Energy Efficient Dynamic Particle Swarm Optimization (EEDPSO) Resource Allocation in Cloud Computing

M. Balakrishna
Assistant Professor, Department of Computer Science, Maharani's Science College for Women, J.L.B. Road, Mysuru, India. E-mail: balakrishna.mysore@gmail.com

Dr.K. Siddaraju
Assistant Professor, Department of Computer Science, Maharani's Science College for Women, J.L.B. Road, Mysuru, India. E-mail: dr.ksiddaraju0103@gmail.com

Received May 26, 2020; Accepted July 30, 2020
ISSN: 1735-188X
DOI: 10.14704/WEB/V17I2/WEB17021

Abstract

In cloud computing, a key characteristic is On-demand resource management. For proper resource allocation, fair computational resource sharing should be done by cloud providers. Better resources should be allocated to all users. Resource utilization enhancement is also focused by reducing resource fragmentation, where virtual machines are mapped to physical servers. In cloud environment, an adaptive resource allocation mechanism is proposed in this recent work. A limited resource quantity mapping to independent user for finishing their jobs is focused. However, based on operational costs, there will be an increase in energy requirement for operating cloud infrastructure. According to literature, energy minimization is focused in this work by CPU utilization regulation while operating at maximum frequency. For computing energy consumption, resource utilization and fairness, introduced a Dynamic Particle Swarm Optimization (DPSO) model. The computation is done while executing jobs by VMs on cloud computing resources in absent presence. The Google workload trace is used in simulation and resource wastage are minimized using proposed algorithm and achieves a better utilization of resource when compared with other allocation techniques as demonstrated in results.

Keywords

Dynamic Particle Swarm Optimization, Energy Efficiency, Resource Allocation, Cloud Computing.

Introduction

In recent decades, in data store and large-scale computing, cloud computing has gained a more attention due to its ability to enable computing resources sharing which are distributed over the world [1]. In isolated systems, physical barriers inherent are broken down by the cloud computing infrastructures. Resources group management is automated as a single entity and for users, data storage and computational power are provided by this, which ranges from single laptop access by user to thousands of computing nodes allocation which are distributed around the world [2].

With virtualization technologies, workload balancing and automated systems management, for data and computation centres, cloud computing is a natural evolution. A seamless computing platforms can be formed by integrating globally distributed resources in cloud based services.

A cloud is made from various physical machines. Each physical machine runs different virtual machines which are introduced to the end-clients, or alleged customers, as the registering assets. The engineering of virtual machines depends on physical PCs with comparative usefulness [3]. In distributed computing, a virtual machine is a visitor program with programming assets which works like a genuine physical PC. However, high outstanding burden on virtual machines is one of the difficulties of distributed computing in the allotment of virtual machines.

The errand, mentioned by a customer, needs to hold on to be apportioned to the work and the assets required. This procedure is free of the chief need of the assignments [4]. In any case, the customer who claims the assignment may offer bigger incentive for it to attempt to raise his/her need and in the long run may prevail with regards to assuming responsibility for the assets required. Clients can devour administrations dependent on the administration level arrangement that characterizes their necessities of Quality of Services (QoS) boundaries.

Asset assignment is a method that guarantees the allotment to virtual machines when numerous applications need various assets of CPU and info/yield memory [5]. In distributed computing there are two specialized limitations. Initially, the limit of the machines is truly restricted; also, needs for the usage of the errands ought to be in amicability with augmenting the proficiency of assets.

At last, the holding up time and the finishing time are to be diminished, so as to diminish the expense of framework execution. Asset distribution in the cloud condition is used to accomplish consumer loyalty with insignificant preparing time [6]. Diminishing the expenses of renting assets notwithstanding guaranteeing nature of administration and improving throughput for trust and fulfillment of the specialist co-op is considered as another target.

In this work, the decency issue is considered in asset assignment, which implies nobody is allotted much preferred assets over others. In multi asset condition, assets of different sorts, for example, CPU, memory, and circle stockpiling, are required by clients with various requests [7]. In this situation, reasonable designation expects to balance the biggest asset division of all out accessibility assigned to every client. The other objective of asset portion is to ensure the computational assets to be completely used.

Because of the assortment of asset necessities with various VM types, loads of asset parts in physical workers could be created during the VM organization [8]. Consequently, an effective asset assignment strategy ought to limit the measure of assets parts.

Then again, the energy utilization is essentially expanding alongside the hazardous development of cloud server farm. Numerous reports have indicated that PCs devour over 8% of the all out energy created, which turns into a raising danger to nature. In this circumstance, elevated level energy proficiency in cloud server farm is broadly concentrated to decrease energy utilization by specialists everywhere on the world [9].

In cloud data centre, high energy consumed because of two reasons. They are, improper resource allocation in cloud computing and increase in cloud users count and computers. Because of massive size of computers and cloud users, noticeable amount of energy is consumed by this. Allocation of resources like bandwidth, memory, disk and CPU are major problem and it requires to be resolved. In cloud data centre, more energy consumption is produced due to unreasonable resources allocation [10].

Energy consumption can be minimized greatly using a resource allocation algorithm having energy efficiency. In cloud computing field, it has been studied widely.

For satisfying Service Level Agreements or Quality of Service requirements while minimizing energy consumption, resource allocation is highly needed one in cloud data centre. In heterogeneous data center, storage and CPU utilizations are analysed for minimizing energy consumption while giving satisfied performance requirements. As a result, optimum balance between energy metrics and resource utilization are computed which resides around 70% storage and % CPU usage.

Resource utilization are maintained in an optimum manner for achieving an energy conservation [11]. There is a need to minimize the active nodes count and there is a need to turn off the idle nodes for minimizing total energy consumption. For energy efficiency for VMs allocations, solutions can be found using a traditional heuristic algorithms, but they may fall into locally optimum solutions in an easy way.

The particle swarm optimization (PSO) algorithm and genetic algorithm (GA) are also used for solving multidimensional bin packing problem. An adaptive heuristic algorithm is GA which is premised on natural genetic and selection's evolutionary ideas. A stochastic optimization method based on population is particle swarm optimization (PSO) which is inspired by fish schooling or bird flocking's social; behaviour.

In distributed system, better solution can be obtained using PSO algorithm than GA as illustrated in various studies. Convergence of PSO algorithm is faster than GA. So, in cloud data centre, for optimizing multi resources energy efficient allocations, it is feasible to use particle swarm optimization.

