UNIVERSITY OF GLOUCESTERSHIRE

Usman, Mohammed Joda, Ismail, Abdul Samad, Chizari, Hassan ORCID logoORCID: https://orcid.org/0000-0002-6253-1822, Abdul-Salaam, Gaddafi, Usman, Ali Muhammad, Gital, Abdulsalam Yau, Kaiwartya, Omprakash and Aliyu, Ahmed (2019) Energy-efficient Virtual Machine Allocation Technique Using Flower Pollination Algorithm in Cloud Datacenter: A Panacea to Green Computing. Journal of Bionic Engineering, 16 (2). pp. 354-366. doi:10.1007/s42235-019-0030-7

Disclaimer

PLEASE SCROLL DOWN FOR TEXT.

Virtualization oriented Green Computing in Cloud Datacenter: Flower Pollination Approach

Usman Mohammed Joda[1][*], Abdul Samad Ismail[1], Hassan Chizari[2], Gaddafi Abdul-Salaam[3], Ali Muhammad Usman[4], Abdulsalam Yau Gital[5], Omprakash Kaiwartya[6], and Ahmed Aliyu[7]

1. *Department of Computer Science, Universiti Teknology Malaysia, 81310 Skudai Johor, Malaysia*
2. *School of Computing and Technology, Park Campus, University of Gloucestershire*
3. *Deparment of of Computer Science Kwame Nkrumah University of Science and Technology, Akara Ghana*
4. *Department of Maths and Computer Federal College of Education Technical Gombe, 072158 Gombe, Nigeria*
5. *Department of Maths, Abubakar Tafawa Balewa University Bauchi, 81027 Bauchi, Nigeria*
6. *Department of Computer and Information Technology Northumbria University, Newcastle, NE1 8ST, UK*
7. *Department of Maths, Bauchi State University Gadau, 81007 Bauchi, Nigeria*
Corresponding Email: *umjoda@gmail.com*

Abstract

Cloud computing has observed significant interest due to the increasing service demands from organizations offloading computationally intensive tasks to datacenters. Meanwhile, datacenter infrastructure comprises hardware resources consuming a high amount of energy and increasing carbon emissions at a hazardous level. In Cloud datacenter, Virtual Machine (VM) need to be allocated on various Physical Machines (PM) in order to minimize resource wastage and increase energy efficiency. Resource allocation problem is NP-hard, hence finding an exact solution is complicated especially for large-scale datacenters. In this context, this paper proposes an Energy-oriented Flower Pollination Algorithm (E-FPA) for VM allocation in Cloud datacenter environments. FPA is a Natured-Inspired optimization technique used in solving global and numerical optimization problems. A system framework was developed to enable energy-oriented allocation of various VMs on a PM. The allocation uses a strategy namely, Dynamic Switching Probability (DSP). The framework finds near optimal solution quickly and balances the exploration and exploitation of the global and local search. It considers a processor, storage, and memory constraints of a physical machine while prioritizing energy-oriented allocation for a set of virtual machines. Simulations are performed on MultiRecCloudSim utilizing planet workload. It is evident that the E-FPA outperforms state-of-the-art techniques in terms of energy consumption including Genetic Algorithm for Power-Aware (GAPA) by 21.8%, Order of Exchange Migration (OEM) ant colony system by 21.5%, and First Fit Decreasing (FFD) by 24.9%. This implies that, the datacenter performance and environmental sustainability has been improved significantly due reduction in energy consumption and as well carbon emission.

Keywords: Virtualization, Green computing, Cloud, Datacenter, Energy optimization

1 Introduction

Cloud Computing is a new paradigm that provides computing over the Internet on a pay-per-use basis, and its broader acceptance, coupled with the latest virtualization technologies, contribute to the

establishment of large-scale cloud datacenters to provide the computing services. Viewed from a long-term viewpoint, cloud computing may be perceived as alike to the evolution of the centralized distribution of electricity. The early electricity systems were firstly available on a rather small scale in the form of unconnected networks which slowly moved towards integration and centralization (Lang, 1969). Various services are offered by cloud datacenters to users at different levels.

Dynamic resource allocation consists of automating the allocation or de-allocation of resources in the datacenter without changing the system and or user running applications [1]. Fig 1 is a classification and model diagram of cloud computing showing the various services. Cloud services are categorized into Infrastructure as a Service (IaaS), Software as a service (SaaS), and Platform as a Service (PaaS) [2, 3]. The benefits of using cloud computing are many including pay-per-use, instant on-demand self-service provisioning, speedy elasticity, and resource sharing. Due to the benefits, there is increasing demand for cloud services by enterprises and the scientific applications, which intend calls for the expansion and or building new datacenters. Therefore, the concept of resource allocation (RA) has a meaningful impact on the operations of datacenters. Specifically, in pay-per-use deployments model, which include public, private, community and hybrid Cloud [4].
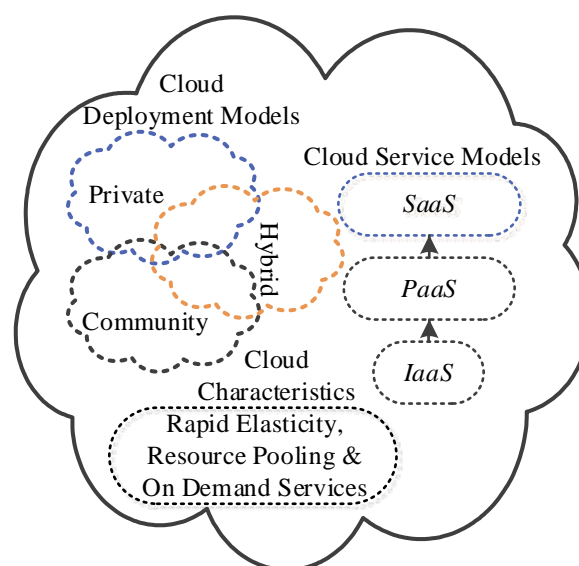

Fig. 1. Categorization and Model of Cloud Computing

However, over the years, the high energy consumption of these cloud datacenters have become a major concern as a result of increasing demands of resources and services by enterprise and scientific applications. Due to the large number of equipment contained in datacenters, enamors amount of energy is consumed leading to huge carbon emission [5]. Therefore, the high energy consumption has become a great concern to researchers. For example, Greenpeace (2010) claimed that the Cloud phenomenon may increase the problem of carbon emissions and global warming. The rationale given is that the aggregate demand for computing resources is anticipated to further grow dramatically in the next few years. The technological innovation aimed to reduce overall use of energy is directly related to cost, size and scale of datacenters [6]. Today's cloud datacenters contain thousands of physical and logical servers for hosting the internet, and other related services that cost millions of Dollars to power them. Even if we don't consider the financial aspect at present, energy efficiency becomes relevant in design and planning of setting up cloud datacenter [7]. In the datacenter, 75% of energy consumed is because of the linear energy consumed by the PMs [8]. Furthermore, due to the use of high-performance computing with integrated multi-core processors in the PMs, there exists power hunger and dissipation of considerable heat within the datacenter environment [9].

Furthermore, the datacenter energy consumption is proportionate to the resource utilization, beside its virtually considered as the world's largest consumers of electricity [10]. The inefficient usage of the IaaS, poor scheduling policies, and resource under-utilization that are causing the high energy consumption and not their size or low energy-efficiency of the hardware resources [11]. In fact, the utilization level of resources with their corresponding energy used by these datacenters is not trivial [12]. Another reason for this is because less utilized resources waste more energy than those utilized. In this regards, various resource management techniques that are considered to be energy-efficient using classical metaheuristics algorithms have been designed [13-15]. The techniques did not sufficiently prevent underutilization of resources causing the high energy consumption [4]. Similarly, [16-19] have also

proposed scheduling technique using metaheuristics algorithms to scale down the datacenter energy consumption, and resource utilization that includes other related service parameters. Therefore, energy-efficient resource allocation still remains an issue for cloud datacenter service providers. Cloud datacenters offer an abundance of resources, which makes the computing model maintain on-demand resource allocation, such abundance also leads to non-optimal allocation of resources on IaaS which cannot be optimally handled with the existing resource allocation techniques [20, 21]. However, resource optimization is NP-hard problem and all the solution proposed in the literature are based on soft computing method. Since optimality of results for NP-hard problem in soft computing is not provable, thus the focus of the proposed solutions is to optimize the methods to get better results.

