithm	Proposed algorithm improved result of PABFD algorithm about 20%

optimum solutions. The fundamental of the algorithm is formed based on assimilation policy, imperialist competitive and revolution. This algorithm simulates the process of social, economic and political evolution of the countries and sequentially proposes some operators in the form of algorithm which mitigates complex optimization problems. Indeed, the algorithm regards the solution of the optimization problem in the form of empires and tries to gradually optimize them through the frequent processes [28]. The various steps of the algorithm will be investigated as follows.

At the first step, we assume there are N initial empires, $N_{country}$. Then, imperialists are selected as the best members of the population (empires with minimum cost function) and shown by N_{imp}. The remains are categorized as separate colonies, N_{col} which belongs to different empires. The number of dedicated colonies to every empire depends on the empires power. The more power of the empire, the greater the numbers of colonies. In this research, empires represent the list of duties in various sizes. The duties correspond to Cloudlets in simulator and each index includes VMs. Table 2 illustrates a sample of an empire.

Table 2: A sample of an empire, the number of duties and machines are equivalent (6)

VM6	VM5	VM4	VM3	VM2	VM1
Cloudlet6	Cloudlet2	Cloudlet5	Cloudlet3	Cloudlet4	Cloudlet1

3.1.1. Colonial and imperialist displacement

Some of these colonies may reach to better position than their imperialist (the points in the cost function in which produce less cost than the amount of cost in imperialist position). In this case, the colonizer and the colonized will change positions with each other and the algorithm is continued with the colonizer in the new position, and this time, the new imperialist country that begins to apply assimilation policy on its colonies.

3.1.2. Imperialistic Competitive

Empire's power is equal to the power of an imperialistic country, plus a percentage of total power of its colonies. By forming empires, imperialist competitive between them begins. Every empire that fails to compete and increase its power will be removed from the scene of imperialist competitive. During the imperialist competitive, the power of larger empires is gradually increase and weaker empires will be removed. Empires will be forced to develop their own colonies to increase their power.

3.2. The Role of Genetic Algorithm

Genetic algorithm is based on Darwin evolutionary theory (1895). Indeed, genetic algorithms belong to the larger class of evolutionary algorithms. In genetic algorithms, genetic operators are employed through reproduction stage. Regarding the impact of these operators on a population, the next generation would be generated. Selection, crossover and mutation operators are most applicable operators in Genetic algorithms. Algorithm 1 shows pseudo code of Genetic Algorithm. In this study we employ crossover operator and apply it in the displacement step of imperialist competitive algorithm. The main thing is the continues search environment of imperialist competitive algorithm is changed to discrete one via employing crossover operator.

Algorithm 1 Pseudo code of Genetic Algorithm

```
1: Choose initial population
2: repeat
3:     Evaluate the individual fitness of a certain proportion of the population
4:     Select pairs of best-ranking individuals to reproduce
5:     Apply crossover operator
6:     Apply mutation operator
7: until terminating condition
```

3.3. Imperialist Competitive mixed with Genetic Algorithm (ICGA)

The developed algorithm ICGA includes five steps: generating initial empires, assimilation policy, applying crossover and revolution operators and cost function that describe as follows.

3.3.1. Generating Initial Empires

Generating initial population step aims to select some random points on the function and generate initial empires. In this step, every empire modeled is defined as an array which its length is equivalent with the number of VMs. Moreover, the indexes include the numbers from 0 to physical hosts number. In other words, in each array its index and value represent VM number and physical host number, respectively. Table 3 illustrates VMs allocation issue using imperialistic competitive algorithm. One-dimensional array is represented by the Table 3 while arrays show VM number and the value present physical host number.

Table 3: VM number and value presents physical host number

VM ID	**0**	**1**	**2**	**3**
HOST ID	**2**	**4**	**0**	**1**

3.3.2. Assimilation Policy Implementation

Assimilation policy, the second stage of the proposed algorithm, is introduced as "move colonies toward imperialist countries" According to the policy, it is tried to move colonies closer to the imperialists. In this case, we choose some indexes of array and change them according to the imperialism ones.

