



## Improving the Load Balancing and Dynamic Placement of Virtual Machines in Cloud Computing using Particle Swarm Optimization Algorithm

A. Yousefipour<sup>a</sup>, A. M. Rahmani<sup>\*a</sup>, M. Jahanshahi<sup>b</sup>

<sup>a</sup> Department of Computer Engineering, Science and Research Branch, Islamic Azad University, Tehran, Iran

<sup>b</sup> Central Tehran Branch, Islamic Azad University, Tehran, Iran

### PAPER INFO

#### Paper history:

Received 20 August 2020

Received in revised form 16 November 2020

Accepted 19 January 2021

#### Keywords:

Cloud Computing

Placement

Dynamic Migration

Load Balancing

Particle Swarm

### ABSTRACT

Nowadays, maximizing profits, decreasing operating cost and scheduling tasks are the most important issues of cloud computing with its growing usage. In this regard, one of the challenges in cloud computing is to provide an efficient method to deploy virtual machines on physical machines with the aim of optimizing energy consumption, fair load distribution and task scheduling. The purpose of present study is to provide a method for improving task scheduling through an improved particle swarm optimization algorithm. In the proposed method of present study, selection of a proper objective function has led to balanced workload of virtual machines, decreased time of all tasks as well as maximum utilization of all resources and increased productivity in addition to dynamic placement of virtual machine on physical machine. The results of simulation showed that the proposed method has provided an optimized solution for scheduling tasks, equal allocation of tasks in virtual machines and placement on the appropriate physical machine and less time with an improvement of 0.02 has been spent on the process of outsourcing virtual machines.

doi: 10.5829/ije.2021.34.06c.05

## 1. INTRODUCTION

The third revolution in information technology after computer technology and the internet has been occurred with cloud computing [1]. This technology has led to a decreased in the cost of technology infrastructure due to its large data centers [2]. Nowadays, cloud computing has become a part of the market because many organizations and companies do not have the necessary resources due to the high cost of software and hardware. To overcome this problem, cloud computing practically provides the required software and hardware at an affordable rental cost [1]. These resources enjoy from the feature of remote access and users can access them from anywhere in the world. In the term of placement, cloud computing can be classified into four categories of public, private, hybrid and social [3]. In cloud computing, resources are provided to users as services [4]. These services can be categorized into three categories of software as a service,

platform as a service, and infrastructure as a service [3, 2]. In infrastructure, this service is provided for client through hardware virtualization, which includes many suggestions such as virtual server space, network connections, bandwidth, IP addresses and load balancer [3]. In data centers, there are a significant number of physical and virtual machines with a large number of possible metrics [5], which resulting in a varied need for different resources. The efficient and dynamic use of shared resources through workloads varying from time to time [6]. Virtualization technology in a cloud data center eliminates server heterogeneity and provides server integration and enhances the efficiency of server usage [7]. Virtualization makes it possible to map several virtual machines on a physical machine with varying levels of access and performance quality [8]. However, one of the important challenges in cloud computing is to provide an efficient way for mapping of virtual machines on physical machines so that they can meet the quality of

\*Corresponding Author Institutional Email: [rahmani@srbiau.ac.ir](mailto:rahmani@srbiau.ac.ir) (A. M. Rahmani)

service requirements; may required by service providers in addition to maximum usage of physical machines [9, 6]. Virtual machine selection and mapping on any virtualized physical machine is known as virtual machine placement [9]. In this case, the virtual machine will migrate to the next most appropriate host when the host on which virtual machine is placed fails to provide the increasing demand-driven services [3]. Virtual machine migration in data centers helps to achieve goals such as balanced load distribution, fault tolerance (FT), energy management and decreasing service response time [10]. Once virtual machines are placed in a physical machine, the host of virtual machine provides different resources with different workloads at different times. This matter can lead to resource imbalances and a decline in the performance of cloud services. As a result, the virtual machine must migrate on the physical machine. To overcome such challenge, the overload of a virtual machine is removed and transferred to a machine with a lower-load in order to balance the load [11]. Load balancing is a technique for balanced distribution of load across resources [12]. Solving virtual machine placement and load balancing challenges is a type of nondeterministic polynomial time (NP)-Hard problem. Solving an NP-Hard problem through common methods such as Graph Theory is time-consuming and very costly in the term of computations. Therefore, it is better to solve such problems through other methods using meta-heuristic and heuristic techniques. meta- heuristic methods include particle swarm optimization (PSO) algorithm, ant colony optimization algorithm, genetic algorithm, neighborhood search algorithm and Tabu search algorithm [13]. Among the available meta-heuristic methods in this research, a method for optimization will be selected that has more optimal solutions and less computational cost than other algorithms.

Therefore, according to studies, PSO has been selected and the purpose of present study was to propose a method of virtual machines placement on physical machines to optimize energy consumption, balance of workload for fairly distribution of tasks, decreased time of completing all tasks and increase resource efficiency.

The proposed method and algorithm and its mathematical topics have been described in section three and the conclusion of present study has been provided in section five.

## 2. RELATED WORKS

Available solutions for virtual machines placements are utilizing from definitive algorithms include finite programming, linear programming, integer programming, and dynamic programming. Utilizing from heuristic algorithms for a large number of physical

and virtual machines is impractical. Most of the heuristic-based articles on the algorithms such as case reduction algorithm, the best case algorithm, the first case algorithm, the heaviest case algorithm and the worst case algorithm have been conducted properly. Meta-heuristic algorithms are desirable to obtain good solutions at the right time. Among such algorithms, it can be referring to Memetic algorithm at an extension of the traditional genetic algorithm, particle swarm optimization (PSO) algorithm, ant colony optimization algorithm, genetic algorithm, neighborhood search algorithm and Tabu search algorithm [13]. Meta-heuristic, genetic, particle swarm optimization (PSO) and ant colony optimization algorithms have been also used to balance the load on virtual machines [14, 15]. In this section, the previous studies conducted on virtual machine placement as an important issue for cloud providers have been investigated.

Dong et al. [16] have suggested multi-resource constraint of physical machine for the problem of virtual machine placement. In this suggestion, the traffic model between virtual machines can be changed by minimizing all the traffic through placement of virtual machine in the data center and placing the high traffic virtual machine in the same physical machine.

Liu et al. [17] have proposed the multi-objective algorithm NS-GGA to solve the problem of virtual machine placement through a non-dominated sorting genetic algorithm. The proposed method attempts to achieve the optimal Pareto front based on non-dominated sorting and grouping of genetic operators. It is necessary for infrastructure providers to specify virtual machine places. The proposed algorithm utilizes from fuzzy logic system to combine multiple objectives into one function in order to solve the problem.