According to literature, energy minimization is focused in this work by CPU utilization regulation while operating at maximum frequency. For computing energy consumption, resource utilization and fairness, introduced a Dynamic Particle Swarm Optimization (DPSO) model. The computation is done while executing jobs by VMs on cloud computing resources in absent presence.

The rest of the research work is organized as follows, section 2 reviews the recent methods for resource allocation in cloud computing. Section 3 explains the proposed methodology. Section 4 discuss the results and discussion. Section 5 deals with conclusion and future work.

Literature Review

In cloud environment, to allocate resources, different resource management techniques with various principles and policies are reviewed in this section.

A distributed negotiation mechanism is proposed by An et al [12]. Over decommitment penalty and contract price, agents negotiate, which makes agents for decommitting from contracts at a cost. Representative workloads and scenarios are used for making experimental comparison. Fixed price model and combinatorial auctions of Amazon's Elastic Compute Cloud are used in experimentation and a high social welfare can be achieved using negotiation model as shown.

A multiple Service Level Agreement (SLA) parameter and resource allocation is presented by Pawar et al [13]. In cloud, in execution of high priority task, resource utilization can be enhanced using pre-emption mechanism. An algorithm is proposed in this work by considering multiple SLA parameters like needed CPU time, network bandwidth and memory, pre-emptable task execution. In a condition with resource contention is fierce of this algorithm, better resource utilization can be provided using this algorithm as shown in experimental results.

On the cloud, a resource allocation mechanism according to game theory's uncertainty principle and coalition formation is proposed by Pillai et al [14] for machines. A comparison is made with other resource allocation algorithms which are deployed on cloud. With better resource utilization, high request satisfaction is provided by this resource allocation technique using machines coalition-formation on cloud.

For allocating services effectively to participants, market mechanism is proposed by Fujiwara et al [15]. For co-allocations and workflows, service combination can be ordered by users as enabled by this mechanism and it also enables to reserve current/future services in a spot/forward market. In probable setting, better performance can be shown by this mechanism as shown in evaluation.

A new approach which allocates resources with minimum wastage and maximum profit is introduced by Gouda et al [16]. Various parameters like processor request count, cost and time are forms the base for this developed resource allocation algorithm. In cloud environment, for a better jobs resource allocation, priority algorithm is developed. In an effective way, for various jobs or models simulation, it can be used.

An evaluation is done after allocating resources to different jobs in an effective manner and with profit, it can produce better performance in cloud computing as illustrated. In different systems, all algorithms performance study and case studies are also presented.

In cloud computing, for network resource allocation, a framework is proposed by Arfeen et al [17] according to tailored active measurements. Major role is played in resource allocation are, optimality condition change along with dynamic user requirements, traffic considerations and network topology as concluded by them. In vital applications like cloud computing, in shaping resource allocation and management techniques, internet application protocols can architectures are very important.

For balancing a load, a hybrid algorithm is presented by Mousavi et al [18]. The cloud provider network's throughput can be maximized using this hybrid algorithm. The grey wolves optimization algorithm (GW) and teaching-learning-based optimization algorithm (TLBO) are used in this hybrid algorithm. When compared with single algorithm's usage, better performance can be achieved using hybrid algorithm.

According to cost and time, load balancing can be done effective and it also priorities balances and it avoids trapping of local optimum. This produces minimum waiting time. A comparison is made with GW and TLBO algorithm for evaluating proposed algorithm's effectiveness and presented the experimental results.

In server's multidimensional resource utilization, for measuring unevenness, "skewness" concept is introduced by Xiao et al [19]. In a better manner, various workload types can be combined by reducing skewness and it enhances server resources. In systems, overloading can be prevented effectively by developing a heuristics set while minimizing used energy. Good performance can be achieved using proposed algorithm as demonstrated in experimental results an trace driven simulation.

An opportunistic decode-and-forward (ODF) protocol is introduced by Gunduz et al [20], where based on channel state, relay are used. System's delay-limited capacity is enhanced significantly using Opportunistic cooperation and its performance is very near to cut-set bound. With respect to minimum outage probability, system performance is considered.

In outage probability perspective, performance close to cut-set bound is provided by ODF as shown. For cooperative systems, feedback's importance is emphasized by results which have delay sensitive applications.

A time series analysis techniques for predicting expected workload parameters from measured system metrics is presented by Chandra et al [21]. Then for allocating server resources according to computed application requirements, a constrained non-linear optimization is presented. In system model, while incorporating nonlinearity, applications transient behaviour can be captures using this method, which is a major advantage of it.

With real-world as well as synthetic web workloads, simulations are done for evaluating this technique. System resources can be allocated judiciously using this technique as shown in results, especially in transient overload conditions.

A game theoretic resources allocation algorithm is introduced by Xu et al [22], where, resource utilization and fairness among users are considered. In maintain fairness, this algorithm’s optimality is shown by experimentation using FUGA implementation on an 8 node server cluster, where comparison is made with Hadoop scheduler evaluation. The Google workload trace based simulations shows that resource wastage can be minimized using this algorithm and a better utilization of resources can be achieved when compared with other allocation techniques.

These techniques are not dynamic and are not suitable for solving allocation problems as observed from the above review. There are large search space in these techniques and it results in a huge amount of possible solutions and it is difficult to find optimum solution. For solving these problems, there is no effective technique in current situation.

In these conditions, instead of computing semi-optimal solution, fully optimized solutions are computed by traditional approach in a shorter time. So, it is important to use meta-heuristic algorithm for ensuring solution convergence and they will be trapped into local minima and they have a global overview. Consequently, for enhancing accuracy and local optimization, selected the Dynamic PSO algorithm.

Proposed Methodology

Energy minimization by CPU utilization regulation is focused in this work while operating at maximum frequency. For computing energy consumption, resource utilization and fairness, introduced a Dynamic Particle Swarm Optimization (DPSO) model while executing the jobs using VMs in absent presence on cloud computing resources.

1). Resource Management System Modeling

With heterogeneous physical servers, there are distributed as well as large scaled data centre in every cloud provider and like a pay-per-use business models, it provides various computational resources. User can apply for a virtual machine and for the occupied time, it can charged as provided by Infrastructure-as-a-Service [22]. On a physical server, hypervisor, VMwre or Xen creates VM.