3.3.3. Crossover Operator

We apply combination function to two answers via crossover operator. Middle point of the array is selected randomly and then, former part of primary answer is combined with latter part of secondary one and vice versa. The earlier answers would be replaced by the calculated findings, if recent answers are optimum.

3.3.4. Revolution Operator

Revolution and mutation operators have almost the same function in imperialistic competitive algorithm. Both make an effort to avoid trapping in local optimal. Implementing this step is feasible and only needs to modify some indexes, randomly.

3.3.5. Propriety Function

Imperialistic competitive algorithm, like other algorithms, has a propriety (cost) function. In this research, we assume that the cost function is calculated by the energy consumption. The developed algorithm named Imperialist Competitive mixed with Genetic Algorithm (ICGA) is described in details in algorithm 2.

Algorithm 2 Algorithm Imperialist Competitive mixed with Genetic Algorithm (ICGA)

```
1: Create and initialize the colonies
2: Select the empires
3: repeat
4:     Move the colonies toward their relevant imperialist
5:     Apply crossover operator
6:     if New children of crossover are better than their parent then
7:         Exchange their position
8:     else if There is a colony which has lower cost than it's imperialist then
9:         Exchange their position
10:    end if
11:    Compute the total cost of all empires
12:    Pick the weakest colony from the weakest empire; give it to the empire that has the most -likelihood to possess it
13:    Eliminate the powerless empires (empire with no colony)
14: until Just one empire remained
```

4. Evaluation and Results

4.1. Simulation Setup

In this research, we employ a powerful and accurate simulation for cloud computing known as Cloud Sim simulator. Some features of the simulators include flexibility, graphical output, tables and diagrams information report and so on. In order to evaluate the proposed approach, the first step is to simulate cloud datacenter circle, then, VMs' interface and finally duty lists.

Table 4: Simulation parameter setting

	Storage	RAM	MIPS	Band Width	Max Power
Physical Host	1,000,000	10,000	1000, 2000, 3000	100,000MBPS	250w
Virtual Machine	2500	128	250, 500, 750, 1000	2500	

In this study, the input simulation scenario includes a data center with 10 physical host, $10VMs$ and also 10 duties with $150,000MIPS$ length. Every physical host comprises a processor that will randomly select. Also cloudsim simulator [32], Netbean software and Java library is used to implement the scenario. Table 4 shows the simulation parameter setting and required features.

4.2. Evaluation

In this regards, two scenarios are studied. In first we employ the combination of imperialistic competitive and genetic algorithms and in second scenario without employing imperialistic competitive and genetic algorithm, we employ with PABFD. Finally, these two scenarios compare to each other. We selected PABFD because it is a basic algorithm in cloudsim which is employed in several works to compare with proposed algorithm or creating VMs [8, 32, 33, 36, 37].

4.2.1. Scenario 1: With Employing Imperialistic Competitive and Genetic Algorithm (Hybrid Algorithm)

In this scenario, we first employ the combination of imperialistic competitive and genetic algorithms. Figures 1, 2 and 3 show the results with 10 initial population that run 30, 50 and 70 times, respectively.

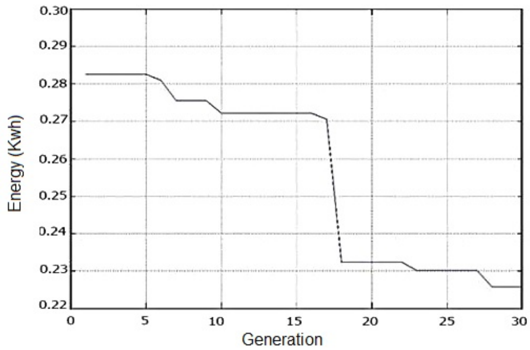


Fig. 1: Energy consumption with 30 iteration

In these results, it is evaluated the efficiency of our proposed method by measuring the energy consumption in Cloud Sim simulator at neat beans environment. The x vector shows the number of runs and the y vector shows the energy consumption in KW. As it is shown in Figure 1, the energy consumption stays in a constant value during the iteration numbers 10 to 17 and the value is just over $0.27KWh$. The energy consumption achieves a great improvement from iteration 17 to 27, however it decreases by echelon and reach around the value of $0.226KWh$. Finally, the ratio of energy consumption keeps constant through 3 latter iterations while stands at $0.226KWh$. In this result, the number of displacement reaches 0 which is a reliable and efficient achievement for the simulation.