Joseph, Chandrasekaran [22], Wu, Tang [23] and Wang, Wang [24] proposed VM placement using Genetic Algorithm (GA) to improve the convergence speed of the GA to produce global optimal solution by the cloud datacenter resource allocation strategy. Furthermore, Particle Swarm Optimization (PSO) algorithm has been explore by various researchers e.g., [25, 26]. Genetic Algorithm for Power-Aware (GAPA) has been developed to resolve the static VM allocation issue in order to improve the datacenter energy-efficiency [13]. Sharma and Reddy [15] proposed a hybrid technique that combined Dynamic Voltage Frequency Scaling (DVFS) with GA to reduce the datacenter energy consumption, increased resource utilization and convergence of the solutions. Alternatively, reducing energy consumption will be realized by turning off or switching PMs that are in the idle state to low-power mode state (i.e., sleep, hibernation) using Order Exchange and Migration (OEM) strategy with Ant Colony System (ACS) [33]. These techniques mapped VMs to PMs randomly and used the fitness evaluation function (objective function) to fit in VMs on PMs with a small number of running application. The jobs are processed based on arrival from 1 to n jobs. These techniques reduce the idle power consumption in the datacenters but has become complicated and difficult to manage due to the imbalance between local and global search of the algorithms which leads to inefficiency in allocating VMs on PMs that also leads

to energy resource wastage. However, these algorithms mostly focus on finding the initial global best solutions and focused only on one-dimensional resource. That is, CPU of PM and the computing requirement of VMs. In choosing a Nature-Inspired technique, there is the need to combine both global and local search methods to balance intensification and diversification. However, a larger solution search space does not always assure a superior optimal solution [21]. This shows the importance of striking a balance between local and global optima, which impacts on the quality of the allocation results. Furthermore, it has been observed in other works, FPA uses Differential Evolution (DE) algorithm to do a local search and also uses static Switching Probability to switch from local to global search space. Experiment results also showed that the local search ability of DE is somewhat limited (Yang and Deb, 2012). This implies that there is need to modify FPA since the central idea of this paper is reducing energy consumption and improving resource utilization in the datacenter. Therefore, the algorithm uses Dynamic Switching Probability (DSP) strategy to find the global optimal solution quickly which increases the convergence speed of the algorithm. Therefore, the adaptation of FPA optimization algorithm to address energy-efficient resource allocation and balancing between global and local search remain as challenging research issue in cloud datacenter environments.

In this context, this paper proposes an Energy-oriented Flower Pollination Algorithm (E-FPA) scheme for virtual machine allocation in cloud datacenter environments. The framework finds the energy oriented optimal solution quickly and balances the exploration and exploitation in the global and local search. The framework can be described majorly in four folds as contribution of the paper:

1) Firstly, an adapted flower pollination model is derived for green computing in cloud datacenter environments.

2) Secondly, a system framework is developed for reducing energy consumption in cloud datacenters focusing on user request model and E-FPA.

3) Thirdly, mathematical analysis of E-FPA is presented based on energy and resource utilization.

4) Finally, the framework is tested for comparative performance assessments with state-of-the-art techniques considering resource utilization and energy as a metrics within the cloud datacenter settings. The rest of this paper is organized as follows. Section 2 presents the detail of the proposed framework for energy-oriented green computing in cloud datacenter environments using FPA.   Simulation setting, and comparative analysis of assessment results are analyzed in section 3. Finally, section 4 present the conclusion followed by future research direction.

2 Energy oriented flower pollination scheme

This section provides the design and development of energy-oriented virtualization scheme using FPA. The scheme addresses the issue of high energy consumption and resource under-utilization due to the imbalance between local and global search which leads to premature convergence and inefficient resource allocation. The scheme contains the following: An overview of FPA, FPA based virtualization in cloud datacenter, and energy-oriented FPA based virtualization.

2.1 Overview of flower pollination modeling for green computing

Flower Pollination Algorithm (FPA) is among the state-of-the-art Nature-Inspired algorithm inspired by the analogy of biological process of pollination [35]. The FPA witnessed significant applications in engineering Ochoa et al. (2014) and various research domain (Abdel-Raouf and Abdel-Baset, 2014; Platt, 2014, Wang and Zhou, 2014). compare to PSO, ACO, CSO, GA, NSGA II and CSA. The performance and effectiveness of FPA are verified using some widely used benchmark problems. The results support its applicability in solving optimization tasks [36]. The summary obtained from FPA evolutionary line, it shows that the algorithm has the readiness, flexibility, capability, and efficiency of being adaptable to solve different types of problems in different NP-hard situations.

Similarly, many researcher have implemented FPA to resolve the NP-hard problems either in continuous or discrete search spaces and found to outperform the compared metaheuristics algorithms [37-39]. The E-FPA deals with the selection of population size (N) and a parameter (P) which help to select the

value of self-pollination and cross-pollination to take place. The algorithm proceeds by initializing the defined number of population (*N*), with each one carrying a group of variables which are optimized using the objective function. This algorithm incorporates an indexing term called flower constancy for each population which determines how well their variables minimize the objective function. Based on the flower constancy, the population is queued, and best among them is found as describe in algorithm 1.

1) The cross-pollination which is also called biotic pollination is responsible for carrying-pollen to pollinators performing Levy flights movement. The Levy flights is one of the stable distribution which is important in the study of Brownian motion and named after the French mathematician Paul Levy (Borodin and Salminen, 2012). This movement is considered as a global pollination method.

2) The term abiotic and or self-pollination describes the local pollination method.

3) The mean of flower constancy is regarded as probability of reproduction system of the flower that is directly associated to other distinct flowers.

4) Switching probability is employed to control between exploration and exploitation that are commonly known as local and global search. It is defined as *p* [0, 1].

The rules above are expressed by three main attributes such as global search, local search, and the switching probability respectively. In FPA, the pollination takes place between two classes except for the fittest function, switching probability, and the levy flight. One of these classes is called the global solution and or global search. In this class, every flower receives single pollen and individual flower drops single pollen gamete only. Consequently, a solution x_i is equal to a flower and at the same time a pollen. The pollinator's searches for a solution within the search space to locate the current position of the optimum solution. Therefore, global optimization conforms to the biotic method and or cross-pollination that moves pollen from one location to another different location conforming levy flight law. Levy flight can be express as in Eq. 1.

$$X_i^t = X_i^t + L\left(g_* - X_i^t\right) \qquad (1)$$

Here, X_i^t denotes the ith solution produced by the pollen at the *t* time of iteration. The present solution

found within all possible solutions is the best solution and is denoted by g_* during the iteration. *L* is also

a constraint value indicating movement and scope of the Levy distribution represented as Eq. 2.

$$L \sim \frac{(\beta+1)*sin\left(\frac{\pi\beta}{2}\right)}{\pi} * \frac{1}{s^{1+\beta}}, (s \gg s0 > 0) \qquad (2)$$

Here β is the standard gamma function while the Levy distribution is valid for large steps *s* > 0. The

second class which is the local search is the composition of flowers after getting the global optimum to

find a more optimal solution in the neighborhood. FPA takes local pollinators for searching of a solution

within search space due to their effectiveness to locate and obtained a better solution. This class nor-

mally finds an improved solution from the current set of solutions evaluated by the objective function.

Mathematically it is represented as in Eq. 3.

$$X_i^t = X_i^t + \in (X_j^t - X_k^t) \qquad (3)$$

where, X_j^t and X_k^t are the pollens of flowers that are alike, but they belong to different species of flowers.

$\in$ is assumed to be a uniform distribution in [0,1] which become a random walk.