For accomplishing cloud users missions like Map reduce or web services jobs, on a VMs cluster, high-performance applications are deployed by cloud users, which are termed as jobs in this paper. For users, in order to simplify the selection, possible VM types groups are offered typically by cloud providers and other resources quantities, storage size, memory size and CPU cores count are used for defining every type of VM.

There is a need to adjust resource allocation decisions of providers in a dynamic manner as heterogeneous VMs are needed by various users and it may vary with time. So, for cloud, a resource management system is designed here.

- **Resource Management System**

For centralizing control and coordinating physical resources, it is necessary to have resource management system. On a distributed and complex cloud system, an effective as well as fair resource allocation mechanism is designed. Proposed cloud resource management system's architecture is illustrated in Figure 1. There are four components in this resource management system, namely, control center (CC), infrastructure management (IM), cloud environment monitor (CEM) and register center (RC) and are described in the following section.

(i) RC: For management and connection in cloud data centre, information about every physical server should be registered to RC.
(ii) CEM: Information like disk storage, memory, CPU consumption, status like shutdown, running and starting monitoring, IP address of physical servers and host names are retrieved using this component.
(iii) IM: Virtualized infrastructures like creation and release of virtual machines are deployed and manages using this.
(iv) CC: About resource allocation, most appropriate decisions are provided using this computing centre.

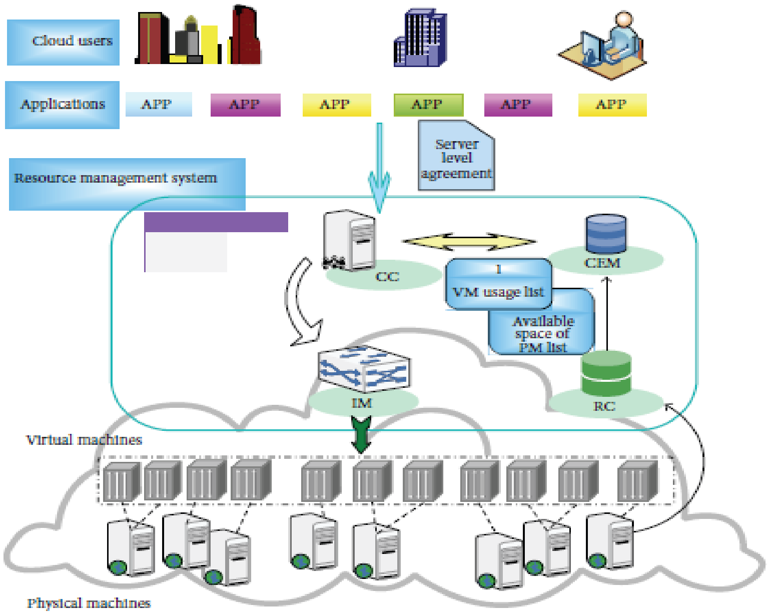


Figure 1 A framework of cloud resource management system

In RC, registered physical servers resource consumption and statuses are monitored using CEM. If a new physical server is joining in cloud, then an information like IP address, MAC address should be registered in RC. If a service request is send by a user to cloud, CC receives the resource requirements in this request. According to the information collected by CEM, an intelligent resource allocation decision are made by CC. For placing virtual machines and physical servers, IM executes the allocation decision.

In cloud environment, an adaptive resource allocation mechanism is proposed in this work, where limited resource's quantity mapping into an independent users problem is dealt for completing the job. In a time-slotted paradigm, resources are allocated in this resource allocation mechanism. Recorded the dynamically arriving user request in current slot and at the next time slot's starting, they will be served for allocating resources. Time slot's every start is termed as decision moment.

2). Mathematical Model

In a cloud cluster, assume there are p physical servers and every server is represented as m, where $1 \leq m \leq p$. This work assumes k resources types and a capacity vector is used for describing every physical server m's available resources and are monitored using CEM. Every resource type is represented as j. For instance, (4,8,40) indicates, 40GB disk storage, 8 GB memory and 4 CPUs in a physical server.

The job submitted by user i is represented as Ji, where $i \in \{1, 2, . . . ,s\}$. Cloud providers predefines various VM types and are encoded using a vector as $\vec{r_i} = (r_{i1}, r_{i2}, r_{i3}, \dots \dots .. r_{ij}, \dots \dots r_{ik})$. For a VMs cluster, every job applies with similar type which needs to be fully executed. Increase in assigned VMs count to this job will increase the performance in general. However, creation of more VMs will increase the cloud provider's cost. So, decisions regarding VMs count assigned to every job is decided by cloud provider.

Two physical servers are considered in figure 2 with capacity vectors (4, 6, 50), (4, 8, 40). For three VMs type, three users have applied and are described as (2, 2, 10), (1, 1, 10) and (2, 4, 20). During the creation of virtual machine on this server, resources like physical server storage, memory and CPU are occupied. In data centre, all physical servers capacity states are checked by resource management system at every decision moment and all user request are analysed for generating resource requirement matrix.

