ISSN (Print) : 0974-6846
Indian Journal of Science and Technology, Vol 9(4), DOI: 10.17485/ijst/2016/v9i4/80561, January 2016 ISSN (Online) : 0974-5645

An Appraisal of Meta-Heuristic Resource Allocation Techniques for IaaS Cloud

Syed Hamid Hussain Madni*, Muhammad Shafie Abd Latiff, Yahaya Coulibaly and Shafi'i Muhammad Abdulhamid

Faculty of Computing, Universiti Teknologi Malaysia, Skudai - 81310, Johor, Malaysia; madni4all@yahoo.com, shafie@utm.my, coulibaly@utm.my, shafii.abdulhamid@futminna.edu.ng

Abstract

Background/Objectives: This appraisal investigates the meta-heuristics resource allocation techniques for maximizing financial gains and minimizing the financial expenses of cloud users for IaaS in cloud computing environment. **Methods/Statistical Analysis**: Overall, a total of ninety-one studies from 1954 to 2015 have been reviewed in this paper. However, twenty-three studies are selected that focused on the meta-heuristic algorithms for their research. The selected papers are categorized into eight groups according to the optimization algorithms used. **Findings:** From the analytical study, we pointed out the various issues addressed (optimal and dynamically resource allocation, energy and QoS aware resource allocation, VM allocation and placement) through resource allocation meta-heuristics algorithms.Whereas, the improvement shows better performance concerns minimizing the execution and response time, energy consumption and cost while enhancing the efficiency and QoS in this environment. The comparison parameters (makespan 35%,execution time 13%, response time 26%, cost 22%, utilization21% and other 13% including energy, throughput etc) and also the experimental tools (CloudSim 43%, GridSim 5%, Simjava 9%, Matlab 9% and others 13%) used for evaluation of the various techniques for resource allocation in IaaS cloud computing. **Applications/Improvements:** The comprehensive review and systematic comparison of meta-heuristic resource allocation algorithms described in this appraisal will help researchers to analyze different techniques for future research directions.

Keywords: IaaS Cloud, Meta-Heuristic Algorithms, Resource Allocation, Resource Utilization, Resource Management

1. Introduction

In cloud computing, resource allocation plays a major role as it is a resource constrained environment. It is the procedure of allocating resources to the cloud user according to the requirements and usage of cloud resources which contain mostly virtual resources shown in Figure 1. The cloud user request for virtualized resources is defined through a set of parameters specifying the CPU, processors, I/O, networks, servers, storage, applications, services, etc[1-3].

Effective and efficient utilization of resources is a necessary requirement in a cloud computing environment. When the workloads and availability of resources are fluctuating dynamically, meeting the quality of service constraints together with preserving a satisfactory level of utilization and system performance are the crucial problems to tackle[2,4]. This dynamic behavior in turn arouses challenges between cloud users and cloud providers to maximize their mandatory tasks or resources. Therefore, resource allocation is considered as the central theme of the cloud computing[5]. Meta-heuristic algorithms have made a strong place in the research field in the past few years due the high efficiency and effectiveness to solve some of the most problematic issues in cloud computing.

Although resource allocation in cloud computing environment has already raised the interest of the global research community, recent studies and reviews draw attention towards the resource allocation achievement. The crucial aim of the resource allocation technique is to find an allocation that is optimal and feasible for a given service[6]. An efficient resource allocation strategies that

*Author for correspondence

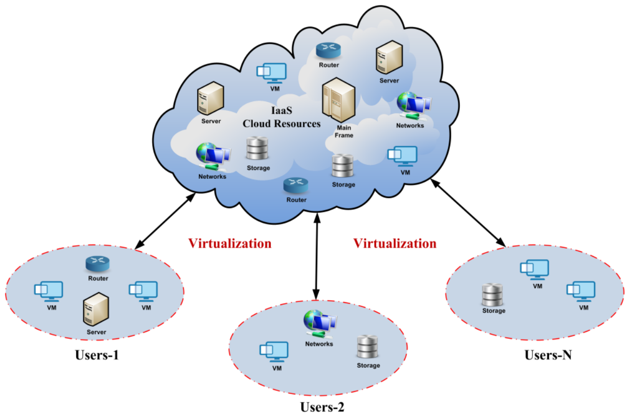

Figure 1. Resource Allocation in IaaS Cloud

utilize resources effectively in the resource constrained environment of cloud computing are classified.