2.2 Integration of flower Pollination for resource allocation in cloud computing datacenter

Resource allocation problem is an NP-hard problem. Due to the NP-hard nature of the problems,
heuristic algorithms cannot effectively obtain global optimum solutions. Therefore, this research
adapted FPA in order to reduce the datacenters challenges in respect to energy and resource utilization.
The representation of the FPA in cloud datacenter is shown in Fig 3 which consist of two layers. Each
pollen is represented as n-dimensional vector of *VM* resources in the *VM* layer and each vector is rep-
resented as $p(ij), 1 \leq i, j \leq n$ which consist of random values in the range of [0,1] such that $0 < p(ij) < 1$.
The PM layer represent the flower which is a set of PMs where the VMs are placed. The pollen agents
serve as the mapping function between *VMs* and *PMs*. The flowers value for mapping $VM(n)$ on $PM(m)$
is associated with active *PM* within the datacenter. After completion, a unique mapping for each *VM*
correspond to the *PM* for each pollen and flower using the pollen agents that is based on resource op-
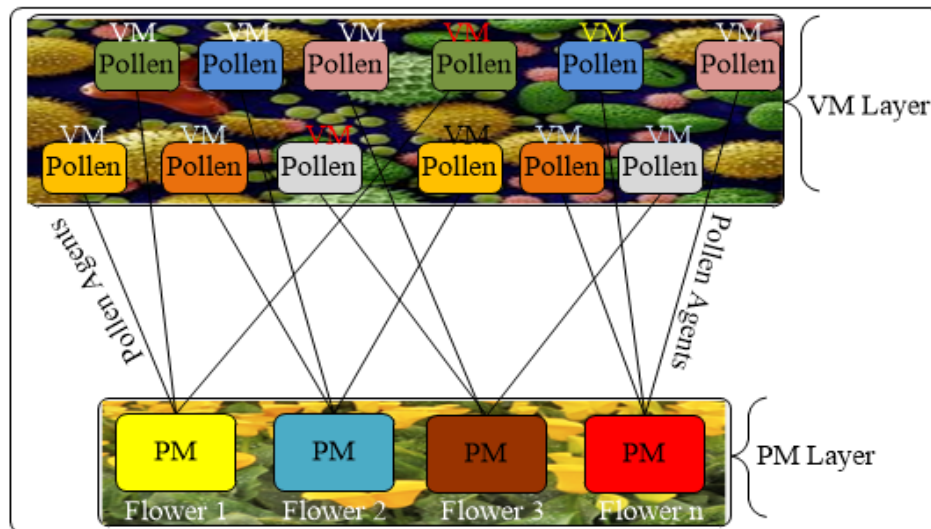timization.

Fig 3. Representation of Flower Pollination in datacenter

2.3 Energy oriented virtualization in cloud computing datacenter

The system architecture of resource allocation in cloud environment assigns available resources that are accessible to the various mode. These modes are viewed as a plan for provisioning in datacenter environment using different methods and or schemes as discussed in the introduction section. The methods have not solved the problem of inefficiency of the resource allocation policy that resulted in-adequate resource utilization and energy management in cloud datacenters. However, our method has taken into consideration the problem above and introduces E-FPA a new optimization technique for cloud datacenter to solve the problem. Fig 4 illustrates the components for the proposed energy-efficient resource allocation which is composed of three main entities: Cloud users, service providers, and the datacenter resource management. It shows how the user request is handled by the broker to the Cloud and finally to the datacenter for optimizing VM allocation.

The user's request is submitted to the Cloud service provider first; then the broker will return the result to the user based on the need, date-line, resource service operation, and capacity/performance management of the available datacenters that they subscribe. When the broker's request reaches the datacenter, the

cloud information system (CIS) resource manager then looks at the request, compares it with the pool of

available resources and then decides. The CIS's acceptance of any request is based on the system

availability. Upon accepting the request, CIS passes it to the scheme for allocation to find global optimal

solution. The solution is passed to the resource module which is E-FPA for initial VM placement and

monitoring. After which it is placed with the utilized resource that is energy-efficient. In the next stage,

the VM manager and scheduler module will identify whether the heterogeneous VMs provide the

characteristics of the resource requirement such as reservation, on demand, availability, and allocation.

In the following section, we described the user request model, resource mapping and energy models that

are applied in realizing energy-efficient resource allocation of cloud datacenter.
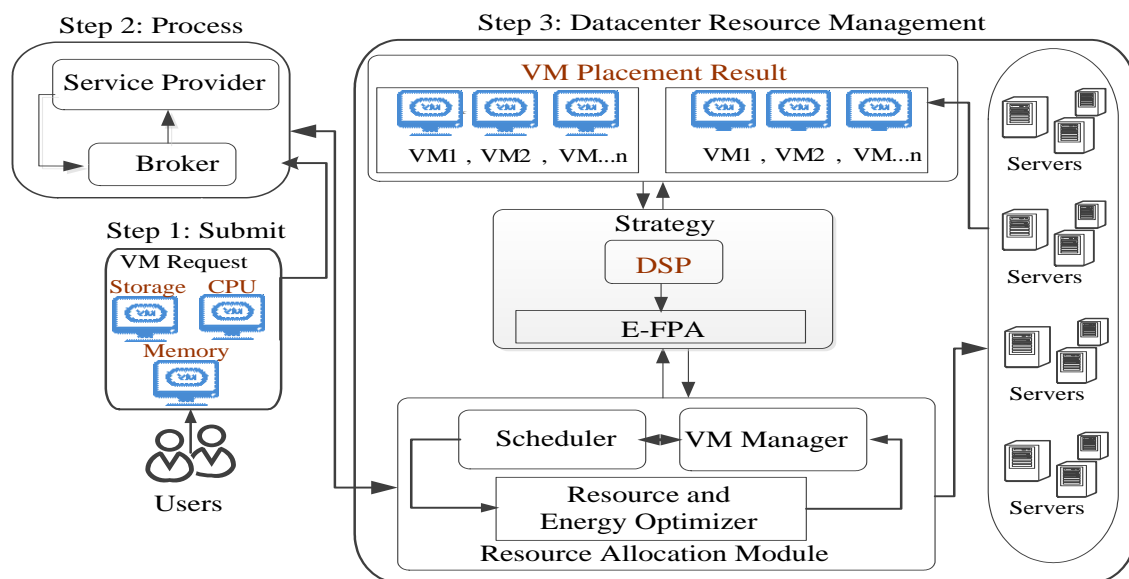

Fig 4. Energy oriented virtualization system components

2.3.1 User request model

The user request resources of the datacenter through the broker or cloud provider for their various

application needs. The user requests a set of resources known as VMs. Each of the requested resource

(VMs) have their required components of performing a task. We denote user request as UR which is the

subject. The users send one or many requests at a time which are UR $(A_i, ..., A_n)$ for $i = 1,2,3,...,n$ which are executed based on a First-Come-First-Serve (FCFS). A_i is the components of the VMs. The resource components of VMs include α_s^i as CPU, β_s^i as Memory and γ_s^i as Storage. The corresponding i and s represent the number of resources and their measuring capacity, respectively. Mathematically, we can represent the request as:

$$Ai \subset UR \text{ and } \alpha_s^1, \beta_s^1, \gamma_s^1 \subset Ai$$

$$\alpha_s^1, \beta_s^1, \gamma_s^1 \subset Ai \subset UR \Rightarrow \alpha_s^1, \beta_s^1, \gamma_s^1 \subset UR$$

Therefore, when the user sends a request for only one resource, it will be expressed as in Eq. 4 and 5.

$$UR^1 = A_i \qquad (4)$$

where $i = 1$, and UR^1 is when the resource required is only one.

$$Ai = (\alpha_s^1 + \beta_s^1 + \gamma_s^1) \qquad (5)$$

On the other hand, if the user request is more than one, then the request is expressed as in Eq. 6 and 7.

$$UR^n = \sum_{i=1}^n = Ai = A_1 + A_2 + A_3 + ...A_n \qquad (6)$$

$$= (\alpha_s^1 + \beta_s^1 + \gamma_s^1) + (\alpha_s^2 + \beta_s^2 + \gamma_s^2) + ... ((\alpha_s^n + \beta_s^n + \gamma_s^n))$$

$$UR^n = \sum_{i=1}^n (\alpha_s^i) + \sum_{i=1}^n (\beta_s^i) + \sum_{i=1}^n (\gamma_s^i) \qquad (7)$$

2.3.2 Energy and resource utilization model

Given a set of $VM = \{vm_i \backslash i = 1,2,..., n\}$ to be allocated on a set of PMs (Servers) $PM = \{PM_j \backslash j = 1,2,..., m\}$. Each VM is denoted as a d-dimensional vector of demand resources [40], i.e. $VM_i = (A_{i,1}, A_{i,2} ..., A_{i,d})$. Similarly, each PM is denoted as a d-dimensional vector of capacity resources [41], i.e. $PM_j = (B_{j,1}, B_{j,2} ..., B_{j,d})$. The various PM_j resources considered in this study are 3 including processor (CPU), physical memory (RAM), and storage. Hence, the dimension $d = 3$ [40]. Furthermore, the VM allocation has a starting and stopping time, i.e., each VM_i is started at a fixed time

(S_{t_i}) and execution time (E_{t_i}). Therefore, the overall time expended during the allocation of vm_i is mathematically represented as $S_{t_i} + E_{t_i}$. where $A_{i,s}$ is resource capacity $(\alpha_s^1, \beta_s^1, \gamma_s^1)$ requested by the VM_i $(1, 2, ..., n)$ and $B_{i,s}$ resource capacity $(\alpha_s^1, \beta_s^1, \gamma_s^1)$ of the PM_j $(1, 2, ..., m)$. The resource allocation problem has the following (hard) Requirements:

1) Requirements 1: Resource most be compatible with the request

2) Requirements 2: $\forall VMs$ request is $\leq$ the PMs' total resource capacity.