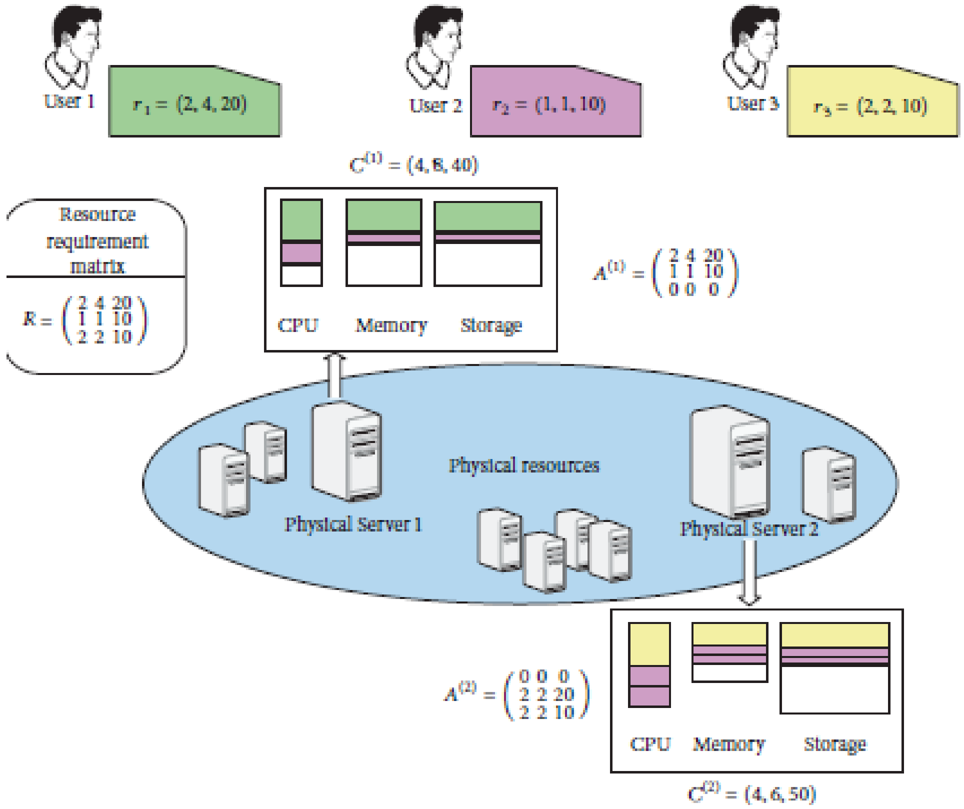

User 1
$r_1 = (2, 4, 20)$
User 2
$r_2 = (1, 1, 10)$
User 3
$r_3 = (2, 2, 10)$
$C^{(1)} = (4, 8, 40)$
Resource requirement matrix
$R = \begin{pmatrix} 2 & 4 & 20 \\ 1 & 1 & 10 \\ 2 & 2 & 10 \end{pmatrix}$
CPU
Memory
Storage
$A^{(1)} = \begin{pmatrix} 2 & 4 & 20 \\ 1 & 1 & 10 \\ 0 & 0 & 0 \end{pmatrix}$
Physical resources
Physical Server 1
Physical Server 2
$A^{(2)} = \begin{pmatrix} 0 & 0 & 0 \\ 2 & 2 & 20 \\ 2 & 2 & 10 \end{pmatrix}$
CPU
Memory
Storage
$C^{(2)} = (4, 6, 50)$

Figure 2 An example of cloud resource allocation

Definition 1 (resource requirement matrix).A matrix is used for defining resource requests submitted by various users. Assume an$s \times k$ dimensional matrix as R, where every user's VM type needs are specified by its rows and various resources quantity are described by its columns as illustrated in Figure 2.

$$R = \begin{pmatrix} \vec{r_i} \\ \vec{r_2} \\ \vdots \\ \vec{r_s} \end{pmatrix} = \begin{bmatrix} r_{11} & \cdots & r_{1k} \\ \vdots & \ddots & \vdots \\ r_{s1} & \cdots & r_{sk} \end{bmatrix} \qquad (1)$$

Resource allocation matrix is produced by resource allocation problem and reasonable mapping from resources to cloud users are computed using physical servers capacity sets. In can also be stated as, various resources types on every physical server needs to be distributed in a effective as well as fair manner for creating its required VMs.

Definition 2 (allocation decision). An allocation matrix (m) is used for describing a possible resource allocation state for the physical server m.

$$A^{(m)} = \begin{pmatrix} \overrightarrow{a_i}^{(m)} \\ \overrightarrow{a_2}^{(m)} \\ \vdots \\ \overrightarrow{a_s}^{(m)} \end{pmatrix} = \left[\begin{bmatrix} a_{11}^{(m)} & \cdots & a_{1k}^{(m)} \\ \vdots & \ddots & \vdots \\ a_{s1}^{(m)} & \cdots & a_{sk}^{(m)} \end{bmatrix}\right] \qquad (2)$$

Where, on physical server m, resources quantity j allocated to user i is represented as $a_{ij}^{(m)}$.

According to resource requirement matrix, every physical server's possible allocation status collection is maintained in an allocation decision A.

$$A = \{A^{(1)}, A^{(2)}, \dots ., A^{(m)}, \dots . A^{(p)}\}. \qquad (3)$$

An allocation decision example is shown in figure 2. The allocation matrix is represented as $A^{(1)}$, if one type-$\overrightarrow{r_2}$ VM and one type-$\overrightarrow{r_i}$VM are created by Physical Server 1. There exist a own allocation matrix for every physical servers as shown in Figure 2 and it has an allocation decision $A = \{A^{(1)}, A^{(2)}\}$.

Further, $\varphi_{ij} = \sum_m a_{ij}^{(m)}$ represents the resources count j allocated to user i. In notation section, summarized their description and parameters.

3). Energy Efficient Resource Allocation Model

A multi-dimensional problem with different prices and bin sizes is used for describing resources allocation problem in cloud data centre. Consider VMs count as N and they must be allocated to M hosts count in cloud data centre. The heuristic algorithm's search space is represented as $M^N$and it is used in optimal VMs allocations computation problem.

Optimum energy efficient solution is computed using allocation. The entire data centre's energy efficiency is highly effected by resources. Optimum energy conservation can be achieved by maintaining every resources utilization in an optimum level of utilization. Here, as in the following manner, total Euclidean distance δ is defined.

$$\delta = \sum_{i=1}^{n} \sqrt{\sum_{j=1}^{d} (u_j^i - ubest_i)^2}, \qquad (4)$$

Where, dimension is represented as d and it represents a resource type like bandwidth, memory, disk and CPU. In cloud data centre, hosts count is represented as n, for resource i and host j, utilization is represented as u_j^i, for u_j^i energy efficient, best utilization is represented as $ubest_i$ and there will be w best utilization value for every resource type like 70% storage and 50% CPU usage.

Between energy consumption and resource utilization, optimal balance is represented using total Euclidean distance δ. In the entire system, optimum energy efficiency can be obtained by reducing total Euclidean distance δ. In this condition, as mentioned below, described the resources energy efficiency.

$$\text{objection: min } \delta \qquad (5)$$
$$\text{constraints: } x_h^j = 0 \qquad (6)$$
$$\sum_h x_h^j = 1, \qquad \forall j, \qquad (7)$$

Where, virtual machine VM$_j$ allocated to node h is represented as $x_h^j = 1$, VM$_j$ not allocated to node h are represented as $x_h^j = 0$. Every VM is allowed to be allocated with only one node as indicated in expression (7).