The challenges for resource allocation in distributed clouds and concentrating the main problematic issues for such a cloud paradigm are designated by Endo et al[7]. These are resource modeling, resource treatment and offering, resource monitoring and discovery, resource availability and resource selection. Although distinct challenges are mandatory in the new research, distributed clouds are encouraging and may develop to be seen in various contexts.

Resource allocation approaches comprise of dynamic autonomous resource management in order to be responsible for elasticity, scalability, and deduction of allocation size and cost. Approaches are used in existing research are an optimization, criteria decision analysis, multi agent system, multiple simulation, graph methods, prediction, service oriented structural design and theoretical formulation[8]. Forthcoming research comprises enhancement of the dynamic resource allocation, as per the standardization and application running in cloud.

The term resource provisioning includes finding the resource from the cloud providers, scheduling the Virtual Machine (VM) for utilizing the resources, utilizing the resource efficiently based on the pricing schema, managing the workload for each resource, allocating the resources and its VM for third parties, etc. The evolution of cloud computing with various resource provisioning techniques have been discussed here using various resource provisioning techniques have been discussed here using various parameters such as price, utility, customer satisfaction, demand, reliability, efficiency, scalability, power consumed by the resources and last but foremost important factor of cloud Quality of Service (QoS)[9].

The objectives of this paper are to review and analyze the different issues and meta-heuristic techniques used in the resource allocation schemes in the IaaS cloud computing environment. The remaining sections of the papers are organized as follows: Section 2 focuses on the description and importance of Meta-heuristics resource allocation algorithms. The different meta-heuristic algorithm used to solve resource allocation problem are presented and analyzed in Section 3 and 4. In Section 5, we present the conclusion and recommendations.

2. Meta-Heuristic Algorithms

Traditional resource allocation techniques are not adequate for cloud computing as they are based on virtualization technology with distributed nature. Cloud computing brings together new challenges for manageable and flexible resource allocation due to heterogeneity in hardware capabilities, workload estimation, and characteristics in order to meet Service Level Objectives of the cloud consumer's applications. The ultimate goal of resource allocation in cloud computing is to maximize the financial gain for cloud providers and to minimize the financial expenses for cloud users[4]. However, traditional algorithms are easier to understand, and simpler to implement than other optimization techniques identical numerical programming and analytical approaches. The results are not guaranteed to be optimal produced by traditional algorithms[10].

"Meta" and "heuristic" are of Geek origin, where the former means "higher level" or "beyond" and the later one means "to know" or "to investigate". Now to find a good optimal solution with low computational cost Heuristic methods are used without guaranteeing optimality or feasibility. To enhance the efficiency of heuristic procedures, a set of meta-heuristics intelligent strategies are used. To improve the performance of meta-heuristics numerous techniques are considered in the algorithms. While, in solving all problems none of the meta-heuristic algorithms is competent to performance as compared to others. Moreover, the current algorithms have some performance laggings like trapping into low optima, slow rate of convergence, extensive computational time, having complex operators and design for binary or real search space only. Therefore, presenting new meta-heuristic algorithms to cater the weaknesses is a serious problematic issue[11].

In simple words, Meta-heuristics is a term given to a general class of algorithm used to find solutions to optimization problems when exact techniques prove inadequate. With meta-heuristic, the objective function can be of any type and precede into consideration various objectives. On another side, all meta-heuristic algorithms have need of a suspicious modification of optimization parameters that are crucial for discovering a better solution lacking extreme computational time[10].

Many different meta-heuristics are in presence and novel variations are frequently suggested for the resource allocation in various fields. There are many different meta-heuristic algorithms are prominent and remarkable in the area of cloud computing for the resource management like Ant Colony Optimization (ACO), Cuckoo Search (CS), Differential Evolution (DE), Firefly Algorithm (FA), Harmony Search (HS), Immune Algorithm (IA), League Championship Algorithm (LCA), Memetic Algorithm (MA) and many many more. Some of the most significant meta-heuristic algorithms are listed in Table 1.