In fact, several resource allocation algorithms have been suggested for efficient energy management in the cloud environment [18]. The modified best fit decreasing (MBFD) algorithm focuses on locating virtual placement and deals with sending new virtual machine requests and assigning them to hosts. In this method, threshold, minimum and maximum utilizations of the hosts are used to optimize the current virtual machines.

Tordsson et al. [19] have been presented virtual infrastructures for placement optimization according to the criteria specified by users in different clouds. The proposed algorithm is on the basis of integer programming formula and cost reduction.

The main point in virtual machine placement is to consider the CPU and memory of the physical machine on which the virtual machine is placed. In previous studies, this important point has not been taken into account in virtual machine and unused physical machine kept in idle or stand-by mode. Several studies have ever been conducted on workload balancing. In general, load balancing algorithms are divided into two groups of static

and dynamic. In static method, allocation of tasks to virtual machines is based on the capabilities of virtual machine and initial state of each machine. In dynamic method, the distributor allocates the tasks to virtual machine based on its current state and its available workload in addition to the initial capabilities of each virtual machine [12]. For this reason, only the studies conducted on dynamic load balancing have been investigated in this section.

Song et al. [20] have been proposed an online packet classification-based algorithm called VISBP that dynamically allocates data center resources through live migration. The proposed algorithm is efficient in real resource allocation systems. The method has been designed implemented in CloudSim environment, which saves energy and decreases the number of active physical machines.

Alguliyev et al. [11] have considered the load balancing of virtual machines as a NP-Hard problem. In this article, a particle swarm-based algorithm called  $\alpha$ PSO-TBLB has been proposed for load balancing.

Abdi et al. [21], have proposed a scheduling method using particle swarm optimization algorithm with the aim of decreasing the time of completing all tasks. In the proposed method, the particle swarm optimization algorithm has been improved through the algorithm of assigning the shortest task to the fastest CPU.

Finally, the proposed method has been comprised with genetic algorithm. Acharya et al. [12] Agnihotri, and Sharma [22] have been stated that load balancing is one of the important challenges of virtual machines in the cloud environment. In these references, live migration via a particle swarm-based algorithm has been performed to solve the problem of load balancing in virtual machines.

Here are some tasks related to multi-objective optimization:

Feng et al. [23] have introduced to diversify the needs of users in cloud computing, it is necessary to provide an efficient strategy in data centers. So virtualization is one of the most important aspects of cloud computing. Integrating resources for a single purpose and reducing energy consumption in cloud computing is possible. However, integrating resources using multi-objective optimization is challenging. In this research, a combined exploratory algorithm of PSO and ACO is presented in order to reduce costs and energy consumption. The PSO algorithm is used for the virtual machine positioning speed and the ACO is used for the virtual machine position. The simulation results showed that it has efficiently allocated dynamic resources.

Cloud computing enables users to use resources (hardware, software and operating system) through the network and virtualization technology as a major part of cloud computing [24]. The virtual machine in cloud computing is placed on the physical machine. But in a

virtual machine, a few things like memory, processor, bandwidth are considered and energy consumption should be minimized. In a virtual machine, multi-objective algorithms are used when multiple goals are optimized. In this research, four algorithms NSGA-II, eMOEA, PAES and SPEA2 were used to locate and minimize energy consumption and the environment is clodsim simulation. The results showed that the NSGA-II algorithm performed better than the others.

Cloud computing offers a variety of services. Data centers in cloud computing have physical machines with millions of virtual machines on them [25]. Therefore, in this situation, load balance on any virtual machine is very important, which is a challenge for service providers in cloud computing. In this paper, two algorithms for load balancing are compared. The proposed algorithm is gray wolves (GWs) optimization and then the results of this algorithm are compared with the next multi-objective particle swarm optimization algorithm. The results showed that the proposed algorithm has shown good results in many experiments, but the multi-objective particle mass algorithm had better results than the proposed algorithm in some similar experiments.

In reference There are countless heterogeneous virtual machines in cloud computing data centers [26]. Therefore, the items that should be considered are energy consumption, resource usage, usage time and virtual machine location. Therefore, this paper presents a multi-objective optimization method for power consumption reduction (SLAV) called multiobjective combines salp swarm and sine-cosine algorithms (MOSSASCA). The results showed that the proposed method performed better than MOPSO, NSGA-II, MOEAD and MOSCA methods. In this paper, after the proposed algorithm, the MOPSO algorithm performed better than other algorithms.

Load balance is a very important task in scheduling tasks that directly affect cloud computing resources [27]. So this article is a nature-inspired method called Binary Load Balancing – Hybrid Particle Swarm Optimization and Gravitational Search Provides Algorithm (Bin-LB-PSOGSA). The results show that the BinLB-PSOGSA algorithm has a better result than Bin-LB-PSO.

#### **Discussion:**

As can be seen in the related articles, a variety of single-objective and multi-objective optimization algorithms have been used to optimize energy consumption, load balance, virtual machine placement, and so on.

This research includes single-objective and multi-objective optimization and particle swarm optimization algorithm has been used for optimization because this algorithm has more optimal solutions and less computational cost than other algorithms. Therefore, this article pursues several goals. The first goal is a method of virtual machines placement on physical machines to

optimize energy consumption; the second goal is balance of workload for fairly distribution of tasks and the third goal is to decrease time of completing all tasks and increase resource efficiency. To achieve these three goal, the proposed method is convergence and based on the improved particle swarm optimization, which avoids linearization by selecting appropriate objective functions and allocates the weights to indicate priority. In the proposed method, it has been prevented from linearization and premature convergence of objective function using particle swarm optimization algorithm and considering a variable called alpha with a value between 0 and 1. The better results can be obtained at the optimum time by assigning this weight to each of the objective criteria. Finally, the fourth goal is to optimize the previous three goals using the multi-objective particle swarm optimization (MOPSO) algorithm. The reason for this choice is that it can be seen in studies that this algorithm performed better than similar algorithms such as NSGA-II.

But A new and innovative aspect of the proposed method is the simultaneous optimization of target functions using a multi-objective particle swarm optimization and an improved particle swarm optimization, which has not been done in previous studies with this scope. Therefore, in this study, the objective is to optimize the response variables of the three functions and find the best combination of variable regulation using multi-objective PSO.