Following inequality constraints must be satisfied by every resource for satisfying limitations and are given by,

$$\sum_j r_j^{CPU} * x_h^j \leq c_h^{CPU}, (8)$$
$$\sum_j r_j^{RAM} * x_h^j \leq c_h^{RAM}, (9)$$
$$\sum_j r_j^{BW} * x_h^j \leq c_h^{BW}, \qquad (10)$$
$$\sum_j r_j^{DISK} * x_h^j \leq c_h^{DISK}, (11)$$

In expression (8) to (11), CPU demand is represented as r_j^{CPU}, memory is represented asr_j^{RAM}, bandwidth is represented as r_j^{BW}, disk for VM$_j$is represented as r_j^{DISK}, these resources capacity for VM$_j$ is represented as c_h^{CPU}, c_h^{RAM}, c_h^{BW}, c_h^{DISK}. On node h, if there are various allocated VMs, node h’s capacity should be greater than VMs total resources demand.

4).Resource Allocation based on Dynamic Particle Swarm Optimization (DPSO)

In virtual machine allocation, for minimizing energy consumption, an EEDPSO algorithm is designed and implemented in this section and in cloud data centre, resources can be dealt using this algorithm.

a). Particle Swarm Optimization (PSO) Algorithm

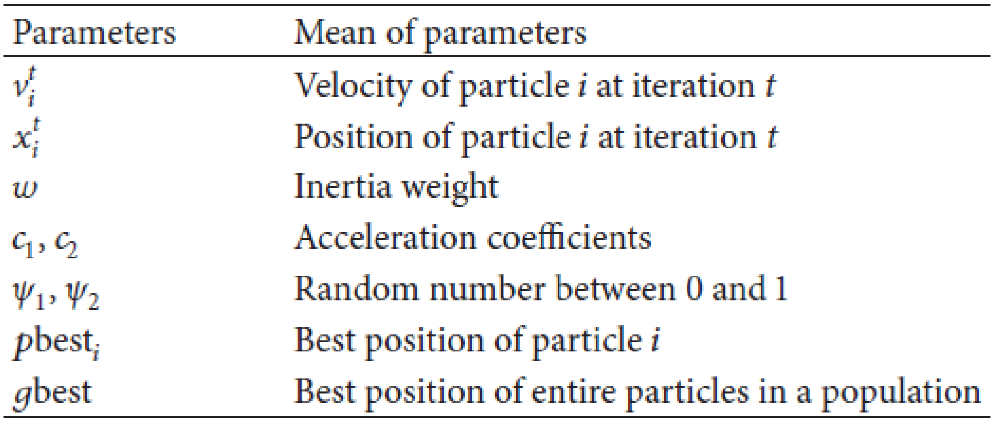
A stochastic optimization method based developed population is Particle swarm optimization (PSO). Organisms social behaviour motivates them for looking into specie's collaboration effect on attaining their goals as group like bird flocking and fish schooling [23]. In past several years, in various research and applications, PSO found its wide application. In a faster as well as cheaper manner, better results can be obtained using PSO when compared with other techniques like GA as proven.

Potential solutions in PSO are termed as particles and they fly follow the current optimum particles and fly through solution space. Particles have memory and are used for storing its previous state. In any case, individuality of all particles are preserved and with no restriction, they share same point in belief space. Using expression (12), in problem space, every particle's position is evolved in PSO algorithm.

There is an initial random velocity for every particle and two randomly weighted factors having influence on particle movement. They are, sociality and individuality. Ability to move towards best previous position of particle is defined by individuality and ability to move towards best previous position of neighbourhood is defined by sociality. Towards *p*best and *g*best locations, every particle's velocity is changed by particle swarm optimization at every time step.

A random term is used for weighting this acceleration, which is different from random numbers generated for acceleration towards pbest and lbest locations. In addition, specific applications focusing on specific requirement can use this particle swarm optimization. There are few parameters needed to adjusted, so Particle swarm optimization gains the wide use. Table 1 shows the parameters and mean of this parameters.

Table 1 Parameters and the mean of parameters.



Parameters	Mean of parameters
v_i^t	Velocity of particle *i* at iteration *t*
x_i^t	Position of particle *i* at iteration *t*
w	Inertia weight
c_1, c_2	Acceleration coefficients
ψ_1, ψ_2	Random number between 0 and 1
$p\text{best}_i$	Best position of particle *i*
$g\text{best}$	Best position of entire particles in a population

Consider

$$v_i^{t+1} = w.v_i^t + c_1.\psi_1.(pbest_i - x_i^t) + c_2.\psi_2.(gbest - x_i^t) \quad (12)$$

$$x_i^{t+1} = x_i^t + v_i^t \quad (13)$$

- **Problem Definition**

In order to deal with problems where, a surface or point is used for representing best solution in an n-dimensional space, a global optimization algorithm called Particle swarm optimization is used. In this space, hypothesis is plotted and with an initial velocity, they are seeded. Between particles communication channel is established. Through the solution space, particles moves and after every timestamp, based on some fitness function, they are evaluated.

With respect to time, there will be an acceleration of particles towards particles having better fitness in their group. Swarm size as well as topology will be varied in dynamic PSO. In dynamic particle swarm optimization, every particle's velocity is changed or accelerated towards its best values pbest and lbest at every time step. A random term is used for weighting this acceleration, which is different from random numbers generated for acceleration towards pbest and lbest locations.

- **Dynamic Weight**

The PSO performance can be enhanced using inertia weight adjustment. Local optima trapping will happen with fixed inertia weight and optimum point missing will happen with linear decrease in inertia weight as mentioned above. So, in the following manner, inertia weight is computed for rectifying PSO drawbacks.

$$w = \begin{cases} w_{min} - \frac{(w_{max} - w_{min})*(f_i - f_{min})}{f_{avg} - f_{min}}, & f_i \leq f_{avg} \\ w_{max} & , f_i > f_{avg} \end{cases} \quad (14)$$

Where, maximum inertia weight is represented as w_{max}, minimum inertia weight is represented asw_{min}, particle I's current fitness value is represented as f_i, all particle's current average fitness is given by f_{avg} and all particles minimum fitness is given by f_{min} .Automatic increase in inertia weight will happen, if all particle's fitness value is converged or tended to converge.

There will be an automatic reduction in inertia weight, if all particle's fitness values are scattered. This is due to the dynamic change in inertia weight value along with fitness value. It is termed as dynamic inertia weight. Following are the advantages of dynamic inertia weight.

a. Small inertia weight values are derived by the particles with fitness values greater than average fitness value in an automatic manner. So that, their flight velocity will be reduced and be protected.

b. Large inertia weight values are derived by the particles with fitness values less than average fitness value in an automatic manner. So that, their, flight velocity will be increased and they move closer to excellent particles in a fast manner.