Table 1. Some of the well-known Meta-Heuristic Algorithms from (1954 to 2015)

Sr. No	Algorithm	Year	Introduced By
1	Evolution Process (EP)	1954	Barricelli[12]
2	Random Search (RS)	1963	Rastrigin[13]
3	Random Optimization (RO)	1965	Matyas[14]
4	Evolutionary Programming (EP)	1966	Fogel et al.[15]
5	Graph Partitioning Method (GPM)	1970	Kernighan and Lin[16]
6	Genetic Algorithm (GA)	1975	Holland[17]
7	Scatter Search (SS)	1977	Glover[18]
8	Metaplan	1978	Mercer and Sampson[19]
9	Simulated Annealing (SA)	1983	Kirkpatrick et al.[20]
10	Tabu Search (TS)	1986	Glover[21]
11	Memetic Algorithm (MA)	1989	Moscato[22]
12	Ant Colony Optimization (ACO)	1992	Dorigo[23]
13	Immune algorithm (IA)	1993	Mori et al.[24]
14	Multi-Objective GA (MOGA)	1993	Fonseca and Fleming[25]
15	Reactive Search Optimization (RSO)	1994	Battiti and Brunato[26]
16	Particle Swarm Optimization (PSO)	1995	Kennedy and Eberhart[27]
17	Differential Evolution (DE)	1997	Storn and Price[28]
18	Harmony Search (HS)	2001	Geem et al.[29]
19	Artificial Bee Colony Algorithm (ABC)	2005	Karaboga[30]
20	Glowworm Swarm Optimization (GSO)	2005	Krishnanand and Ghose [31]
21	Honey bee Mating Optimization (HbMO)	2006	Haddad et al.[32]
22	Imperialist Competitive Algorithm (ICA)	2007	Atashpaz-Gargari[33]
23	Intelligent Water Drops (IWD)	2007	Shah-Hosseini[34]
24	Firefly Algorithm (FA)	2008	Yang[35]
25	Cuckoo Search (CS)	2009	Yang and De [36]
26	Gravitational Search Algorithm (GSA)	2009	Rashedi et al.[37]
27	League Championship Algorithm (LCA)	2009	Kashan[38]
28	Bat Algorithm (BA).	2010	Yang[39]
29	Galaxy based Search Algorithm (GbSA).	2011	Shah-Hosseini[40]
30	Spiral Optimization (SO)	2011	Tamura and Yasuda [41]
31	Teaching Learning Based Optimization (TLBO)	2011	Rao et al. [42]
32	Krill Herd (KH)	2012	Gandomi and Alavi [43]
33	Coral Reefs Optimization (CRO)	2013	Salcedo-Sanz et al.[44]
34	Swallow Swarm Optimization (SSO)	2013	Neshat et al.[45]
35	Interior Search Algorithm (ISA)	2014	Gandom[46]
36	Gradient Evolution Algorithm (GEA)	2015	Kuo and Zulvia[47]
37	Lion Optimization Algorithm (LOA)	2015	Yazdani and Jolai[48]
38	Optics Inspired Optimization (OIO)	2015	Kashan[49]
39	Water Wave Optimization (WWO)	2015	Zheng[50]

(*Continued*)

Resource allocation for IaaS in cloud computing is considered very comprehensively in various literature. The problem of determining an optimal consignment of demands to resources allocator is NP-hard. Some heuristic algorithms are anticipated by research scholars for optimal allocation of resources in IaaS Cloud. Although the few meta-heuristic algorithms are functioned for the resource allocation in cloud computing are Artificial Bee Colony (ABC), Ant Colony Optimization (ACO), Genetic Algorithm (GA), Immune Algorithm (IA), League Championship Algorithm (LCA), Particle Swarm Optimization (PSO) and Simulated Annealing (SA) are discussed in next section.