### 3. METHODOLOGY AND PROPOSED ALGORITHM

As it was mentioned in the introduction, the considered cloud coputing challenge in present study was proper placement of virtual machines on physical machines using CPU and memory; then, load balancing on virtual machines to prevent migration and completing all tasks of virtual machines in an appropriate scheduling. Therefore, the proposed method of present study has been aimed to right placement of virtual machine on the physical machine based on scheduling tasks to maximize profit as well as decreasing operating costs and preventing performance loss in service delivery. The procedure of the proposed method based on proposed algorithm is shown in Figure 1. In the first step for the first goal, virtual machine was placed on physical machine based on the amount of CPU and memory and the rest of unused physical machine was kept in idle or standby mode. In the second step for the second goal, the load balancing of virtual machines was created based on task scheduling and task execution time. In the third step for the third goal, the purpose was to complete all tasks on a virtual machine to increase efficiency by completing time of tasks. It should be noted that all three steps were

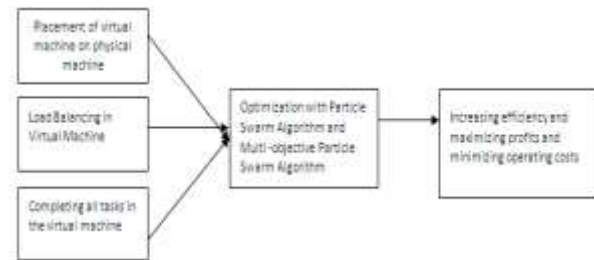


Figure 1. The proposed method of present study

finally for the fourth goal tested simultaneously using multi-objective particle swarm optimization algorithm.

#### 3. 1. The Proposed Algorithm

The particle swarm optimization algorithm was proposed by Kennedy and Eberhart [28] whose idea was discovered by simulating a simple swarm model. The method of particle optimization is rooted in Reynold's work. The concept of Rooster was also added to this model in order to its further development, which was in the form of a memory of the best positions of each member and its neighbors. Each particle has a position indicating its coordinates in the multidimensional search space. The position of particle changes as it moves over time.  $(X_i(t))$  indicates the position of particle  $i$ th at time  $t$ th. Also, each particle needs a velocity to move through space.  $(V_i(t))$  indicates the velocity of particle  $i$ th at time  $t$ th. A new position can be created for each particle by increasing the velocity of its position. The way of updating particle position is presented in following equation (1):

$$X_i(t+1) = X_i(t) + V_i \quad (1)$$

Whether a particle's position in the search space is a suitable position is evaluated using fitness function. Particles have the ability to memorize the best position they have been in their lifetimes. The best individual experience of a particle or the best visited position of particle is referred as  $y_i$  ( $y_i$  has been also named as  $pbest$  in some algorithms). This position is referred as  $\widehat{y}_i$  ( $\widehat{y}_i$  has been also named as  $gbest$  in some algorithms). The particle velocity vector in optimization process indicates the empirical knowledge of particle, particle information and particle society information. Each particle considers two components to move in search space including: 1) cognition component indicating the best solution a single particle can obtain ( $pbest$ ) and 2) social component indicating the best solution recognized by the whole group ( $gbest$ ), which were mentioned in the previous section. Considering  $pbest$  and  $gbest$ , each particle utilize from following equations (2), (3) and (4) to determine its next position:

$$V_{ij}(t+1) = WV_{ij}(t) + c_1r_1(p_{ij}(t) - X_{ij}(t)) + c_2r_2(p_{ij}(t) - X_{ij}(t)) \quad (2)$$

$$X_{ij}(t+1) = X_{ij}(t) + V_{ij}(t+1) \quad (3)$$

$$W(t+1) = w_{max} - \left(\frac{w_{max}-w_{min}}{t_{max}}\right) \cdot (t+1) \quad (4)$$

where, (i) is the particle index. The constants of ( $c_1$ ) and ( $c_2$ ) determine learning parameters (impact rate) for  $pbest$  and  $gbest$  are usually considered equal to 2. ( $r_1$ ) and ( $r_2$ ) are random numbers in the range of [0 and 1]. ( $X_{ij}(t)$ ) represent the current position of particle. ( $V_{ij}(t)$ ) represent the velocity of particle. The parameter W controls the velocity of particles which speeds up y at the beginning of algorithm implementation and slows down over time by approaching the response [28, 29]. The variables of algorithm have been presented in Table 1.

### 3. 2. Mathematical Model of Algorithm and Objective Function

If m is the number of virtual machine in cloud environment, then we have  $VM = \{VM_1, VM_2, \dots, VM_m\}$  and if m virtual machines are available in p mechanical machine  $PM = \{pm_1, pm_2, \dots, pm_p\}$ , then Equation (5):

$$where\ n_p = \begin{cases} 1 & \text{If PM is active} \\ 0 & \text{Otherwise} \end{cases} \quad (5)$$

where, ( $n_p$ ) is a binary value indicating the active or inactive mode of PM. (n) The implementable tasks on virtual machine  $Task = \{T_1, T_2, \dots, T_n\}$

#### 3. 2. 1. First Goal: Objective Function of Virtual Machine Placement

The workloads of virtual and mechanical machines should be first considered for right virtual machine placement. In cloud computing, the amount of CPU and memory has a critical effect on the system. Therefore, it is necessary to define the workload in virtual and physical machines. By combining the load of CPU and memory of virtual machines or physical machines together, a new criterion is defined in Equation (6) called the volume which denoted by U [30] stated as follows:

$$U = \frac{1}{1-cpu} * \frac{1}{1-mem} \quad (6)$$

**TABLE 1.** the variables of particle swarm optimization algorithm

Variables of algorithm	
W	Internal weight
$c_1, c_2$	Acceleration Coefficient
$r_1, r_2$	random numbers in the range of [0 and 1]
$X_{ij}(t)$	Current position of each particle
$V_{ij}(t)$	Current velocity of each particle
$pbest$	The best position that particle has ever received
$gbest$	The best position in all particles

In fact, the CPU and memory are resources used by virtual machine or physical server, which normalized through the number of CPUs and attributed memory size. The above equation indicates that the volume is increased with increase in the usage of a resource (CPU and memory). The ith CPU and ith memory are the CPU and consumed memory of virtual machine  $V_i$  and the pth CPU and pth memory are current CPU load and memory of physical machine p. If all of the k virtual machines are placed on the physical machine p, then the total load of p is obtained using below Equation (7):

$$U_p = \frac{1}{1-(cpu_p + \sum_{i=1}^k cpu_i)} * \frac{1}{1-(mem_p + \sum_{i=1}^k mem_i)} \rightarrow$$

$$F_1 = \min \left( \frac{1}{1-(cpu_p + \sum_{i=1}^k cpu_i)} * \frac{1}{1-(mem_p + \sum_{i=1}^k mem_i)} \right) \quad (7)$$

$$\forall p = 1, \dots, p \quad \forall i = 1, \dots, k$$

( $\sum_{i=1}^k cpu_i$ ) and ( $\sum_{i=1}^k mem_i$ ) are the total CPU and used memory of virtual machines, respectively. Also, better results can be obtained by assigning weight instead of one on the numerator of Equation (8) and generally the CPU and memory load of a physical machine is not more than 100%, so the following equation holds.