Figure 2 reports the result collected by 50 iterations. The displacement rate is still none (same as the result reported by figure 1). The algorithm has experienced consistency during iteration numbers 8 to 15. In spite, there is seen some more fluctuation than figure 1 particularly in the runs number 17 to 37. The variation is because of applying the mutation operation. Finally, the algorithm reaches its efficient value at $0.2562KWh$ by $50th$ run.

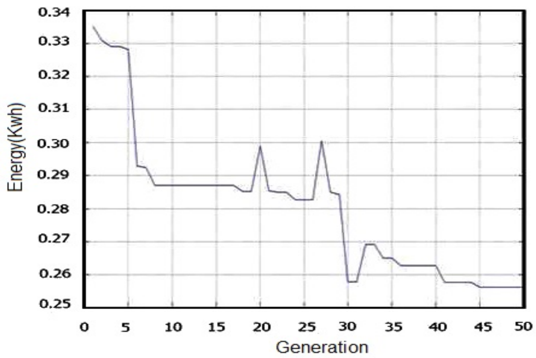


Fig. 2: Energy consumption with 50 iteration

Figure 3 illustrates a ratio of energy efficiency for 70 runs. This result also keeps the displacement rate 0 and furthermore, improves the energy efficiency. The energy consumption was reported around $0.335KWh$ initially by the first run and increased efficiently to beat 0.2257975 at $70th$ iteration. This figure reports $0.1092041KWh$ decline from first to $70th$ run while it has a better improvement (greater difference) than 50 runs.

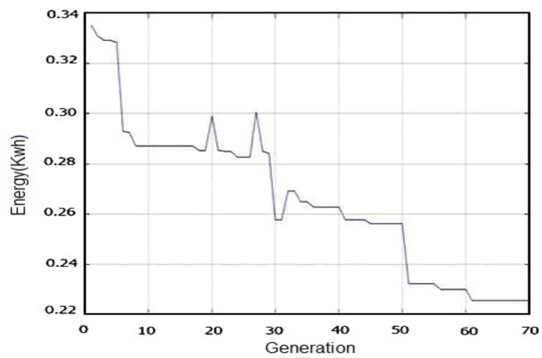


Fig. 3: Energy consumption with 70 iteration

Table 5: Energy consumption with PABFD in 10 iterations

Exp.	1	2	3	4	5	6	7	8	9	10
Energy	0.29kWh	0.32 kWh	0.42 kWh	0.38 kWh	0.28 kWh	0.43 kWh	0.31 kWh	0.34 kWh	0.28 kWh	0.32 kWh

4.2.2. Scenario 2: Without Employing Imperialistic Competitive and Genetic Algorithm

Power Aware Best Fit Decreasing (PABFD) is one of the most popular algorithms in the field of VMs' allocation to hosts and it is utilized by default in Cloud Sim simulator. In this algorithm, all the VMs are sorted in the decreasing order of their current CPU utilizations and allocate each VM to a host that provide the least increase of the power consumption caused by allocation. The algorithm is time-consuming and this feature counts as one of its disadvantages. Moreover, there is no point of view to other VMs at the time of a VM allocation.

Since PABFD is not an evolutionary algorithm, so iteration is meaningless for such an algorithm. Thus, to have a reliable comparison, we run PABFD 10 times and compare the collected report with the data achieved by 10 latter runs from the proposed algorithm. Table 5 shows the energy consumption for 10 runs in PABFD which shows different numbers because it is not a revolutionary algorithm. In this group of results, the energy consumption of the algorithm is evaluated without applying imperialistic competitive and genetic algorithm. It specifies how the ICGA improves the efficiency.