3. Meta-Heuristic Algorithms for Resource Allocation in IaaS Cloud

In Cloud Computing, it is desirable to search an optimal solution for resource allocation within the short period of time. Meta-heuristic based techniques have been proved to provide immediate optimal results within the reasonable time for this kind of problem. Here we provide a survey of cloud computing based on the seven popular meta-heuristic techniques, and the analysis of these meta-heuristic techniques is presented in Section 4.

3.1 Artificial Bee Colony

Artificial Bee Colony (ABC) algorithm is one of the peaks newly familiarized swarm based algorithms. The intellectual honeybee swarms foraging attitude is simulated in ABC. And the fitness of the associated solution is corresponded by the nectar amount of the origin of food. A potential solution to the optimization problem and the nectar amount of the origin of food match up to the quality (fitness) of the associated outcome are represented by the ABC algorithm[51]. Pseudo-code of ABC algorithm for the optimization is shown in Figure 2. If the nectar amount of the latest origin of food is greater than the last one in their memory, they remember the latest location and fail to remember the last one. Consequently, ABC balances exploitation and exploration procedure with local and universal search methods in various phases and acquires the best solution. This algorithm has local and universal search capability, implement with numerous optimization problems and accessible for hybridization combination with other algorithms[52].

A model for energy aware resource utilization procedure is suggested to accomplish efficiently the cloud resources and improve their utilization. An artificial bee colony depends upon energy aware resource utilization technique equivalent to the model is considered for the allocation of tasks to the resources in a cloud computing. Furthermore, it supports minimizing the usage of energy for clouds by using server association through virtualization lacking demeaning the tactics of user's demand[54].

3.2 Ant Colony Optimization

To resolve the challenging combinatorial optimization problems, ant colony optimization is familiarized in the early 1990's[23]. The foraging attitude of actual ants is the encouraging the origin of ACO. Ants explore their adjacent area near to their nest in a random routine even though they are searching for food. Whenever an ant searches an origin of food, it first examines the feature and quantity of food, and then it brings some portion of it on its back to the nest. While returning back, then it spreads a chemical pheromone trail on the surface. This pheromone directs other ants to find the origin of the food. The quality of the pheromone depends upon the quality of food. Pseudo-code of ACO algorithm is shown in Figure 3 for the optimization of resource allocation problem in cloud. This algorithm has strong point in cooperation of local and universal searches and implemented with numerous optimization problems.

```
Pseudo-code for Artificial Bee Colony
Initialize the population of solutions x_{i,j}, i = 1...SN, j = 1...D;
Evaluate the population;
for i=1 to N
        Produce new solutions v_{i,j} for the employed bees by using
        (2) and evaluate them;
        Apply the greedy selection process;
        Calculate the probability values P_{i,j} for the solutions x_{i,j} by
        (1);
        Produce the new solutions v_{i,j} for the onlookers from the
        solutions x_{i,j} selected depending on P_{i,j} and evaluate them
        Apply the greedy selection process;
        Determine the abandoned solution for the scout, if exists,
        and replace it with a new randomly produced solution x_{i,j} by
        (3);
        Memorize the best solution achieved so far;
        i=i+1
end
```

Figure 2. Pseudo-code for the ABC[53]

```
Pseudo-code for Ant Colony Optimization
Input: An instance P of a CO problem model P = (S, f, Ω).
Output: P_best
InitializePheromoneValues (T);
P_best ← NULL;
while termination conditions not met do
        S_iter ← θ;
        for j = 1 to N
                s ← ConstructSolution (T);
                if s is a valid solution then
                s ← LocalSearch(s); {optional}
                if (f (s) < f(P_best)) or (P_best = NULL) then P_best ← S;
                S_iter ← S_iter ∪ {s};
                end
        end
        ApplyPheromoneUpdate (T ,S_iter , P_best);
end
return P_best ;
```

Figure 3. Pseudo-code for the ACO[23]

An ant colony framework for adaptive resource allocation in cloud computing environments for hosting the applications with specified QoS necessities as a response time and throughput that minimizes the usage of energy of data center resources by making an allowance for the dynamic workloads of servers by means of various ant agents[55].