- **Dynamic Learning Factors**

The PSO algorithm's two learning factors are constants in general. Undesired outputs are produced due to inappropriate factors as stated above. So, in the following manner computed the learning factors in DPSO's optimization process.

$$C_1 = C_1, init + \frac{C_1, fin - C_1, init}{t_{max}} * t \qquad (15)$$

$$C_2 = C_2, init + \frac{C_2, fin - C_2, init}{t_{max}} * t \qquad (16)$$

Where, $C_1, init > C_2, init$; $C_1, fin < C_2, fin$; DPSO's maximum running time is represented as t_{max} .

With respect to time, there will be automatic increase or decrease in learning factors in expression (15) and (16), so, they are termed as dynamic leaning factors. The C_1 is changed to minor from major, while C_2 is changed to major from minor. So, in optimization's initial stage, weak social-learning ability and strong self-learning ability are exhibited by particles, which leads to enhanced global search ability of algorithm.

On the other side, in optimization's later stages, strong social-learning ability and weak learning ability are exhibited by particles, which leads to accelerated convergence to global optimum solution. Using following expressions, positions and velocity of particles are updated after computing best values.

$$V[id] = v[id] + C_1 * r(id) * (pbest[id] - x[id]) + C_2 * r * (id)(gbest[id] - x[id]) \qquad (17)$$

$$x[id] = x[id] + v[id] \qquad (18)$$

Where, particle velocity is represented as $v[id]$, current particle is represented as $x[id]$, random number is represented as r(id) and its value lies between 0 to 1, learning factors are represented asC_1 and C_2 and in general, its values are assigned as 2.

- **Algorithm Description**

Algorithm 1shows the flowchart of DPSO based energy efficient allocation algorithm. There are four major steps in this algorithm, which are explained as,

(1) Particles are initialized. In a random manner, VM's N sequences are generated. Using First Fit algorithm, VM is allocated to node for every sequence. At the first time, resources are supplied using this algorithm. Nparticles are obtained, which constitutes a swarm.

(2) For every particle, fitness value is evaluated. The $g$best and $p$best positions of particles are initiated.

(3) Particles positions and velocity are updated. Particles are updated, if they satisfies the constrains in expression (4) and (7). Else, particles will not be updated. For VM j, while Σjh> 1, it cannot be allocated to more than one node, with Σ= 0, it cannot be allocated to any node.

(4)If maximum value is exceeded by iteration number, stop; else move to second step.

Algorithm 1.Pseudocode of Dynamic Particle Swarm Optimization (DPSO)

```
Input: No. of VMs
Output: Allocated resources
        For every particle
        Function value is initialized
        End
        Average fitness value is computed
        Do
        For every particle
        If fitness value is less than average
        Consider the particle
        Fitness value is computed
        If fitness value is better than best
        Fitness value (pbest) in histore
        Current value is assigned as new pbest
        End
        Particle with best fitness value of all particles is selected as gbest
        For every particle
        Using expression (17), particle velocity is computed.
        Using expression (18), particle position is updated
        End
```

Results and Discussion

On a Dell Optiplex9010 with JDK 1.7, simulations are run. Following assumptions are made for minimizing simulations complexity. They are, in simulations, two resource types namely, memory and CPU are considered. Various resources, predicted maximum consumption is indicated every job request submitted by user and VMs cluster with similar type is used for handling the request. Cloud provider estimates the total resources amount provided for every time slot.

Table 1 CPU utilization

Number of Physical servers (p)	ODF	GT-RA	EEDPSO
100	0.64	0.73	0.79
200	0.61	0.66	0.71
300	0.63	0.68	0.75
400	0.65	0.74	0.80
500	0.67	0.75	0.79

The table.1.shows the CPU utilization rate of the proposed and existing method.

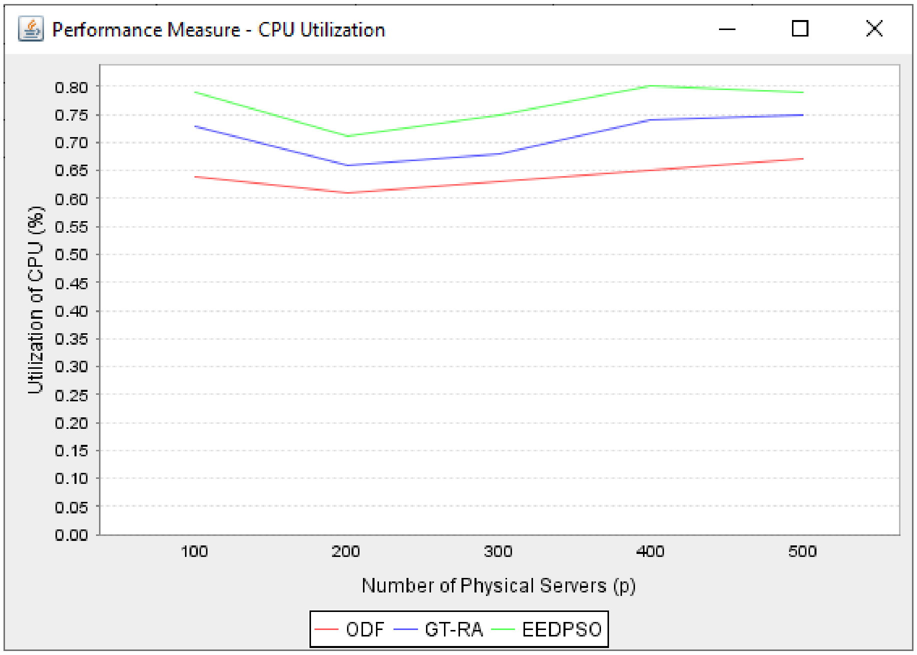


Figure 3 Comparison of CPU utilization between the proposed and existing methods

The proposed EEDPSO's resource utilization is better when compared with GT-RA and ODF technique as shown in figure 3. In a low level, disk and CPU utilization are maintained while virtual machines count increases to 20 from 10. With increase of virtual machines count to 50 from 20, there will be a huge increase in proposed technique's resource utilization.

With 50 virtual machines, there will be an increase in CPU utilization of 35%. This is a better utilization value and it makes optimum balance between energy consumption and resource utilization. Cloud data centre's energy efficiency can be enhanced using this technique.