In Table 5, for energy consumption with PABFD in 10 iterations the optimum reported result is 0.28KWh. It is not an acceptable achievement compared with the combined imperialistic competitive and genetic algorithm.

Figure 4 illustrates a critical result of proposed method. This figure shows a comparison between PABFD and our proposed algorithm in terms of energy consumption. It is obvious that our proposed method considerably improves the energy efficiency than PABFD. ICGA improves the energy consumption around 19.39% than existing algorithm (PABFD).

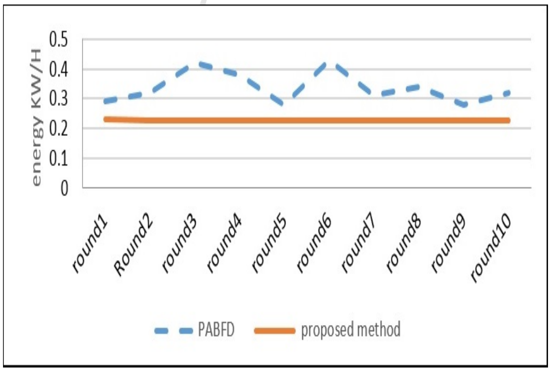


Fig. 4: Energy consumption comparison between the proposed method and PABFD

4.3. Changing the number of Virtual Machines

In this section, we evaluate the impact of different numbers of VMs on energy efficiency of the algorithm. In this regards, 10 tasks and 10 iterations are assumed. Figure 5 illustrates the energy consumption for 5, 10, 20 VMs.

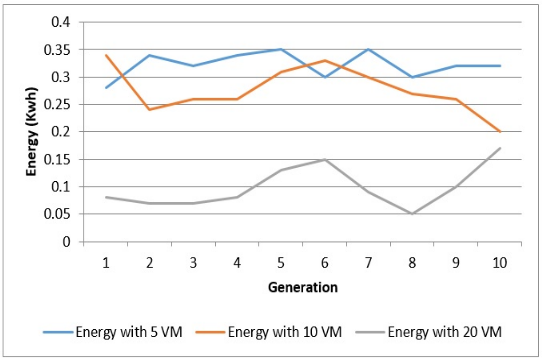


Fig. 5: Comparison between three different numbers of VMs in 10 iteration for 10 cloud let

The standard deviation is computed by the equation 1:

$$std = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_{(i)} - \mu)^2} \quad (1)$$

Where N is the iteration number, x_i is energy per run and μ is mean energy value for these 10 iterations. According to 5 VM numbers, standard deviation equals 0.0222. And with 20 VMs is 0.0415. For 5 VMs the standard deviation is near to zero rather than 20 VMs which shows results are close to the average with low dispersion. There is a crucial comparison between three different numbers of VMs and the equivalent energy consumption which is obvious in Figure 5. It is a significant result while the energy efficiency reduces by increasing the number of VMs. However, it is seen some fluctuations where the energy consumption reduces.

5. Conclusion and Future Work

This research focus on resource allocation of virtual machines to reduce energy consumption, and preventing of the environment pollution in cloud computing. We have employed crossover operator from genetic algorithm and apply it in the displacement step of imperialist competitive algorithm. Proposed algorithm named ICGA was simulated by Cloudsim software in Netbean environment with two different Scenario. This method has successfully achieved a significant improvement in terms of energy efficiency in comparison with PABFD algorithm. There are several future research line that can be taken in continues of this study such as quality service improvement and reducing required time in order to further improvement of cloud computing.

References

[1] J. Rezazadeh, K. Sandrasegaran and X. Kong, ”A location-based smart shopping system with IoT technology,” 2018 IEEE 4th World Forum on Internet of Things (WF-IoT), Singapore, 2018, pp. 748-753.