The problematic issue of VM placement is expressed as a multi-objective combinatorial optimization problem, whose core goal is to optimize concurrently total resource wastage and energy consumption. An improved form of the Ant Colony System (ACS) algorithm is offered and structured to deal effectively with the possible enormous solution place for the data centers[56].

Firstly, new ACO algorithm foreseen the ability of the possibly existing resource nodes then, it examined some aspects, for instance, network qualities and response times to accomplish a set of optimal calculate nodes. In the last, the tasks have dispersed to the appropriate nodes[57]. The algorithm is forecasted the competency of the prospectively available resource node, during the allocating and analyzing with the usage of bandwidth, the quality of networks and the response time[58].

In the IaaS where Virtual Machines (VM) are launched in applicable hosts accessible in a Cloud. At that time, proper resource scheduling is significant, and it is essential to develop efficient scheduling strategies to the appropriate allocation of VMs to physical machines. A Cloud scheduler depends on Ant Colony Optimization (ACO). The main performance metrics to analyze the number of serviced users by the Cloud and the total number of generated VMs in online (non-batch) scheduling situations[59]. However for sustaining the servers in the data center a large amount of energy is consumed. Additional physical servers mean extra energy consumption and supplementary cost. For that reason, the VM Placement (VMP) problematic issue is substantial in cloud computing. A technique based on Ant Colony Optimization (ACO) to resolve the VMP problem, known as ACO-VMP, so as to effectually work for the physical resources and to decrease the amount of consecutively physical servers[60].

3.3 Genetic Algorithm

John H. Holland in early 1970's introduced Genetic Algorithm (GA). GA is a probabilistic optimization algorithm that imitates the progression of natural evolution. The biological evolution process in chromosomes became the idea of GA. It is based on the idea of the fittest survival where new better solutions are obtained by recombination of with each other. Figure 4 displays the pseudo-code for GA algorithm for the optimal solution in cloud. In the algorithm, a population of strings known as chromosomes encrypts applicant solutions for optimization problems[17].

Allocating enough and suitable resources to VMs is a challenging issue for their source manager. Virtualization technology which is the basis of cloud computing facilitates the process of sharing PM's resources between VMs. Hence, the resource manager aggregates all the accessible resources and enhances the resource utilization. An advantage of Genetic Algorithm (GA) to resolve the problem of resource allocation and recommends a new model to enhance the result of the decision making process[61].

An objective is discovery trade-off solutions between tasks completion time and system energy consumption, for cloud computing data centers that perform joint allocation of computational and network resources[62]. The algorithm is designed using genetic algorithms that allow both to explore solutions space and to search for the optimal solution in an efficient manner. It is both scalable and energy efficient and is based on a model developed to capture specifics of the data center network topology and device power consumption.

In the study, a novel architecture for IaaS cloud computing environment in which VM allocation is done through genetically weight optimized neural network[63]. In this situation, the host load of each machine is occupied as its resource information. The neural network forecasts the host workload of each machine in near future based on the recent past host workload. It helps the VM allocator to choose the right machine for the allocation of resources.

Pseudo-code for Genetic Algorithm

```
Input: Population_size, Problem_size, P_crossover, P_mutation
Output: S_best
Population ← InitializePopulation (Population_size, Problem_size);
EvaluatePopulation (Population);
S_best ← GetBestSolution (Population);
while : StopCondition ( ) do
    Parents ← SelectParents (Population, Population_size);
    Children ← θ;
        foreach Parent_1, Parent_2 ϵ Parents do
        Child_1, Child_2 ← Crossover (Parent_1, Parent_2,
        P_crossover);
        Children ← Mutate (Child_1, P_mutation);
        Children ← Mutate (Child_2, P_mutation);
    end
    EvaluatePopulation (Children);
    S_best ← GetBestSolution (Children);
    Population ← Replace (Population, Children);
end
return S_best ;
```

Figure 4. Pseudo-code for the GA[53]

Fast optimal allocation of computing resources is the most important technical issues in cloud computing. the computation time which is an important influence on cloud computing for fast response to cloud user request[64]. GA rapidly provides a good quality schedule that can optimally allocate computing resources and satisfy user's demands.