Table 2 Memory Utilization

Number of Physical servers (p)	ODF	GT-RA	EEDPSO
100	0.61	0.67	0.72
200	0.62	0.66	0.69
300	0.63	0.64	0.70
400	0.61	0.68	0.71
500	0.59	0.70	0.74

The table.2.shows the memory utilization rate of the proposed and existing method

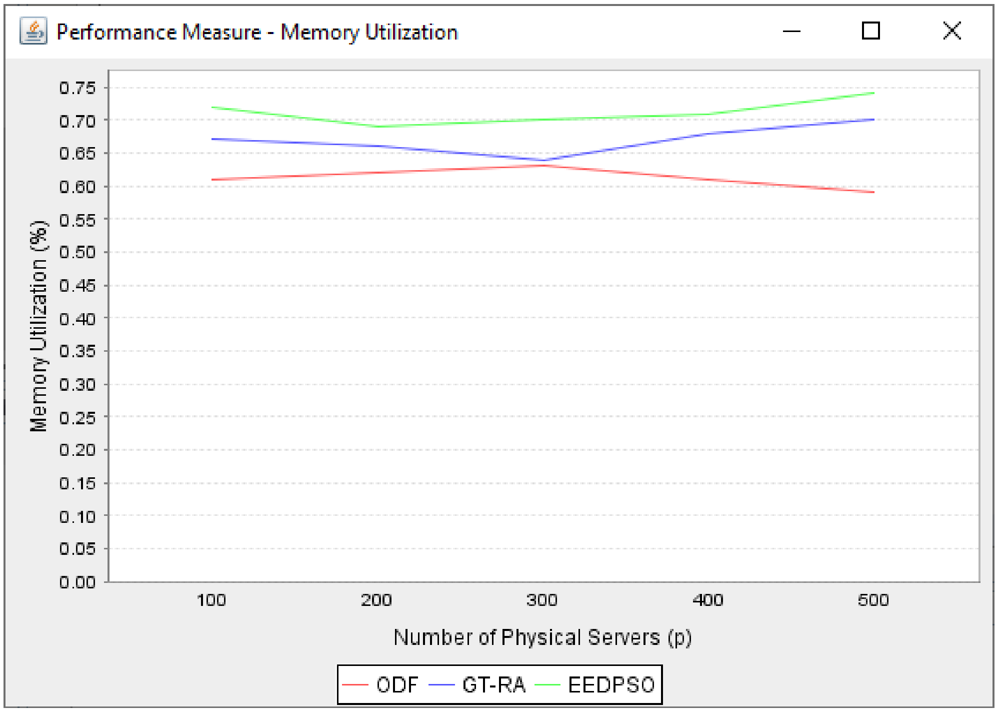


Figure 4 Comparison of memory utilization between the proposed and existing methods

The proposed EEDPSO's memory utilization is lower when compared with GT-RA and ODF technique as shown in figure 4 with the increase in virtual machines order. There will be a slow increase with increase in virtual machines count. There wont be much increase in EEDPSO's memory utilization, when virtual machines count increased to 50 from 30.

For traditional heuristic algorithms like GT-RA and ODF technique, there will be linear variation in memory utilization as shown. With same virtual machines count, high energy efficiency is provided by EEDPSO, when compared with GT-RA and ODF.

Table 3 Disk Utilization

Number of Physical servers (p)	ODF	GT-RA	EEDPSO
100	0.64	0.69	0.72
200	0.62	0.68	0.76
300	0.65	0.68	0.70
400	0.67	0.69	0.75
500	0.63	0.66	0.72

The table.3.shows the disk utilization rate of the proposed and existing method.