$$cpu_p + \sum_{i=1}^k cpu_i < 1 \quad mem_p + \sum_{i=1}^k mem_i < 1 \quad (8)$$

#### 3. 2. 2. Second Goal: Objective Function of Load Balancing

The optimal value between the tasks and virtual machine should be determined based on the objective function in order to obtain scheduling and the best transfer time. Here, minimizing the task execution time ( $T_{exe}$ ) and task transfer time ( $T_{trans}$ ) have been considered as objective function [11]. It is calculated as Equation (9):

$$T_{exe} = \sum_{i=1}^n \sum_{k=1}^m x_{ik} \frac{DE_i}{VM_{m_k}} = \frac{\text{number operations}}{\text{number operations/sec}} \quad (9)$$

$$\forall i = 1, \dots, n \quad \forall k = 1, \dots, m$$

Where, ( $DE_i$ ) represent the work load of ith task (number of operations performed by task). ( $VM_{m_k}$ ) represent is the speed of CPU (number of operations per second) in kth virtual machine. M is the number of virtual machine and n is the number of tasks. Also, ( $x_{ik}$ ) is ith task on kth virtual machine with Equation (10):

$$T_{trans} = \sum_{i=1}^n \sum_{k=1}^m \sum_{z=1}^m (1 - x_{ik} x_{iz}) \frac{DT_i}{B_{kz}} \quad (10)$$

$$\forall i = 1, \dots, n \quad \forall k = 1, \dots, m$$

where, ( $DT_i$ ) is the volume of data exchange between kth and zth virtual machines and ( $B_{kz}$ ) is the bandwidth between kth and zth virtual machines. If ith task is on zth virtual machine, the value of  $x_{iz}$  is equal to 1 otherwise its value is 0. Better results can be obtained at the optimum time by weighting each criterion of objective function. Therefore, the following Equation (11) is obtained:

$$F_2 = \min(\alpha T_{exe} + (1 - \alpha)T_{trans}) \quad \forall \alpha = 0. \dots 1 \quad (11)$$

**3. 2. 3. Third Goal: Objective Function of Energy Efficiency**

In this step, the purpose of algorithm was to decrease the completing time of all tasks and increase the efficiency of resources. The completing time of all tasks I defined through following Equation (12) procedure [26].

$$\begin{aligned} & \text{Total time to complete all tasks} = \\ & \max \sum T_i VM_m * x_{im} \quad (12) \\ & \forall i = 1. \dots n \end{aligned}$$

where,  $(T_i VM_m)$  is the completing time of all ith tasks on mth virtual machine.  $(x_{im})$  is equal to 1 when ith task is placed on mth virtual machine, otherwise its value is equal to 0. According to this equation, among a number of virtual machines between 1 to m, the value of machine with the highest total completing time of all assigned tasks is considered as the completing time of tasks. The value of efficiency from each resource is calculated using below Equations (13):

$$\begin{aligned} \text{Utilization}_j &= \frac{\sum T_i VM_m * x_{im}}{\text{Total time to complete all tasks}} \quad (13) \\ & \forall i = 1. \dots n \end{aligned}$$

where, the efficiency y of source j is equal to the sum of all tasks assigned to mth virtual machine by the processing ratio of machine with the highest total processing time. It is clear that the efficiency of each virtual machine is less than or equal to one stated in Equation (14).

$$\text{AvgUtilization} = \frac{\text{Utilization}_i}{m} \quad \forall i = 1. \dots n \quad (14)$$

As a result, the objective function is obtained by decreased completing time of tasks and increased efficiency of resources with Equation (15):

$$f3 = \min \left( \frac{\sum T_i VM_m * x_{im}}{\text{Avgutilization}} \right) \quad \forall i = 1. \dots n \quad (15)$$

**3. 2. 4. Fourth Goal: Multi-objective Particle Swarm Optimization (MOPSO)**

At this stage, the simultaneous algorithm of the objective function of Equations (7), (11) and (15) is optimized.

The pseudo-code of particle swarm optimization algorithm is presented in Table 2.

**4. EVALUATION AND SIMULATION**

In present study, the particle swarm optimization algorithm has been used to optimize the virtual machines. The simulation environment of MATLAB R2015a software was used for modeling and the computer with Intel i3-2350 series processor, 4GB of RAM and 500GB of hard drive used for simulation.

**TABLE 2.** the pseudo-code of particle swarm optimization algorithm

---

**Input:**  $VM = \{VM_1, VM_2, \dots, VM_m\}, PM = \{pm_1, pm_2, \dots, pm_p\}, Task = \{T_1, T_2, \dots, T_n\}$

**Output:**  
 best position of Tasks on the VMs(Gbest) and best position of placement VMs on the PMs(Gbest) and best position of average utilization on the VMs(Gbest)

**Start:**  
 1: Set particle dimensionas equal to the size of ready tasks, placement and average utilization.  
 2: Initialize particles position randomly and velocity  $v_i$  randomly.  
 3: for each particle run load balancing and placement algorithm for balance and placement particles position  
 4: For each particle, calculate its fitness value as in Equation(7,18 and 26) and set the alpha value  
     If (fitness value < previousbestpbst)  
         set the current fitness value as the newpbst  
 5: After Steps 4 for all particles, select the best particle as gbest  
 6: For all particles, calculate velocity using Equation (2) and update their positions using Equation (3)  
 7: If (stopping criteria or maximum iteration is not satisfied) repeat from Step 4.  
     else  
         Return gbest

---

**Stop.**

---

The simulations and steps of particle swarm optimization algorithm have been described below:

**Step 1: Random selection of particles initial population:** random generation of initial population includes random determination of particles initial place through a uniform distribution in the solution space (search space). The initial proposed range for particle velocity can be extracted through equation (16).

$$\frac{X_{min} - X_{max}}{2} \leq V \leq \frac{X_{max} - X_{min}}{2} \quad (16)$$

**Step 2: Select the number of initial particles:** considering the problem, the initial population was determined equal to 50. Increasing number of initial particles leads to a decrease in the number of iterations required for algorithm convergence.