[2] M. Sattarian, J. Rezazadeh, R. Farahbakhsh, A. Bagheri,. ” Indoor navigation systems based on data mining techniques in internet of things: a survey,” Wireless Networks, 2018.

[3] B. Rajkumar, J. Broberg, and A. M. Goscinski,. ”Cloud computing: Principles and paradigms.” Vol. 87. John Wiley & Sons, 2010.

[4] M. Peter, T. Grance. ”The NIST definition of cloud computing.”, 2011.

[5] A. Farahzadi, P. Shams, J. Rezazadeh and R. Farahbakhsh, ”Middleware technologies for cloud of things-a survey,” in Digital Communications and Networks, 2017.

[6] Q. H. Cao and M. Giyyarpuram and R. Farahbakhsh and N. Crespi, ”Policy-based usage control for a trustworthy data sharing platform in smart cities,” in Future Generation Computer Systems, 2017.

[7] S. H. Sahraei, M. M. R. Kashani, J. Rezazadeh, R. Farahbakhsh, ”Efficient Job Scheduling in Cloud Computing Based on Genetic Algorithm,” International Journal of Communication Networks and Distributed Systems, 2018.

[8] A. Beloglazov and B. Rajkumar, ”Optimal Online Deterministic Algorithms and Adaptive Heuristics for Energy and Performance Efficient Dynamic Consolidation of Virtual Machines in Cloud Data Centers,” Wiley Inter Science, 2012.

[9] E. Atashpaz-Gargari and C. Lucas, ”Imperialist Competitive Algorithm: an Algorithm for Optimization Inspired by Imperialistic Competition. In Evolutionary Computation,” 2007.

[10] M. Madhusudhan, C. Sekaran, K. C ”A Genetic Algorithm Approach for Virtual Machine Placement In Cloude,” 2013.

[11] R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose and R. Buyya, ”CloudSim: a toolkit for modeling and simulation of cloud,” 2011.

[12] T. Goyal , A. Singh and A. Agrawal , ”Cloudsim: simulator for cloud computing infrastructure and modeling,” ELSEVEIR, vol. 38, pp. 3566-3572, 2012.

[13] R. Buyya , . R. Ranjan and R. N. Calheiros, “Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities,” IEEE, 2009.

[14] a. Gmach, J. Rolia, L. Cherkasov and A. Kemper, ”Resource Pool Management: Reactive versus Proactive or Lets be Friends,” Computer Networks: The International Journal of Computer and Telecommunications Networking, vol. 53, no. 17, pp. 2905-2922, 2009.

[15] S. S, K. A and Z. F, ”Energy Aware Consolidation for Cloud Computing,” in in USENIX Hot Power 05: Workshop on Power Aware Computing and Systems OSDI, 2005.

[16] Calheiros and R. N, ”CloudSim: a Toolkit for Modeling and Simulation of Cloud Computing Environments and Evaluation of Resource Provisioning Algorithms,” Software: Practice and Experience 41.1, pp. 23-50, 2011.

[17] R. Nathuji and K. Schwan, ”VirtualPower: Coordinated Power Management in Virtualized Enterprise Systems,” in the 21st ACM Symposium on Operating Systems Principles (SOSP), 2007.

[18] R. Raghavendra, P. Ranganathan, V. Talwar, Z. Wan and X. Zhu, ”No power struggles: Coordinated multi-level power management for the data center,” in ACM SIGARCH Computer Architecture News, 2008.

[19] D. Kusic, J. Kephart, J. Hanson, N. Kandasamy, ”Power and performance management of virtualized computing environments via lookahead control,” in IEEE International Conference on Autonomic Computing (ICAC), 2008.

[20] H. Taleb, S. Hamrioui, P. Lorenz and A. Bilami, ”Integration of energy aware WSNs in cloud computing using NDN approach,” 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN), 2017,

[21] J. Rezazadeh, M. Moradi, A. S. Ismail, and E. Dutkiewicz, ”Superior path planning mechanism for mobile beacon assisted localization in wireless sensor networks,” Sensors Journal, IEEE, vol. 14, pp. 3052-3064, 2014.