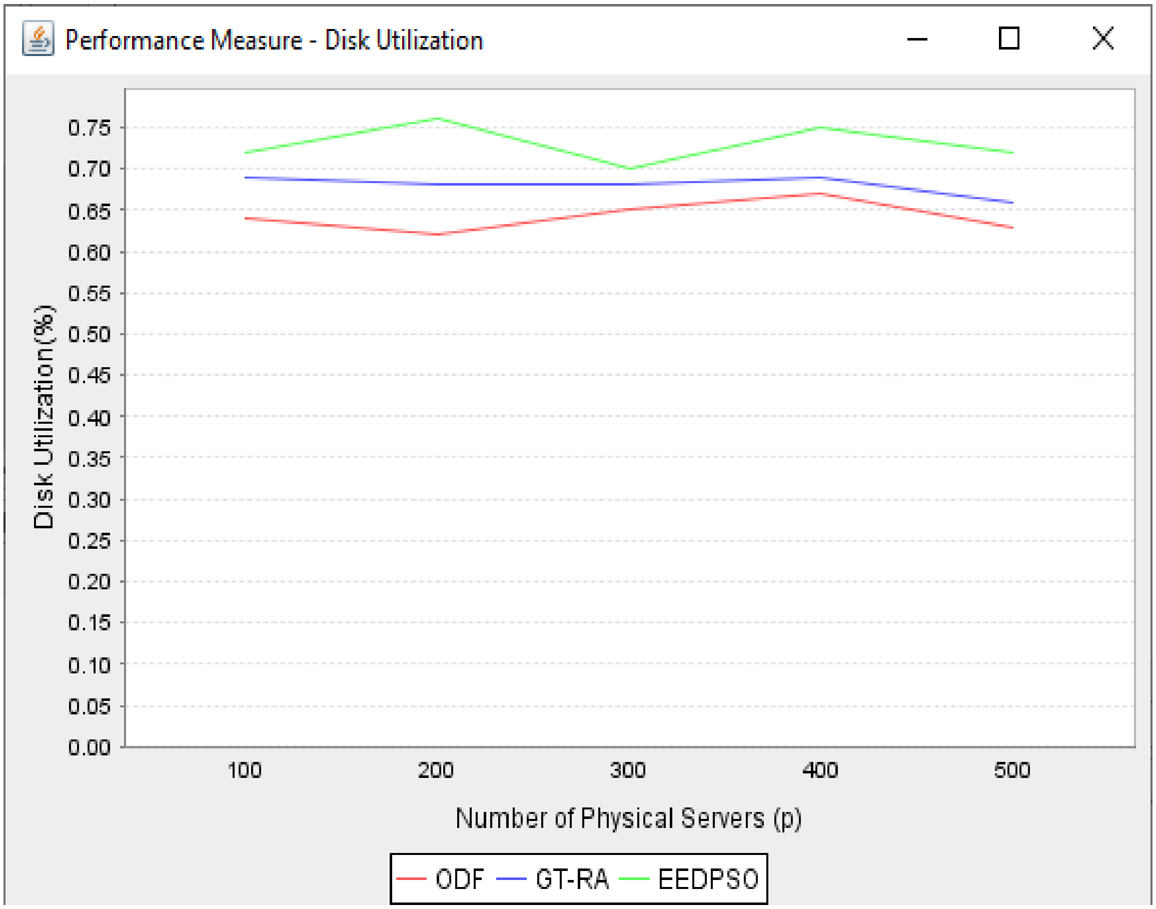


Figure 5 Disk utilization between the proposed and existing methods

With these three algorithms, also investigated the disk usage and Figure 5 shows the investigation results. When compared with GT-RA and ODF, EEDPSO increase disk usage slowly as seen from that. With same virtual machines count, less desk usage is exhibited by EEDPSO when compared with ODF and GT-RA. So, better energy efficiency is produced by proposed technique than GT-RA and ODF, which is a major need of cloud service provider.

Conclusion

To increase an energy efficiency, an allocation algorithm called EEDPSO is proposed on virtual machine level by considering various resources like storage, memory and CPU in this research work. In addition to fair resource allocation to users, proposed algorithm supports effective resource utilization for every physical server. In this work, energy efficiency can be enhanced using a described EEDPSO based virtual machine allocation algorithm with multiple resources.

For evaluating proposed techniques performance, various simulations and experiments re conducted and comparison is made with other related works. In fair allocation, better performance can be achieved using proposed resource allocation as shown in results. In Google cluster, when compared with available allocation mechanism and algorithms, more effective resource allocation can be guaranteed by assigning proper fairness parameters and utilization trade-off. In future research, clod service provider management can be introduced with virtual machines migration algorithm with multiple resources.

References

Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., & Zaharia, M. (2010). A view of cloud computing. *Communications of the ACM*, *53*(4), 50-58.

Buyya, R., Yeo, C.S., Venugopal, S., Broberg, J., & Brandic, I. (2009). Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation computer systems*, *25*(6), 599-616.

Vaquero, L.M., Rodero-Merino, L., Caceres, J., & Lindner, M. (2009). A break in the clouds: towards a cloud definition.*ACM SIGCOMM Computer Communication Review*, *39*(1), 50-55.

Ercan, T. (2010). Effective use of cloud computing in educational institutions. *Procedia-Social and Behavioral Sciences*, *2*(2), 938-942.

Buyya, R., Yeo, C.S., & Venugopal, S. (2008). Market-oriented cloud computing: Vision, hype, and reality for delivering it services as computing utilities. *IEEE 10th international conference on high performance computing and communications*, 5-13.

Marston, S., Li, Z., Bandyopadhyay, S., Zhang, J., & Ghalsasi, A. (2011). Cloud computing—The business perspective. *Decision support systems*, *51*(1), 176-189.

A Vouk, M. (2008). Cloud computing–issues, research and implementations. *Journal of computing and information technology*, *16*(4), 235-246.

Pawar, C.S., & Wagh, R.B. (2012). A review of resource allocation policies in cloud computing. *World Journal of Science and Technology*, *2*(3), 165-167.

Akhter, N., & Othman, M. (2016). Energy aware resource allocation of cloud data center: review and open issues. *Cluster Computing*, *19*(3), 1163-1182.

Jayanthi, S. (2014). Literature review: dynamic resource allocation mechanism in cloud computing environment. *In International Conference on Electronics, Communication and Computational Engineering (ICECCE)*, 279-281.

Sheikholeslami, F., & Jafari Navimipour, N. (2018). Auction-based resource allocation mechanisms in the cloud environments: a review of the literature and reflection on future challenges. *Concurrency and Computation: Practice and Experience*, *30*(16), e4456.

An, B., Lesser, V.R., Irwin, D.E., & Zink, M. (2010). Automated negotiation with de-commitment for dynamic resource allocation in cloud computing. *In AAMAS*, *10*, 981-988.

Pawar, C.S., & Wagh, R.B. (2012). Priority based dynamic resource allocation in cloud computing. *In IEEE International Symposium on Cloud and Services Computing*, 1-6.

Pillai, P.S., & Rao, S. (2014). Resource allocation in cloud computing using the uncertainty principle of game theory. *IEEE Systems Journal*, *10*(2), 637-648.

Fujiwara, I., Aida, K., & Ono, I. (2010). Applying double-sided combinational auctions to resource allocation in cloud computing. *In IEEE10th IEEE/IPSJ International Symposium on Applications and the Internet*, 7-14.

Gouda, K.C., Radhika, T.V., & Akshatha, M. (2013). Priority based resource allocation model for cloud computing. *International Journal of Science, Engineering and Technology Research (IJSETR)*, *2*(1), 215-219.

Arfeen, M.A., Pawlikowski, K., & Willig, A. (2011). A framework for resource allocation strategies in cloud computing environment. *In IEEE 35th Annual Computer Software and Applications Conference Workshops*, 261-266.

Mousavi, S., Mosavi, A., Varkonyi-Koczy, A.R., & Fazekas, G. (2017). Dynamic resource allocation in cloud computing. *Acta Polytechnica Hungarica*, *14*(4), 83-104.

Xiao, Z., Song, W., & Chen, Q. (2012). Dynamic resource allocation using virtual machines for cloud computing environment. *IEEE transactions on parallel and distributed systems*, *24*(6), 1107-1117.

Gunduz, D., & Erkip, E. (2007). Opportunistic cooperation by dynamic resource allocation. *IEEE Transactions on Wireless Communications*, *6*(4), 1446-1454.

Chandra, A., Gong, W., & Shenoy, P. (2003). Dynamic resource allocation for shared data centers using online measurements. *In International Workshop on Quality of Service, Springer, Berlin, Heidelberg*, 381-398.

Xu, X., & Yu, H. (2014). A game theory approach to fair and efficient resource allocation in cloud computing. *Mathematical Problems in Engineering*. https://doi.org/10.1155/2014/915878

Zhou, C., Gao, H. B., Gao, L., & ZHANG, W. G. (2003). Particle Swarm Optimization (PSO) Algorithm [J]. *Application Research of Computers*, *12*, 7-11.