3) Requirements 3: $\forall VMs$ each VM is run by a PM at any given time.

4) Requirements 4: Assume that $a_j(t)$ is the set of VMs that are assigned to a PM.

5) $\sum_{i=1}^n UR^n$ of these assigned VMs is $\leq$ the PMs' total resource capacity.

$$For\ all\ capacity\ (\forall s) = 1, ..., d : \sum_{vm_i \in a_j(t)} A_{i,s} \leq B_{i,s} \tag{10}$$

A feasible resource allocation R^A implies a successful representation of VMs to PMs, i.e., $\forall i \in \{1, 2, ..., n\}$, $\exists j \in \{1, 2, ..., m\}$, allocated (VM_i, PM_j) holds when VM_i is assigned to physical machine PM_j. Therefore, the objective function of R^A is to maximize resource utilization (RU^{DC}) and energy efficiency (DC_U^{Energy}) of the cloud datacenter. Firstly, maximization of RU of PM_j is considered as denoted in Eq. 11.

$$RU_j^d = \frac{\sum_{i=1}^n \rho_{ij} * VM_i^d}{PM_j^d} \ \forall d \in \{\alpha_s^1, \beta_s^1, \gamma_s^1\} \tag{11}$$

To achieve the total RU of the datacenter, the individual PM_j resources are integrated and formulated as thus:

$$RU^{DC} = \int_{t1}^{t2} \frac{\sum_{j=1}^m U_j^{\alpha_s^1} + \sum_{j=1}^m U_j^{\beta_s^1} + \sum_{j=1}^m U_j^{\gamma_s^1}}{|d| \sum_{j=1}^m \rho_j^i} \ \partial t \tag{12}$$

$$\begin{cases} 1 \text{ if } VM_i \text{ assigned to } PM_j \\ 0 \text{ else} \end{cases}$$

We presume that each PM_j can host any VM_i and the energy consumption model $P_j(t)$ of the host PM_j has a linear relationship with resource utilization (the higher the utilization the higher the energy consumed by the PM) [12]. Lin, Xu [42] use the same model with different resource energy consumption as presented in Eq. 13-15. Table 1 presents energy consumption model of HPG4 and HPG5 servers at various level of utilization. The model for the datacenter PM resource energy consumption is given as follows:

$$TEC_j^\alpha = \int_{t1}^{t2} P\big(EU(t)\big)dt \; PM_j \in P \qquad (13)$$

where $EU(t)$ represents the PM utilization at the given time t and $P\left(EU(t)\right)$ represents the power consumption related to $EU(t)$.

$$\text{Maximize } \sum_{j=1}^m = EU(t)_j \qquad (14)$$

$$TEC_j^\alpha = \sum_{j=1}^m = EU(t)_j \qquad (15)$$

where $EU(t)_j \; with \; j = 1,2,\dots,m$ is the total energy consumption of the PM_j.

$i \in \{1,2,\dots,n\}, \; j \in \{1,2,\dots,m\}, \; t \in [0;T]$.

The overall energy consumption for the datacenter can be expressed as presented in Eq. 16.

$$DC_U^{Energy} = \int_{t1}^{t2} P\big(EU(t)\big)(\alpha)dt + EU(\beta) * P_{max} + EU(\gamma) * P_{max} \qquad (16)$$

Consequently, the efficiency of R^A which maximizes resource utilization (RU) and energy efficiency of the Cloud datacenter can be mathematically formulated as follows in Eq. 17.

$$\text{Maximize} \Rightarrow RU_j^d = \frac{\sum_{i}^{n} = \rho_{ij} * VM_i^d}{PM_j^d} \; \forall d \in \{\alpha_s^1, \beta_s^1, \gamma_s^1\} \; \text{s.t. Minimizes} \; \sum_{j=1}^{m} = EU(t)_j \qquad (17)$$

Table 1: Energy consumption by PMs at different load level

PM	0%	10%	20%	30%	40%	50%	60%	70%	80%	90%	100%
HP ProG4	86	89.4	92.6	96	99.5	102	102	106	108	114	117
HP ProG5	93.7	97	101	105	110	116	121	125	129	133	135

2.3.3 Energy oriented flower pollination algorithm

Due to the heterogeneous and large size nature of IaaS cloud and resource management requirement, it is practically impossible to apply FPA directly for resource allocation problem on IaaS cloud due to the large solution space which may take a long time to find an optimal solution. Thus, a new strategy of search operators based on the problem features needs to be re-designed which include the switching probability. Furthermore, it has been observed in other works, FPA uses Differential Evolution (DE) algorithm to do a local search and uses Static Switching Probability to switch from local to global search space. Experiment results also showed that the local search ability of DE is somewhat limited (Yang and Deb, 2012). This implies that there is need to modify FPA since the central idea of the scheme is reducing energy consumption and improving resource utilization in the datacenter. Therefore, the scheme uses Dynamic Switching Probability (DSP) strategy to find the global optimal solution quickly and to increase the convergence speed of the scheme. However, FPA was found to be better compare with other existing resource allocation algorithms and require improvement to meet up with the current increasing number of concurrent users in the cloud datacenter environment. Furthermore, E-FPA requires pollinators at the local search step for efficient search and exploration of solutions within the searching area define by the algorithm. The proposed scheme with the modification is shown in Fig 5. The first step that is performed by the scheme is to find an improved solution from the current solution of the objective function. Any solution that satisfies the constraint of the objective function is considered to be the feasible solution. For example, when using evolutionary algorithm such as GA, candidate solution is de-