[22] J. Rezazadeh, M. Moradi, A. S. Ismail, and E. Dutkiewicz, ”Impact of static trajectories on localization in wireless sensor networks,” Wirel. Netw., vol. 21, no. 3, pp. 809-827, 2015.

[23] J. Rezazadeh, M. Moradi and A. S. Ismail, ”Message efficient localization in mobile wireless sensor networks,” Journal of Communication and Computer (JCC), 9(3), 340-344, 2012.

[24] J. Rezazadeh, M. Moradi and A. S. Ismail, ”Fundamental metrics for wireless sensor networks localization,” International Journal of Electrical and Computer Engineering (IJECE), 2(4), 452-455, 2012.

[25] M. Moradi, J. Rezazadeh, and A. S. Ismail, ”A reverse localization scheme for underwater acoustic sensor networks,” Sensors, vol. 12, no. 4, pp. 4352-4380, Mar. 2012.

[26] J. Rezazadeh, M. Moradi, and A. Ismail, ”Efficient localization via middle-node cooperation in wireless sensor networks,” in Proc. INECCE, Jun. 2011, pp. 410-415.

[27] J. Rezazadeh, M. Moradi, and A. S. Ismail, ”Mobile wireless sensor networks overview,” Int. J. Comput. Commun. Netw., vol. 2, no. 1, pp. 17-22, 2012.

[28] A. Verma, P. Ahuja and A. Neogi, ”pMapper: power and migration cost aware application placement in virtualized systems,” in the 9th ACM/IFIP/USENIX International Conference on Middleware, 2008.

[29] R. Asemi, E. Doostsadigh, M. Ahmadi and H. T. Malazi, ”Energy Efficieny in Virtual Machines Allocation for Cloud Data Centers Using the Imperialist Competitive Algorithm,” IEEE Fifth International Conference on Big Data and Cloud Computing, 2015.

[30] A. Abarghoei, E. Mahdipour and M. Askarzadeh, “Cloud Computing Resource Planning Based on Imperialist Competitive Algorithm,” Cumhuriyet Science Journal, vol. 36, pp. 1311-1324, 2015.

[31] A. Beloglazov, R. Buyya and Y. Choon, “A Taxonomy and Survey of Energy-Efficient Data Centers and Cloud Computing systems,” Advances in Computers, vol. 82, pp. 47-111, 2012.

[32] G. Portaluri, S. Giordano, D. Kliazovich, and B. Dorronsoro, ”A Power Efficient Genetic Algorithm for Resource Allocation in Cloud Computing DataCenters,” in IEEE 3rd International Conference on Cloud Networking, 2014.

[33] R. N. Calheiros, R. Ranjan, C. A. F. De Rose and R. Buyya, ”CloudSim: A Novel Framework for Modeling and Simulation of Cloud Computing Infrastructures and Services,” Grid Computing and Distributed Systems Laboratory, The University of Melbourne, 2009.

[34] M. R. Chowdhury, . M. R. Mahmud and R. M. Rahman, ”Implementation and performance analysis of various VM placement strategies in CloudSim,” Springer, 2015.

[35] A. Rahimi, L. Mohammad Khanli and S. Pashazad, ”Energy efficient virtual machine placement algorithm with balanced resource utilization based on priority of resources,” ComEngApp-Journal, vol. 4, 2015.

[36] A. Horri, M. S. Mozafari and G. Dastghaibyfard, ”Novel resource allocation algorithms to performance and energy efficiency in cloud computing,” Springer, vol. 69, no. 3, 2014.

[37] N. Quang-Hung, D.-K. Le, N. Thoai and N. Thanh Son, ”Heuristics for Energy-Aware VM Allocation in HPC Clouds,” Springer, vol. 8860, pp. 248-261, 2014.

[38] V. R and P. R, ”Measuring the Business Value of Converged Infrastructure in the Data Center,” in White Paper on Information Technology, IDC, 2011.