3.4 Immune Algorithm

Artificial immune systems contain Immune algorithms with methods of computation that is stimulated by the mechanisms and process of the biological immune system. The immune system is an organ system that protects the host from the threats caused by pathogens and toxic materials[65]. Pseudo-code of IA algorithm for the optimization is shown in Figure 5.

The experimental raised area of resource allocation algorithm, performance analysis, and energy optimization, achieve innovative accomplishments in the field of systematic research, for the resource allocation technique based on immune algorithm and energy optimization in cloud computing to deliver inventive concepts and scientific basis[66]. To optimize resource allocation expend an Improved Clonal Selection Algorithm (ICSA) depends on energy consumption models and make span optimization in cloud computing[67]. The ICSA has an influential overall exploration skill in a given realistic solution range and usages the smaller number of execution time. Therefore, the recommended ICSA is well improved and well-adjusted in exploitation and exploration.

Definite micro-economic methods are adapted to resolve the problematic issue of resources management, particularly resource allocation in cloud computing environment. Moreover, the concept of batch matching into the reverse auction, a reverse batch matching auction method is offered[68]. Three assessment standards, comprising the QoS, satisfaction of users, and market efficiency, are operated as the optimization intentions, and the immune evolutionary algorithm is functional to discover the results for optimal resource allocation.

Pseudo-code for Immune Algorithm

```
Input: (l, d, dup, T_B, c).
Output: P_best
Nc ← d x dup;
P^(t) ← Initail_Pop( );
evaluate(P^(0));
while : ( Termination_Condition( ) ) do
    P^(clo) ← Cloning (P^(t), Nc);
    P^(hyp) ← Hypermutation (P^(clo), c, l);
    evaluate(P^(hyp));
    P^(macro) ← Hypermacromution ( P^(clo) );
    evaluate(P^(macro));
    (P_a^(t), P_a^(hyp),P_a^(macro)) ← Aging(P_a^(t), P_a^(hyp),P_a^(macro),T_B);
    P^(t+1) ← (μ+λ)-Selection (P_a^(t), P_a^(hyp),P_a^(macro));
    t←t+1;
end
```

Figure 5. Pseudo-code for the IA[53]

3.5 League Championship Algorithm

In the year 2009, Ali Husseinzadeh Kashan is familiarized the League Championship Algorithm (LCA) optimization algorithm that is inspired by sport league championship. Due to the prospective efficiency in resolving numerous real-world applications and optimization problems, since it has attracted much attention among the researchers[69]. The efficiency of the LCA measure the check functions from a renowned yard-stick, usually adapted to latest constraint handling algorithm strategy. The LCA has also exposed excessive strengths in workout the non deterministic polynomial time for NP-hard problems and its pseudo-code express in Figure 6. The outcomes obtained from the LCA are very realistic with concerns to other famous methods that previously used for reserved optimization[70]. Since the winner determination is an NP-hard problem, a league championship algorithm is familiarized to succeed optimum allocation with the optimization intents existence market surplus and entire reputation[71].

Pseudo-code for League Championship Algorithm

```
Initialize the league size (L) and the number of seasons (S); t = 1;
Generate a league schedule;
Initialize team formations (generate a population of L solutions)
and determine the playing strengths (function or fitness value)
along with them. Let the initialization be also the teams' current
best formation;
while (t ≤ S. (L - 1)) or (no change in the last 100 iterations)
    Based on the league schedule at week t, determine the
    winner/loser among every pair of teams using a playing
    strength based criterion;
    t = t + 1;
    for i = 1 to L
        Devise a new formation for team i for the
        forthcoming match, while take into account the
        team's current best formation and previous week
        events. Evaluate the playing strength of the resulting
        arrangement;
        If the new formation is the fittest one (that is, the new
        solution is the best solution achieved so far for the ith
        member of the population), hereafter consider the
        new formation as the team's current best formation;
    end
    if mod (t , L - 1) = 0
        Generate a league schedule;
    end
end
```