signed with a corresponding string of registers popularly identified as a chromosome. Next, every simulation step eliminates the 'n' worst scenario solution that have been created and formed a new 'n' set of candidate solutions from the best-case scenario of the generated solutions. For every generated solution, there must be a distinctive value that indicates how the solution meets the overall requirement. The goal of resource allocation in the cloud datacenter environment is to allocate the *n* request to the *m* available resources to execute the user request with less resource. The details of the algorithm and its implementation are described in the following subsections.
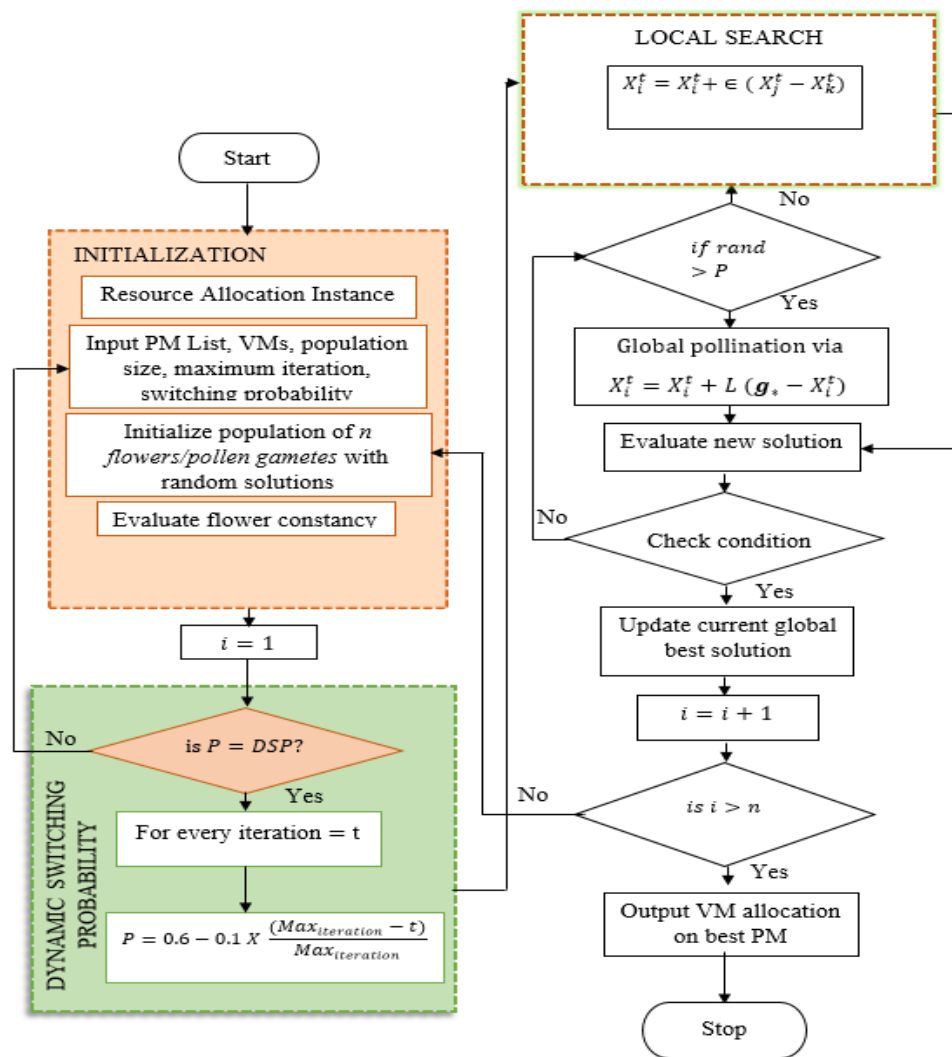

Fig. 5: Operational workflow of the proposed scheme for green computing

3 Implementation of energy oriented flower pollination algorithm

The details of the proposed scheme and its implementation are described in the following section.

3.1 Initialization phase

The first phase of the scheme is the initialization with the aim to find the feasible solution from the current solutions of the objective function, and any solution that satisfies the constraint of the objective function it's considered to be the best solution. The work uses energy-aware objective function instead of random initialization that reduces the effectiveness of the energy consumption of the optimization. Similar to GA-based optimization method where each solution is designed with a corresponding string of registers popularly known as a chromosome, in this phase each step of a simulation eliminates the n worst scenario solutions that have been created and to form new n set of solutions from the best-case scenario of the generated solutions. For every generated solution, there must be a distinctive value that indicates how the solution meets the overall requirement. If the requirements are not satisfied the algorithm will proceed to the next step. The whole process is iterated until the given stopping criteria are met. The solution is represented by the best flower in the final population. Algorithm 1, shows how the objective function R^A calculates the evaluation value of each pollen defined for resource i and UR^n that aim to reduce the datacenter energy consumption as presented in Eq. 17.

Algorithm 1 *Objective Function*

Require: Total Energy Consumption of Datacenter *PMs*
Ensure: Allocate *VM on PM* with efficient energy based on utilization of CPU, Memory, and Storage

1:   **For** each *PM* $\in$ collection of *PMs* **do**
2:   *Utilization of PM:=PM.getUtilization of PM* (CPU; Memory; Storage)
3:   *Power of PM:=getPower (PM.getUtilization of CPU)*
4:   *Power of PM:=getPower (PM.getUtilization of Memory)*
5:   *Power of PM:=getPower (PM.getUtilization of Storage)*
6:   *Energy of Datacenter:= Energy of Datacenter + power of PMs* (CPU; Memory; Storage)
7:   **End for**
8:   Evaluation Value (Pollen):= 1.0/ power of Datacenter

3.2 Global search strategy phase

In this phase, it is assumed that each plant contains only one flower, and each flower produces one pollen gamete. Hence a solution X_i is equivalent to a flower and a pollen gamete. Pollinators need to search the whole search space to find the location of the optimum point. Hence, global optimization adapts the biotic and cross-pollinators to play their role more perfectly as they can travel a long distance obeying levy flight rule. Levy flight is more efficient than Brownian in exploring unknown large-scale search space, and this can be express as in Eq. 2.

3.3 Local search strategy phase

The local search phase is the composition of solutions based on Flower Pollination Algorithm after obtaining global optimum solution. The algorithm intensifies the exploitation to find a more optimal solution within the neighborhood structure. FPA needs local pollination for exploitation as they can better exploit the area at which optimum value lies. This phase finds an improved solution from the current solution of the population size as represented in Eq. 3.

3.4 Dynamic switching probability phase

The switching probability P is used to switch between the global pollination and the local pollination in the FPA and the P is always constant. It is assumed that an algorithm should do a more global search at the beginning of the searching process and global search should be less at the end [43]. Therefore, Dynamic Switching Probability (DSP) strategy has been added to adjust the proportion of two kinds of searching process, to balance the local and the global search exploitation and exploration. The enhancement of the switching probability p has been modified according to Eq 18.

$$P = 0.6 - 0.1 \times \frac{(Max_{iteration} - t)}{Max_{iteration}} \qquad (18)$$

where $Max_{iteration}$ is the maximum iterations of the proposed scheme and t is current iteration time. Special implementation measures of E-FPA including DSP were presented in Algorithm 2 together with the pseudocode.

Algorithm 2 *Enhanced Flower Pollination Algorithm*
Require: Set of population of *n flowers/pollen gametes* with random solutions
Find the best solution $g*$ in the initial population
Ensure: Define a switch probability $P = 0.6 - 0.1 \, X \, \frac{(Max_{iteration} - t)}{Max_{iteration}}$

1:	*Input: PM list, VM, set of parameters*
2:	*Output: VM allocation*
3:	*Execute*: Objective *Max* $R^A = (RU^{DC})(DC_U^{Energy})$ // Equation (4.9)
4:	*Initialize a population of n flowers/pollen gametes with random solutions*
5:	*Find the best solution g_* in the initial population*
6:	Define a switch probability P using Equation (4.13)
7:	**While** (t $< MaxGeneration$)
8:	**For** $i = 1{:}n \ (all \ n \ flowers \ in \ the \ population)$
9:	**If** $rand < p,$
10:	*Draw a (d- dimensional) step vector L which obeys a Levy distribution* $Global \ pollination \ via \ X_i^t = X_i^t + L \ (g_* - X_i^t)$
11:	**else**
12:	$Draw \in from \ a \ uniform \ distribution \ in \ [0,1]$
13:	$Randomly \ choose \ j \ and \ k \ among \ all \ the \ solutions$
14:	$Do \ local \ pollination \ via \ X_{i+1,G} = X_{i,G} + \emptyset \left(X_{n-best_i G}\right) + \quad \vartheta \left(X_{p,G} - X_{q,G}\right)$ $where \ \emptyset = \ \vartheta = \in;$
15:	**end if**
16:	*Evaluate new solutions*
17:	*If new solutions are better,* **Then**
18:	*Update them in the population*
19:	**end for**
20:	*Find the current best solution g_**
21:	**End while**

3.5 Mathematical analysis of E-FPA

The E-FPA is first verified using the benchmarking function proposed by Jamil and Yang [44] and Wang and Zhou [45] which are important to obtain the performance of the optimization algorithm. Table 2 presents the results achieved by the functions used to mathematically compare E-FPA with FPA and ACS using Eq. 19-23.

1) Sphere function $\qquad\qquad\qquad\qquad\qquad\qquad f_1(x) = \sum_{i=1}^{n} x_i^2$ $\qquad\qquad$ (19)

2) Rosenbrock function $\qquad\qquad f_2(x) = \sum_{i=1}^{n}(100 \times (x_{i+1}^2 - x_i^2)^2 + (1-x_i)^2)$ $\qquad$ (20)

3) Cube function $\qquad$ $f_6(x) = 100 \times (x_{i+1}^2 - x_i^3)^2 + (1 + x_i)^2)$ $\qquad$ (21)

4) Chunk function $\qquad$ $f_5(x) = (\sum_{i=1}^n x_i^2)^2$ $\qquad$ (22)

5) Rastrigin function $\qquad$ $f_3(x) = \sum_{i=1}^n [x_i^2 - 10 \times cos2\pi x_i + 10]$ $\qquad$ (23)

Each of the algorithm were examined in various conditions, i.e., adjusting the value of recursive iteration (100, 200, 1000), keeping the iteration size fixed 40, turning the population size (20, 50, 60), and keeping the number of iteration constant 1000, and using a population size n = 25 and p = 0.8 for FPA, crossover probability 0.95 [35, 45], and learning parameters 2 for ACS [33]. The result is analyzed based on their performance regarding maximum, average, and standard deviation.