**Step 3: Evaluation of particles objective function (calculation of the cost or fitness):** in this step, each of the particles should be evaluated as a solution for problem. In present study, three objective functions have been considered including relations (7), (11) and (15). A variable called alpha has been defined for these functions in the range of 0 and 1 in order to prevent linearization and premature convergence. Better results can be obtained at the optimum time by allocating this weight to each of the objective criteria.

**Step 4: Record the  $(P_{i.best})$  and the  $(P_{g.best})$  and update:** in this step, two steps are possible based on the repetition number: (a) in the case of being in first repetition ( $t = 1$ ), the current position of each particle is considered as the best specified place for it.

$$\begin{aligned} P_{i.best} &= X_i(t) \quad .i = 1. 2. 3. \dots d \\ \text{cost}(P_{i.best}) &= \text{cost}(X_j(t)) \quad (17) \end{aligned}$$

In other repetitions, the value obtained for particles in step 2 is comprised with value of the best cost value obtained for each particle. If this cost is less than the best recorded cost of this particle, then the place and cost of this particle is replaced with previous one. Otherwise no change is created in the recorded place and cost of this particle. It means that:

$$\begin{cases} \text{if } \text{cost}(X_i(t)) < \text{cost}(P_{i,\text{best}}) \Rightarrow \\ \quad \text{else Not change} \\ \text{cost}(P_{i,\text{best}}) = \text{cost}(X_j(t)) \\ P_{i,\text{best}} = X_i(t) \end{cases} \quad (18)$$

$i = 1.2.3. \dots d$

The velocity vector update of all particles has been made according to the Equation (2) and the considered parameters have been based on literature [18].

**Step 5: Algorithm stopping condition:** a constant value has been considered for algorithm stopping condition. The simulation parameters and proposed algorithm have been presented in Table 3.

**First Goal:** According to Equation (7), all of virtual machines were placed on a physical machine and migrated to a physical machine in order to optimize the first objective function. The process of this optimization has been presented in Figure 2.

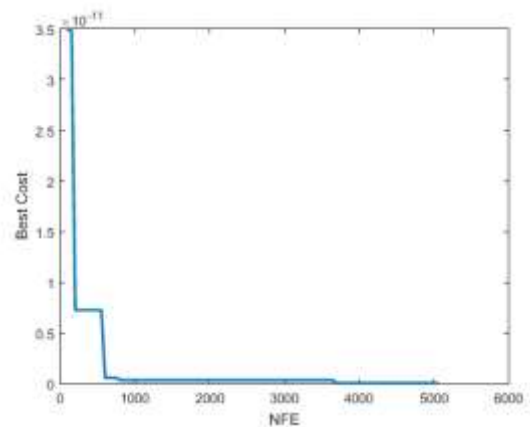
As it can be seen from Figure 4, the best objective function value was  $3.9 \cdot e^{-9381}$  and the objective function calling number for optimal response was approximately 4000 times and then, it has been converged. Finally, virtual machines were placed on the ninth physical machine in the last repetition. The number of these placements has been presented in Figure 3.

**Second Goal:** According to Equation (11), the second objective function was optimized for load balancing of virtual machines. The process of this optimization is shown in Figure 4.

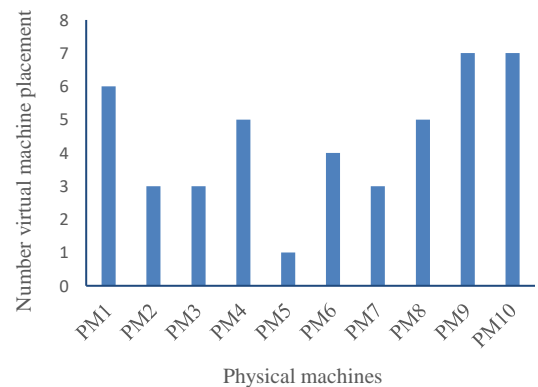
**TABLE 3.** the simulation parameters

Simulation parameters and particle swarm optimization algorithm		
<b>Particle swarm optimization algorithm</b>	Size of initial population	50
	Number of repetitions	100
	Internal weight	0.7
	Acceleration coefficient	1.49445
	$r_1$ and $r_2$	Random numbers in the range of [0 and 1]
<b>Physical machine</b>	Number of physical machines	10
	CPU	Random numbers in the range of [1000 and 8000]

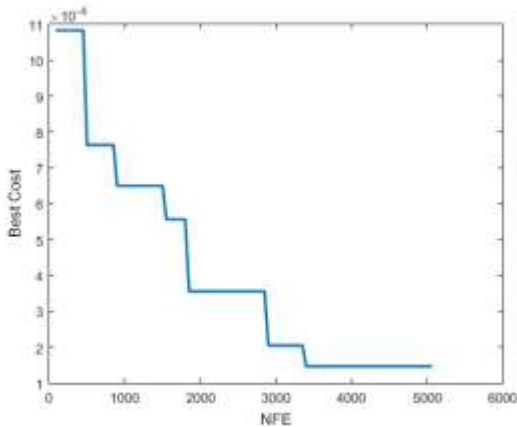
<b>Virtual machine</b>	Memory (Gig)	Random numbers in the range of [1 and 8]
	Number of virtual machines	5
	CPU	Random numbers in the range of [1000 and 8000]
	Memory (Gig)	Random numbers in the range of [1 and 8]
	Number of tasks	11
	Bandwidth (megabits per second)	Random numbers in the range of [1 and 1024]
	Work load of task	Random numbers in the range of [1 and 100]
	CPU speed (number of operations per second)	Random numbers in the range of [1000 and 10000]
	Alpha	Random numbers in the range of [0 and 1]



**Figure 2.** Optimization function of virtual machine placement



**Figure 3.** Number of virtual machines placements on physical machine



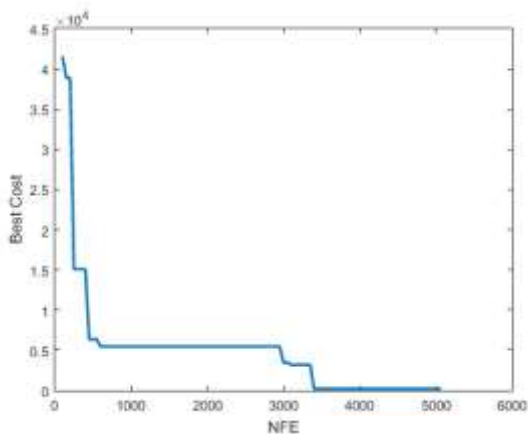
**Figure 4.** Optimization function of virtual machines' load balancing

As it can be seen from Figure 4, the best value of objective function was 0.00014644, the best time to transfer and execute tasks was close to 0.1 and the objective function calling number for optimal response was approximately 3500 times and then, it has been converged.