Figure 6. Pseudo-code for the LCA[72]

3.6 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a swarm based meta-heuristic algorithm which simulates the nature such as a flock of insects, bird's gesture or schooling of fish in order to discover the greatest solution. In the algorithm, PSO consists of a population known as swarm and every participant (the insects, birds or fishes) are considered as particles and are prepared by random velocities and situations[73]. Figure 7 provides a pseudo-code listing of the PSO algorithm for reducing the function of cost. It is universal optimization algorithm, which the finest result can be appeared as a point or surface in a multi-dimensional search. In the algorithm, their fitness values examine the particles. They move in the direction of those particles, which have higher fitness values and as a final point achieve the finest result. The universal search of the algorithm is efficient, with the dependency on the initial solution is minor. The algorithm can easily be implement and has less parameter for modification[74].

To minimize the usage of energy in cloud data center, energy efficient virtual machine allocation algorithm is suggested the Particle Swarm Optimization (PSO) technique and the energy efficient multi-resource allocation model. In this algorithm, the fitness function of PSO is diverse as the entire distance to conclude the optimal point between energy consumption and resource utilization. This algorithm can escape dropping into local optima, which is common in traditional algorithms[75].

Pseudo-code for Particle Swarm Optimization

```
Input: ProblemSize, Population_size
Output: P_g-best
Population ← θ;
P_g-best ← θ;
for i = 1 to Population_size do
    P_velocity ← RandomVelocity( );
    P_position ← RandomPosition(Population_size);
    P_g-best ← P_position;
    if Cost (P_p-best) ≤ Cost (P_g-best) then
        P_g-best ← P_p-best;
    end
end
while: StopCondition( ) do
    foreach P ϵ Population do
        P_velocity ← UpdateVelocity(P_velocity, P_g-best, P_p-best);
        P_position ← UpdatePosition(P_position, P_velocity);
        if Cost (P_position) ≤ Cost (P_p-best) then
            P_p-best ← P_position;
            if Cost (P_p-best) ≤ Cost (P_g-best) then
                P_g-best ← P_p-best;
            end
        end
    end
end
return P_g-best;
```

Figure 7. Pseudo-code for the PSO[53]

Moreover, a strategy is presented to focus on efficient VM allocation to physical servers in order to reduce the total resource wastage and number of servers used[76].

Although the enhancement of dynamically resource allocation and improvement, a supplementary advantage in the data center to VMs migration dynamically by using PSO. It set of scales surplus hosts to assurance the response time and deadline of the work (QoS)[77]. Also, associate under the workload of host and energy to achieve more energy efficiency and power saving.

Deadline Constrained Task Scheduling (DCTS) problem is considered, when the resources are not enough and appropriate for the user demand and each task is concomitant with a strict deadline. This resource allocation problem is solved with the help of Cuckoo Driven Particle Swarm Optimization by assurance of QoS constraint and cloud provider revenue[78].

An innovative technique known as Position Balanced Parallel Particle Swarm Optimization (PBPPSO) algorithm is presented for allocation of resources in IaaS cloud[79]. The core intent of PBPPSO is to discover the resource optimizations for the group of jobs with less make span and lowest cost. The set of directions is produced from the resource optimizations for the preparation method. With the simulation, the resources are distributed to the new cloud users by learning the instructions of the preparation method.

3.7 Simulated Annealing

Simulated Annealing is an initial Meta-heuristic algorithm inventing from an analogy of how an optimal atom arrangement is originated in the statistical mechanics. It is a general method to find a solution for some NP hard optimization problems in a heuristic process, recommended by Kirkpatrick et al[20]. The SA's effectiveness comes out from its extension of two primary heuristic techniques, iterative improvement scheme and divides and conquers approach. It processes the temperature as an explicit strategy to direct the search. The solution space is frequently discovered by taking random tries. The Simulated Annealing method randomly produces a large number of potential results, maintaining together good and bad solutions. It split up the large and small alterations by permitting large changes in the objective function[80], that can be understood by the pseudo-code for SA algorithm in Figure 8.