The above algorithm has been run 20 times for each of the above-mentioned settings on a benchmark function, and the conclusive outcome or results are taken from the average of 20 times running of the experiment. This has reduced the impact of the error rate from the experiment. Since in E-FPA the most optimist pollen of the flower can only pass on the information between local and global search using the DSP strategy. As a result, the algorithm converges with high speed against the FPA and ACS. In Fig 6, we plot the convergence of the E-FPA (considering total number of iterations it took to attain the global optimum solution) by changing size of the population from 10 to 50. It has been observed from the experiments, the proposed algorithm converges to global solution fast, retaining a linear link with rising number of population sizes of 10 and 50. Hence, we have selected 50 as size of the population for the conducted experiment. In the proposed scheme, each flower is modelled as M-dimensional vector. The search space of the flowers is limited to *I* and *I* is also given in the population size. We observed that the proposed scheme outperforms FPA and ACS. This is because the proposed schemes use DSP strategy that enhances its efficiency. ACS gives the guarantee of convergence, however, the time taken to converge is undefined due to the series of random and casual decisions by the overall scheme while running the experiment. The graph reveals the proposed scheme superiority that solve complex placement problem in cloud computing environment. The average and standard deviation of E-FPA are greater than

FPA and ACS. Therefore, E-FPA presents better results in all the function. The principal feature that ensures the high performance is the introduction of DSP strategy that starts the local search procedure from a feasible solution. Furthermore, this allows the E-FPA find solutions in shorter time.
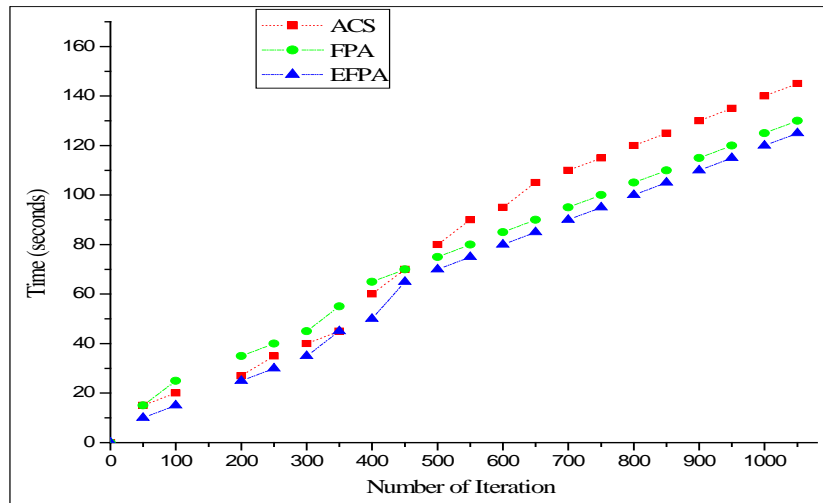

Fig 6. Convergence and computational performance

Table 2: Benchmark Functions Comparison Results

Function	Performance	E-FPA	FPA	ACS
Sphere $f_1(x)$	Maximum	3.97E-04	1.55E-04	9.54E-56
	Minimum	2.63E-04	1.70E-04	2.60E-43
	Average	5.78E+04	1.90E-04	7.87E-51
	Standard Deviation	8.61E+04	3.18E-04	8.22E-44
Rosenbrock $f_2(x)$	Maximum	3.97E-04	1.55E-04	1.32E-03
	Minimum	4.04E-04	2.05E-04	4.64E+00
	Average	2.73E-04	1.24E-04	2.08E+01
	Standard Deviation	2.14E+04	1.50E-04	1.71E+00
Cube $f_6(x)$	Maximum	5.78E+00	2.32E-04	1.03E+00
	Minimum	3.18E+00	1.62E-04	5.13E+00
	Average	2.26E-04	1.18E-04	2.47E+00
	Standard Deviation	8.01E+04	2.22E-04	1.32E+00
Chunk $f_5(x)$	Maximum	5.15E-04	2.52E-04	1.03E-02
	Minimum	7.96+-04	3.97E-04	3.77E-02
	Average	9.71E+04	5.15E-04	1.90E-02
	Standard Deviation	8.38E+04	2.43E-04	9.03E-03
Rastrigin $f_3(x)$	Maximum	5.78E-04	1.90E-00	1.16E+00
	Minimum	2.63E-04	1.70E+04	8.76E+00
	Average	3.97E+04	1.55E+04	3.50E+00
	Standard Deviation	4.04E+04	2.05E+04	1.98E-00

4 Evaluation method and result analysis

4.1 Simulation setting

The MulRecCloudsim 3.0.4 is run with the IntelliJ IDEA release version 3.4.0. The schemes are implemented on an Intel CoreTM i7 processor, 2GHz processor speed, 1 terabyte hard disc drive and 8 gigabyte memory. Table 3 shows the parameter settings for the PM and VM used in the experiment. Throughout the simulations time, each VM is assigned a workload randomly trace based on the user request using the same parameter as in [35]. For the sake of simplicity, the PMs are considered to be homogeneous, though heterogeneous configuration can also be simulated. The user request ranges from 1-100 at a time, and the E-FPA is applied at the datacenter on the arrival of a new request from the user. To show the effectiveness of our proposed scheme the (RU^{DC}) of (PMs) and DC_U^{Energy} of the data-center are calculated by Eq. 10 and Eq. 17 respectively. The used DSP strategy in the proposed scheme results in global optimum solutions for allocating VMs to PMs which minimizes the datacenter energy consumption. The results of the proposed scheme are compared with Genetic Algorithm for Power-Aware (GAPA) [13] Order Exchange Migration (OEM) ACS [33], and First Fit Decreasing (FFD) [2]. These works have considered resource utilization, number of active PMs and as well their energy consumption. They are clearer and related to the problem we are solving. Furthermore, other researchers have used them in order to compare their work with the same parameter.

Table 3: PM and VM Parameters setting

Cloud Entity	Parameter	Value
Datacenter	Number	1
PM	RAM	2048000 MB
	Disk	10000000 MB
	Operating System	Linux
	Bandwidth	1000000000 MB
	Architecture	x86
	VM Manager	Xen
	CPU Power Model	PowerMod-elSpecPowerX3550XeonX5675
	Storage Power Model	PowerModelStorageSimple
	Memory Power Model	PowerModelMemorySimple
VM	RAM	2048000 MB
	Bandwidth	0.1GB/s
	MIPS	367 MHz
	Storage	1000000 MB

4.3 Workload type

Experiments were conducted via real data from PlanetLab. The data contains more than a thousand servers with their corresponding components utilizations. The workload consists of 5 days of data as shown in Table 4(a) with several resource demand obtained from the CoMon monitoring project [46]. Datacenter workloads are infrastructure representative in Cloud environment. The data traces of the PlanetLab are accessible and copiously working in CloudSim. Similarly, Amazon EC2 instances has been used in the experiment Table 4(b) shows the four kinds of typical (M3) VM instances suggested by Amazon EC2 considering CPU, memory, and storage as type 1, type 2, and type 3 resources, respectively. For example, request (10; 0; 0; 5; 2;) represents a user requesting 10 C3.medium VM instances, 0 C3.large VM instance, 0 C3.xlarge VM instance, C3.2xlarge VM instances and 2 C3.4xlarge VM instances.