**Third Goal:** According to Equation (15), the third objective function was optimized for energy consumption optimization on virtual machines. The process of this optimization is presented in Figure 5. Whatever the value of this relation is lower, the utility is higher in the terms of decreased time of completing all tasks and increased efficiency of resources.

As it can be seen from Figure 5, the best objective function value was 72.122 and the objective function calling number for optimal response was approximately 3500 times and then, it has been converged.

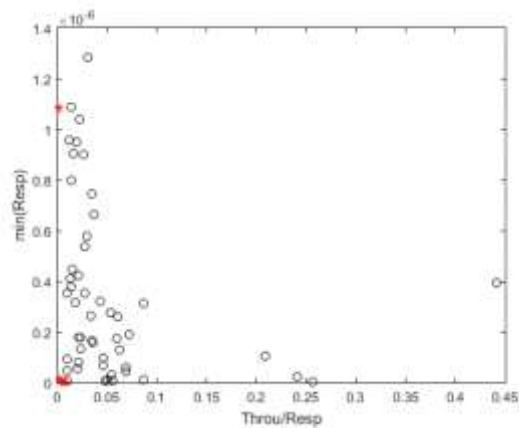
**Fourth Goal:** Then, the multi-objective particle swarm optimization algorithm was used to simultaneously optimize all three objective functions.



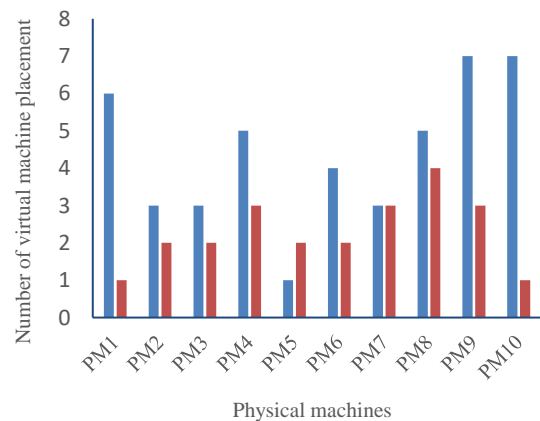
**Figure 5.** Optimization function of virtual machines efficiency

This algorithm is similar to the basic particle swarm optimization algorithm with the difference that objective function and particles are updated based on the best memory of each particle. In this algorithm, the non-dominated of initial population are separated after generation of initial population and stored in the Repository (Rep). Then, the discovered objective space is tabulated and each particle chooses a leader from Rep members and makes its move. Then the best personal memory of each particle is updated and non-dominated members of current population are added to Rep and remove the dominated members of Rep. If Rep members exceed the specified capacity, the extra members will be deleted and ultimately the algorithm is finished with specified repetitions of stopping condition. The optimized places of all three objective functions have been shown in Figure 6.

A comparison between the results of present study and the results of other studies conducted on this field has been presented as follows: Figure 7 represents a comparison between the results of present study and



**Figure 6.** Simultaneous optimization of three objective functions



**Figure 7.** Comparison of virtual machines placement on physical machines



reported data in literature [30] in the term of virtual machine placement.

In this figure, the placement of virtual machine on physical machine is different in both studies. In both studies, the trend of number of virtual machine placements on physical machine has been well illustrated by particle swarm optimization algorithm. Figure (8) represents a comparison between the results of present study and literature [11] in the term of load balancing on virtual machines using task execution time and transfer time.

As it can be seen from Figure 8, the results of both studies are close to each other. In this study, the reason of obtaining better results with an improvement of 0.02 is proper calculation of alpha value. Figure 9 represents a comparison between the results of present study and literature [31] in the term of energy consumption optimization on virtual machines to decrease the completing time of all tasks and increase efficiency.

As it can be seen from Figure 9, the amount of efficiency with the same number of tasks is slightly better for present study compared to literature [31]. The reason is that the utility is higher in the terms of decreased time of completing all tasks and increased efficiency of resources whatever the value of this relation is lower and a lower and more optimized value was obtained in present study. Figure 10 represents the total completing time of all tasks.

As it can be seen form Figure 10, the total completing time of all tasks for present study and literature [31] is approximately equal because both studies have had a similar proposed model. However, a better result has been obtained in present study by determining a proper value of alpha. The significant difference between the results of present study and results of literature [21] is due to the fact that the reported data [21] and its coding and the way of its implementation can be different. It should be noted that the particle swarm optimization algorithm was comprised with genetic algorithm, which

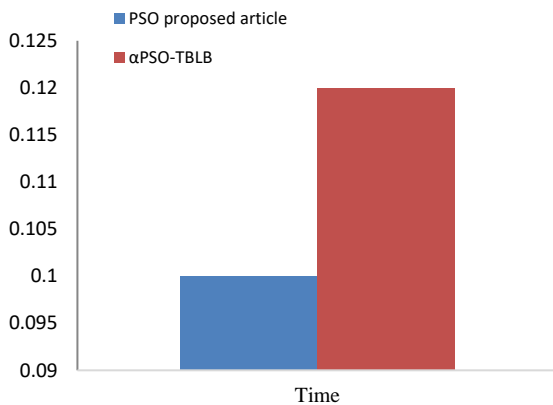


Figure 8. Comparison of load balancing in physical machine

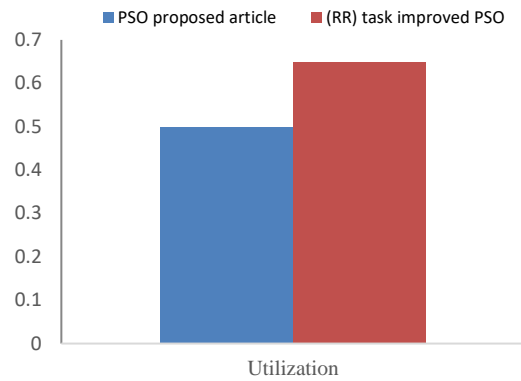


Figure 9. Comparison of physical machine efficiency

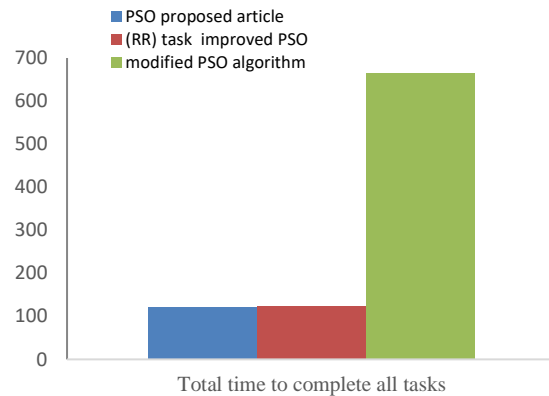


Figure 10. Comparison of total completing time of all tasks in physical machine

the comparison indicated the superiority of particle swarm optimization algorithm to genetic algorithm. In general, the difference point of present study compared to other conducted studies that can be also considered as a new and innovative aspect of present study is simultaneous optimization of objective functions using a multi-objective particle swarm optimization. For this reason, the MATLAB environment was more suitable for implementation of this algorithm.