Table 4. Simulation parameters: (a) planet workload data for 5 days, (b) General purpose (C3) VM instance types offered by Amazon EC2

(a)

Data	No. of VMs	Date	Number of PMs
Workload 1	1085	03/03	800
Workload 2	896	06/03	800
Workload 3	1061	09/03	800
Workload 4	1516	22/03	800
Workload 5	1078	25/03	800

(b)

Name	CPU	Memory (GB)	Storage (GB)
C3.medium	2	3.75	2 x 8
C3.large	4	7.5	2 x 16
C3.x large	8	15	2 x 40
C3.2xlarge	16	30	2 x 80
C3.4xlarge	32	60	2 x 160

4.3 Resource utilization analysis

Fig 5 (a) represents the average resource utilization of CPU, memory, and storages of all the PMs in the datacenter. The total of active PMs inside the datacenter infrastructure is also analyzed. The result shows how the resource utilization is affected as the number of PM increases as depicted in Fig 5(b). The outcomes of results reveal that the E-FPA allocation achieves 95.4% average resource utilization of datacenter as can be seen in Fig 5(c). Whereas in the case of GAPA, OEMACS, and FFD achieve less than 72% average resource utilization of datacenter causing net growth of 23.9% increase in IaaS resources utilization. This improvement in average utilization of PMs is due to the incorporation of DSP strategy that stops the local search from searching specific areas of the search space, thereby making the scheme to explore the neighboring solution. Another reason is the use of DSP strategy that improves the global convergence of E-FPA. The proposed scheme allocates VM on the targeted optimal PM. Overall, the above revealing results justify the benefit of incorporating the DSP strategy in the proposed scheme. This proves that E-FPA is an operational, successful and efficient solution for solving large-scale resource allocation optimization problems.
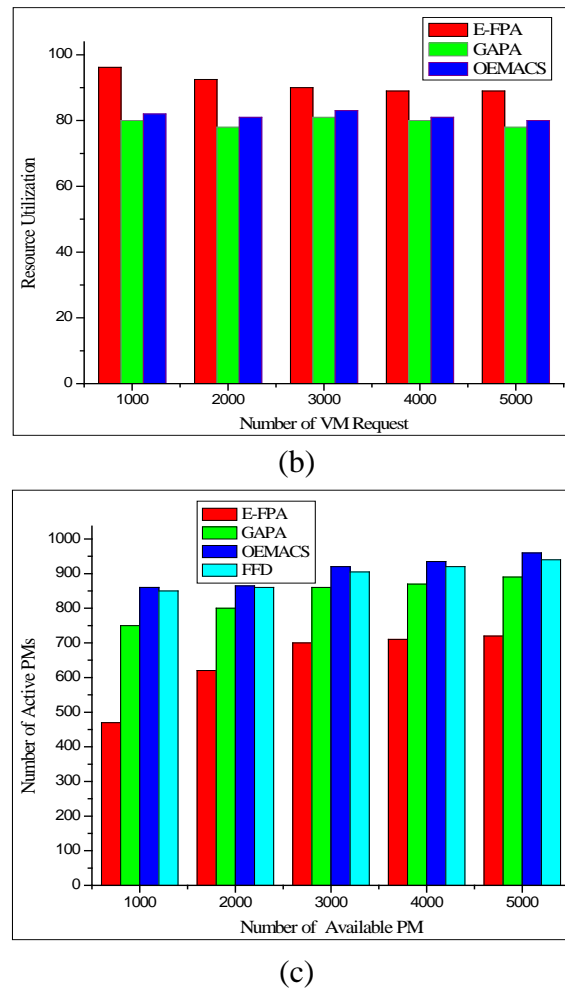

(a)

(b)

(c)
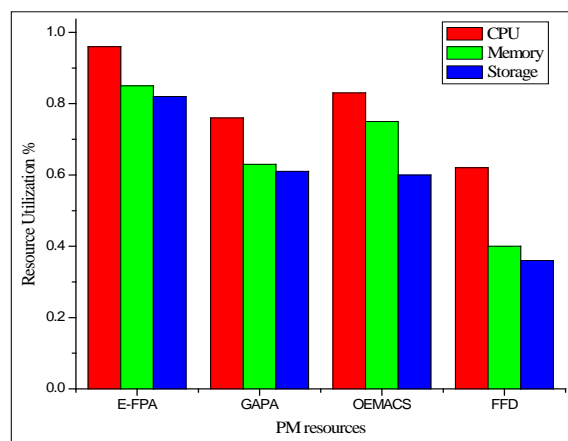
Fig.5. Resource Utilization: (a) PM component's, (b) average, (c) number of active PM

4.4 Energy consumption analysis

Fig 6(a) shows the E-FPA, GAPA, OEMACS and FFD scheme energy consumption in the Cloud

datacenter under different numbers of VM request. The energy consumption of the four schemes in-

creased in different degrees with the increasing number of VM demand by the user. Compared with

GAPA, OEMACS, and FFD, E-FPA has the least energy consumption. As can be seen from the results,

with an increase of the number of VM request, energy consumption becomes larger and larger. Due to an

increasing number of VMs, more PMs will be occupied, bringing greater energy consumption. Also, the

performance of E-FPA is always better than the GAPA, OEMACS, and FFD, this is because the pro-

posed scheme can explore the solution space between the local and global search more effectively so

that it can obtain solutions with a reduced number of used PMs compared with other schemes as depicted in Fig 6(b). Therefore, the number of active PMs is reduced and the rate of the energy consumption is remarkably decreases. Hence, total efficiency is increased in the proposed allocation scheme as shown in Table 5. We observed from the table that, the maximum energy consumption is 5015Kwh for E-FPA while the minimum of GAPA is 6520Kwh. OEMACS is 6450Kwh, and FFD is 6756Kwh for the same number of the user request and active PMs. Thus, there is 20.5 % saving in the overall datacenter infrastructure consumption of energy.

Table 5: Average result for the experiment

Scheme	Resource Utilization (%)	Active PMs	Energy Consumption	User request
E-FPA	95.4	450	5015	2000
GAPA	65.2	736	6520	2000
OEMACS	71.5	712	6450	2000
FDD	60	756	6756	2000
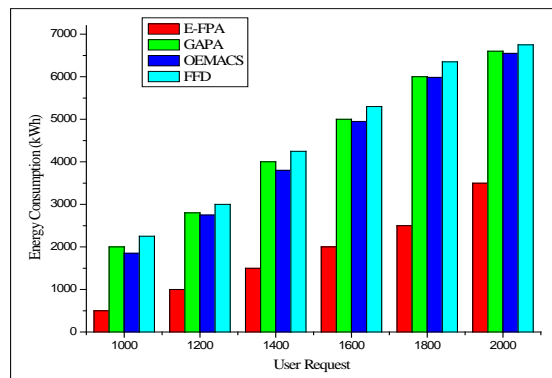
(a)
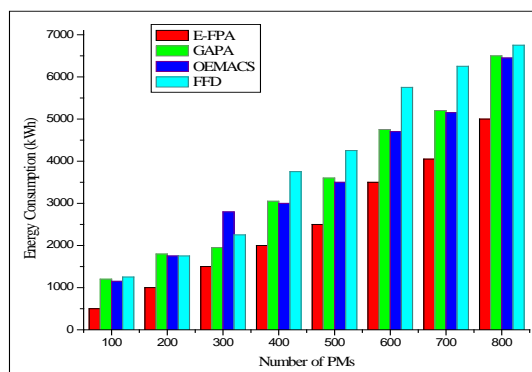
(b)

Fig. 6. Energy Consumption based on: (a) user request, (b) active PMs

5 Conclusion

This research paper, proposed Energy oriented Flower Pollination Algorithm E-FPA for VM allocation scheme with the goal of reducing datacenter energy consumption and improving on the resource utilization of the physical resources. The models and Algorithm's pseudo code have been elaborated. E-FPA is more efficient than the GAPA, OEMACS, and FFD schemes in regarding energy consumption and resource utilization. The energy consumption of the datacenter increases when the VM request changes as well. There is an increase in the energy consumption whenever there is an increase in the VM request by users. The allocation scheme uses DSP strategy to find near optimal solution quickly and to balance the intensification and diversification between the global and local search procedure to enhance the efficacy of the allocation scheme. Future research work will be using Multi-Objective approach of FPA to consolidate the datacenter resources.

Acknowledgments

References

[1] Oppong, E., S. Khaddaj, and H.E. Elasriss. Cloud Computing: Resource Management and Service Allocation. in Distributed Computing and Applications to Business, Engineering & Science (DCABES), 2013 12th International Symposium on. 2013. IEEE.

[2] Beloglazov, A., J. Abawajy, and R. Buyya, Energy-aware resource allocation heuristics for efficient management of data centers for cloud computing. Future generation computer systems, 2012. **28(5)**: p. 755-768.

[3] Foster, I., et al. Cloud computing and grid computing 360-degree compared. in 2008 Grid Computing Environments Workshop. 2008. Ieee.

[4] Usman, M., et al., A Conceptual Framework for Realizing Energy Efficient Resource Allocation in Cloud Data Centre. Indian Journal of Science and Technology, 2016. **9(46)**.

[5] Wadhwa, B. and A. Verma. Energy and carbon efficient VM placement and migration technique for green cloud datacenters. in Contemporary Computing (IC3), 2014 Seventh International Conference on. 2014a. IEEE.

[6] Michael, A.M. and K. Krieger, Server power measurement. 2010, Google Patents.

[7] Younge, A.J., et al. Efficient resource management for cloud computing environments. in Green Computing Conference, 2010 International. 2010. IEEE.

[8] Koomey, J., Growth in data center electricity use 2005 to 2010. A report by Analytical Press, completed at the request of The New York Times, 2011. 9.

[9] Han, G., et al., Resource-utilization-aware energy efficient server consolidation algorithm for green computing in IIOT. Journal of Network and Computer Applications, 2017.

[10] Buyya, R., A. Beloglazov, and J. Abawajy, Energy-efficient management of data center resources for cloud computing: a vision, architectural elements, and open challenges. arXiv preprint arXiv:1006.0308, 2010.

[11] Madni, S.H.H., M.S.A. Latiff, and Y. Coulibaly, An appraisal of meta-heuristic resource allocation techniques for IaaS cloud. Indian Journal of Science and Technology, 2016. **9(4)**.

[12] Beloglazov, A., Energy-efficient management of virtual machines in data centers for cloud computing. 2013: University of Melbourne, Department of Computing and Information Systems.

[13] Quang-Hung, N., et al. A genetic algorithm for power-aware virtual machine allocation in private cloud. in Information and Communication Technology-EurAsia Conference. 2013. Springer.

[14] Rodero, I., et al. Energy-efficient application-aware online provisioning for virtualized clouds and data centers. in Green Computing Conference, 2010 International. 2010. IEEE.

[15] Sharma, N.K. and G.R.M. Reddy. Novel energy efficient virtual machine allocation at data center using Genetic algorithm. in Signal Processing, Communication and Networking (ICSCN), 2015 3rd International Conference on. 2015. IEEE.

[16] Deore, S., A. Patil, and R. Bhargava, Energy-Efficient Scheduling Scheme for Virtual Machines in Cloud Computing. International journal of computer application, 2012. **56(10)**.

[17] Moganarangan, N., et al., A novel algorithm for reducing energy-consumption in cloud computing environment: Web service computing approach. Journal of King Saud University-Computer and Information Sciences, 2016. **28(1)**: p. 55-67.

[18] Phan, D.H., et al. Evolutionary multiobjective optimization for green clouds. in Proceedings of the 14th annual conference companion on Genetic and evolutionary computation. 2012. ACM.

[19] Shu, W., 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, 2014. **14(1)**: p. 1-9.

[20] Kansal, N.J. and I. Chana, Energy-aware Virtual Machine Migration for Cloud Computing-A Firefly Optimization Approach. Journal of Grid Computing, 2016. **14(2)**: p. 327-345.

[21] Tsai, C.-W. and J.J. Rodrigues, Metaheuristic scheduling for cloud: A survey. IEEE Systems Journal,

2014. **8(1)**: p. 279-291.

[22] Joseph, C.T., K. Chandrasekaran, and R. Cyriac, A novel family genetic approach for virtual machine allocation. Procedia Computer Science, 2015. 46: p. 558-565.

[23] Wu, G., et al. Energy-efficient virtual machine placement in data centers by genetic algorithm. in Neural Information Processing. 2012. Springer.

[24] Wang, X., Y. Wang, and H. Zhu, Energy-efficient multi-job scheduling model for cloud computing and its genetic algorithm. Mathematical Problems in Engineering, 2012. 2012.

[25] Pacini, E., C. Mateos, and C. García Garino, Dynamic scheduling based on particle swarm optimization for cloud-based scientific experiments. CLEI Electronic Journal, 2014. **17(1)**: p. 3-3.

[26] Wang, S., et al. Particle swarm optimization for energy-aware virtual machine placement optimization in virtualized data centers. in Parallel and Distributed Systems (ICPADS), 2013 International Conference on. 2013. IEEE.

[27] A Vouk, M., Cloud computing–issues, research and implementations. CIT. Journal of Computing and Information Technology, 2008. **16(4)**: p. 235-246.

[28] Li, B., et al. Enacloud: An energy-saving application live placement approach for cloud computing environments. in Cloud Computing, 2009. CLOUD'09. IEEE International Conference on. 2009. IEEE.

[29] Ho, Y., P. Liu, and J.-J. Wu. Server consolidation algorithms with bounded migration cost and performance guarantees in cloud computing. in Utility and Cloud Computing (UCC), 2011 Fourth IEEE International Conference on. 2011. IEEE.

[30] Xu, J., et al. Survivable virtual infrastructure mapping in virtualized data centers. in Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on. 2012. IEEE.

[31] Shu, W., 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, 2014. **2014(1)**: p. 1.

[32] Ghribi, C., M. Hadji, and D. Zeghlache. Energy efficient vm scheduling for cloud data centers: Exact allocation and migration algorithms. in Cluster, Cloud and Grid Computing (CCGrid), 2013 13th IEEE/ACM International Symposium on. 2013. IEEE.

[33] Liu, X.-F., et al., An energy efficient ant colony system for virtual machine placement in cloud computing. IEEE Transactions on Evolutionary Computation, 2016.

[34] Kaur, T. and I. Chana, Energy Efficiency Techniques in Cloud Computing: A Survey and Taxonomy. ACM Computing Surveys (CSUR), 2015. **48(2)**: p. 22.

[35] Yang, X.-S. Flower pollination algorithm for global optimization. in UCNC. 2012. Springer.

[36] Yang, X.-S., M. Karamanoglu, and X. He, Multi-objective flower algorithm for optimization. Procedia Computer Science, 2013. 18: p. 861-868.

[37] Abdelaziz, A., E. Ali, and S.A. Elazim, Flower Pollination Algorithm and Loss Sensitivity Factors for optimal sizing and placement of capacitors in radial distribution systems. International Journal of Electrical Power & Energy Systems, 2016. 78: p. 207-214.

[38] Abdel-Raouf, O., I. El-Henawy, and M. Abdel-Baset, A novel hybrid flower pollination algorithm with chaotic harmony search for solving sudoku puzzles. International Journal of Modern Education and Computer Science, 2014. **6(3)**: p. 38.

[39] Babu, M. and S. Jaisiva, Optimal Reactive Power Flow by Flower Pollination Algorithm. Asian Journal of Applied Science and Technology (AJAST), 2017. **1(3)**: p. 137-141.

[40] Mishra, M. and A. Sahoo. On theory of vm placement: Anomalies in existing methodologies and their mitigation using a novel vector based approach. in Cloud Computing (CLOUD), 2011 IEEE International Conference on. 2011. IEEE.

[41] Xiong, A.-p. and C.-x. Xu, Energy efficient multiresource allocation of virtual machine based on PSO in cloud data center. Mathematical Problems in Engineering, 2014. 2014.

[42] Lin, W., et al., Multi-resource scheduling and power simulation for cloud computing. Information Sciences, 2017. 397: p. 168-186.

[43] Pan, J.-S., et al., An improvement of flower pollination algorithm for node localization optimization in WSN. Journal of Information Hiding and Multimedia Signal Processing, 2017. **8(2)**: p. 500-509.

[44] Jamil, M. and X.-S. Yang, A literature survey of benchmark functions for global optimisation problems. International Journal of Mathematical Modelling and Numerical Optimisation, 2013. **4(2)**: p. 150-194.

[45] Wang, R. and Y. Zhou, Flower pollination algorithm with dimension by dimension improvement. Mathematical Problems in Engineering, 2014. 2014.

[46] Park, K. and V.S. Pai, CoMon: a mostly-scalable monitoring system for PlanetLab. ACM SIGOPS Operating Systems Review, 2006. **40(1)**: p. 65-74.