### 5. CONCLUSION

The purpose of present study was to propose a method using a meta-heuristic approach as well as multi-objective particle swarm optimization and particle swarm optimization algorithms in order to increase profit and performance, decrease operating costs and optimize the utilization of resources in cloud computing. In this article, several goals have been pursued. The first goal a method of virtual machines placement on physical machines to optimize energy consumption is the second

goal balance of workload for fairly distribution of tasks and the third goal decreased time of completing all tasks and increase resource efficiency. To achieve these three goal, the proposed method is convergence and based on the improved particle swarm optimization, which avoids linearization by selecting appropriate objective functions and allocates the weights to indicate priority. Finally, the fourth goal is to optimize the previous three goals using the multi-objective particle swarm optimization (MOPSO) algorithm. The better results can be obtained at the optimum time by assigning this weight to each of the objective criteria. The results of simulation showed that the proposed method has provided an optimized solution for scheduling tasks, equal allocation of tasks in virtual machines and placement on the appropriate physical machine and less time with an improvement of 0.02 has been spent on the process of outsourcing virtual machines. So the proposed methods can make it possible to obtain the most efficiency with minimum utilization of resources.

## 6. REFERENCES

1. Kaur, G. Sharma, S. "Optimized Utilization of Resources Using Improved Particle Swarm Optimization Based Task Scheduling Algorithms in Cloud Computing", *International Journal of Emerging Technology and Advanced Engineering*, Vol. 4, No. 6, (2014), 110-115.
2. Askarizade H, M. Maeen, M. Haghparast, M. "An Energy-Efficient Dynamic Resource Management Approach Based on Clustering and Meta-Heuristic Algorithms in Cloud Computing IaaS Platforms", *Wireless Personal Communications*, Vol. 104, No. 4, (2018), 1367-1391, doi: <https://doi.org/10.1007/s11277-018-6089-3>.
3. Kumar, A. Sathasivam, C. Periyasamy, P. "Virtual Machine Placement in Cloud Computing", *Indian Journal of Science and Technology*, Vol. 9, No. 29, (2016), 1-5, doi: [10.17485/ijst/2016/v9i29/79768](https://doi.org/10.17485/ijst/2016/v9i29/79768).
4. Goscinski, A. Brock, M. "Toward dynamic and attribute based publication", *Future Generation Computer Systems*, Vol. 26, (2010), 947-970, doi: <https://doi.org/10.1016/j.future.2010.03.009>.
5. Pires, F. L. Baran, B. "A Virtual Machine Placement Taxonomy", 2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing, (2015), 159-168, doi: <https://doi.org/10.1109/CCGrid.2015.15>
6. L'opez-Pires, F. Bar'an, B. "Many-Objective Virtual Machine Placement", *Journal of Grid Computing*, Vol. 15, No. 2, (2017), 161-176, doi: <https://doi.org/10.1007/s10723-017-9399-x>.
7. Vaquero, L. M. Rodero-Merino, L. Caceres, J. Lindner, M. "A break in the clouds: Towards a cloud definition", *ACM SIGCOMM Computer Communication Review*, Vol. 39, No. 1, (2009), 50-55, doi: <https://doi.org/10.1145/1496091.1496100>.
8. Gao, Y. Guan, H. Qi, Z. Hou, Y. Liu, L. "A multi-objective ant colony system algorithm for virtual machine placement in cloud computing", *Journal of Computer and System Sciences*, Vol. 79, No. 8, (2013), 1230-1242, doi: <https://doi.org/10.1016/j.jcss.2013.02.004>.
9. Masdari, M. Nabavi, S. S. Ahmadi, V. "An Overview of Virtual Machine Placement Schemes In Cloud Computing", *Journal of Network and Computer Applications*, Vol; 66, (2016), 106-127, doi: <https://doi.org/10.1016/j.jnca.2016.01.011>.
10. Mishra, M. Das, A. Kulkarni, P. Sahoo, A. "Dynamic Resource Management Using Virtual Machine Migrations", *IEEE Communications Magazine*, Vol. 50, No. 9, (2012), 34-40, doi: <https://doi.org/10.1109/MCOM.2012.6295709>.
11. Alguliyev, R. M. Imamverdiyev, Y. N. Abdullayeva, F. J. "PSO-based Load Balancing Method in Cloud Computing", *Automatic Control and Computer Sciences*, Vol. 53, No. 1, (2019), 45-55, doi: <https://doi.org/10.3103/S0146411619010024>.
12. Acharya, J. Mehta, M. Saini, B. "Particle Swarm Optimization Based Load Balancing in Cloud Computing". IEEE. International Conference on Communication and Electronics Systems, (2017), 1-4, doi: <https://doi.org/10.1109/CESYS.2016.7889943>.
13. Xiao, Z. Jiang, J. Zhu, Y. Ming, Z. Zhong, S. Cai, S. "A solution of dynamic VMs placement problem for energy consumption optimization based on evolutionary game theory' *Journal of Systems and Software*, Vol. 101, (2015), 260-272, doi: <https://doi.org/10.1016/j.jss.2014.12.030>.
14. Wang, L. Ai, L. "Task Scheduling Policy Based on Ant Colony Optimization in Cloud Computing Environment. in LISS", Springer, Berlin, Heidelberg, (2013), 953-957, doi: <https://doi.org/10.1007/978-3-642-32054-5-133>.
15. Pacini, E. Mateos, C. Garino, Ca. G. "Balancing Throughput and Response Time in Online Scientific Clouds via Ant Colony Optimization", *Advances in Engineering Software*, Vol. 84, (2015), 31-47, doi: <https://doi.org/10.1016/j.advengsoft.2015.01.005>.
16. Dong, J. k. Wang, H. LI, Y. Cheng, S. "Virtual machine placement optimizing to improve network performance in cloud data centers", *The Journal of China Universities of Posts and Telecommunications*, Vol. 21, No. 3, (2014), 62-70, doi: [https://doi.org/10.1016/S1005-8885\(14\)60302-2](https://doi.org/10.1016/S1005-8885(14)60302-2).
17. Liu, C. Shen, C. Li, S. Wang, S. "A new evolutionary multi-objective algorithm to virtual machine placement in virtualized data center", In 2014 IEEE 5th International Conference on Software Engineering and Service Science, (2014), 272-275, doi: <https://doi.org/10.1109/ICSESS.2014.6933561>.
18. Moges, F. F. Abebe, S. L. "Energy-aware VM placement algorithms for the OpenStack Neat consolidation framework", *Journal of Cloud Computing*, Vol. 8, No. 1, (2019), 2-14, doi: <https://doi.org/10.1186/s13677-019-0126-y>.
19. Tordsson, J. Montero, R. Moreno-Vozmediano, R. Llorente, I. "Cloud brokering mechanisms for optimized placement of virtual machines across multiple providers". *Future Generation Computer Systems*, Vol. 28, (2012), 358-367, doi: <https://doi.org/10.1016/j.future.2011.07.003>.
20. Song, W. Xiao, Z. Member, S. Chen, Q. Luo, H. "Adaptive Resource Provisioning for the Cloud Using Online Bin Packing". *IEEE Transactions on Computers*, Vol. 63, No. 11, (2014), 2647-2660, doi: <https://doi.org/10.1109/TC.2013.148>.
21. Abdi, S. Sharifian. S. A. Sharifian, S. "Task Scheduling Using Modified PSO Algorithm in Cloud Computing Environment. Proc. International Conference on Machine Learning", Electrical and Mechanical Engineering, Dubai, (2014), 37-41, doi: <http://dx.doi.org/10.15242/IIE.E0114078>.
22. Agnihotri, M. Sharma, S. "Execution analysis of load balancing particle swarm optimization algorithm in cloud data center", IEEE. 2016 Fourth International Conference on Parallel, Distributed and Grid Computing, (2016), 668-672, doi: <https://doi.org/10.1109/PDGC.2016.7913206>.
23. Feng, D. Wu, Z. Zuo, D. Zhang, Z. "A multiobjective migration algorithm as a resource consolidation strategy in cloud computing", *PLOS*, Vol. 14, No. 2, (2019), 1-25, doi: <https://www.researchgate.net/deref/https%3A%2F%2Fdoi.org%2F10.1371%2Fjournal.pone.0211729>.

24. Dörterler, S. Dörterler, M. Ozdemir, S. "Multi-Objective Virtual Machine Placement Optimization for Cloud Computing", IEEE, 2017 International Symposium on Networks, Computers and Communications, (2017), 1-6, doi: <https://doi.org/10.1109/ISNCC.2017.8072013>.
25. Fatima, F. Javaid, N. Butt, A.A. Sultana, T. Hussain, W. Bilal, M. Hashmi, R. m Akbar, M. Ilahi, Manzoor. "An Enhanced Multi-Objective Gray Wolf Optimization for Virtual Machine Placement in Cloud Data Centers", *Electronics*, Vol. 8, No. 2, (2019), 218-250, doi: <https://doi.org/10.3390/electronics8020218>.
26. Alresheedi, S. Lu, S. Elaziz, M.A. Ewees, A.A. "Improved multiobjective salp swarm optimization for virtual machine placement in cloud computing", *Human-centric Computing and Information Sciences*, Vol. 9, No. 15, (2019), 1-24, doi: <https://doi.org/10.1186/s13673-019-0174-9>.
27. Alnusairi, S. Shahin, A. Daadaa, Y. "Binary PSO-GSA for Load Balancing Task Scheduling in Cloud Environment", *International Journal of Advanced Computer Science and Applications*, Vol. 9, No. 5, (2018), 255-264, doi: <https://dx.doi.org/10.14569/IJACSA.2018.090535>.
28. Kennedy, J. Eberhart, R. "Particle Swarm Optimization". IEEE International Conference on Neural Networks, Proceedings, Vol. 4, (1995), 1942-1948, doi: <https://doi.org/10.1109/ICNN.1995.488968>.
29. Reynolds, Craig. W. "Flocks, herds and schools: a distributed behavioral model", *Computer Graphics*, Vol. 21, No. 4, (1987), 25-34, doi: <https://doi.org/10.1145/37402.37406>.
30. Yan, J. Zhang, H. Xu, H. Zhang, Z. "Discrete PSO-based workload optimization in virtual machine placement", *Personal and Ubiquitous Computing*, Vol. 22, (2018), 589-596, doi: <https://doi.org/10.1007/s00779-018-1111-z>.
31. Ebadifard, F. Babamir, S. M. "A PSO - based task scheduling algorithm improved using a load-balancing technique for the cloud computing environment", *Concurrency and Computation: Practice and Experience*, (2017), 1-16, <http://dx.doi.org/10.1002/cpe.4368>.

---

### Persian Abstract

---

#### چکیده

امروزه، با رشد استفاده از رایانش ابری، به حداکثر رساندن سود، کاهش هزینه عملیاتی و زمان‌بندی وظایف از مهمترین مسایل این حوزه هستند. در این راستا، یکی از چالش‌های رایانش ابری، ارائه روشی کارآمد برای جایابی ماشین‌های مجازی بر روی ماشین‌های فیزیکی به منظور بهبود مصرف انرژی، توزیع عادلانه بار و زمان‌بندی وظایف است. هدف از این مقاله ارائه روشی برای بهبود زمان‌بندی وظایف با استفاده از الگوریتم بهبود یافته توده ذرات می‌باشد. در این روش پیشنهادی با انتخاب تابع هدف مناسب علاوه بر جایابی پویا ماشین مجازی بر روی ماشین فیزیکی، سبب متعادل کردن بار کاری ماشین‌های مجازی، کاهش زمان تمامی وظایف و همچنین استفاده حداکثری از تمامی منابع و بالا بردن بهره‌وری شده است. نتایج شبیه سازی نشان می‌دهد یک راه حل بهینه برای زمان‌بندی کارها، تفکیک مساوی کارها در ماشینهای مجازی و قرارگیری در ماشین فیزیکی مناسب ارائه شده است و زمان کمتری با بهبود ۲۰ برای فرآیند واگذاری کارها به ماشینهای مجازی حاصل شده است.

---