A Simulated Annealing Load Balancing (SALB) algorithm is suggested to explore the optimal resource

Pseudo-code for Simulated Annealing

```
Input: ProblemSize, iterations_max, temp_max
Output: S_best
S_current ← CreateInitialSolution (ProblemSize);
S_best ← S_current ;
for i = 1 to iterations_max do
    S_i ← CreateNeighborSolution (S_current);
    temp_curr ← CalculateTemperature (i, temp_max);
    if Cost (S_i) ≤ Cost (S_current) then
        S_current ← S_i;
            if Cost (S_i) ≤ Cost (S_best) then
                S_best ← S_i;
            end
    else if Exp ((Cost (S_Current)−Cost (S_i)) / temp_Curr) > Rand ( ) then
        S_current ← S_i;
    end
end
return S_best;
```

Figure 8. Pseudo-code for the SA[53]

allocation in cloud computing systems. It reduces the standard deviation of the cloud providers load and balance the work load whole system with efficient resource allocation. A VM allocation strategy on load balancing of VM resources applying a genetic algorithm, which deliberates the VM to be organized one by one. In the simulation, the Round Robin scheduling strategy, basic Simulated Annealing algorithm, and SALB algorithm are used and associated. The simulation results express that the suggested SALB algorithm is achieved a better system balanced load[81].

The Virtual Network Embedding (VNE) problem is considered one of the most problematic concerns in the network virtualization. It focuses on the competent distribution of physical resources to virtual networks in the data center of cloud. The study reflects the VNE problem in the situation of high-demand, in which the principal anxiety of Cloud Infrastructure Provider is how to embed more virtual network demands in order to growth the Cloud Infrastructure Provider's durable profits. The capability of simulated annealing method is used in controlling the VNE problem. The simulation outcomes point out that suggested technique accomplishes better than both techniques (GA and PSO) in managing the cost aware VNE problem[82].

A lot of struggles are prepared to generate data centers more energy efficient. To decrease the entire power consumption of the servers is one of them in a data center. It is executed by virtual machine placement through virtual machine consolidation. For the enhancement of energy efficiency in data centers, a virtual machine placement algorithm which depends on Simulated Annealing technique is proposed. This is the primary effort of via SA based algorithms is used to solve optimizing energy efficiency as a VM placement problem. First Fit Decreasing and Random Searching are used for comparison in estimating the performance of the Simulated Annealing algorithm[83].

4. Analysis of Meta-Heuristic Algorithms for Resource Allocation in IaaS Cloud

This section provides a comparative analysis of various existing meta-heuristic resource allocation algorithms in IaaS Cloud Computing as shown in Table 2.

Cloud provider and cloud user have their own incentive when they become the part of the cloud. Cloud provider are concerned with the utilization of cloud resources with gain to have high or maximum profit while the cloud users are interested in the high performance with low cost or minimum budget[5]. Efficient resource allocation can enhance the time, cost and energy consumption. It can also reduce the under-utilization of resources, request loss, balance load, and leasing cost[84]. Also, optimal placement and allocation of VMs play a significant role in the enhancement of efficiency of energy and resource utilization[85]. These rational are expressed as optimization criteria while allocation of the resource.

Makespan: Makespan specifies the maximum completion time or time when the resources are allocated to the users[86]. High performance and less energy consumption are achieved by reducing the makespan for resource allocation.

$$Makespan = max_{task^i}(Fnh_{Time}) \tag{1}$$

Execution Time: Execution time is defined as the time difference between the task completion time and the starting time within the environment[87].

$$ExeTime = Fnh_{Time}(Str_{Time}) \tag{2}$$

Response Time: Response indicates the time of search process, which contains the time to execute the job in cloud environment[87].

Cost: It indicates the total amount of the cloud user demand to pay to the cloud provider for the resource utilization.

$$Cost = \sum_{resource^i} (Ci * Ti) \tag{3}$$

Table 2. Meta-Heuristic Algorithms for Resource Allocation in IaaS Cloud

Algorithm	Problems Addressed	Improvement/ Achievement	Weakness/Limitations	Parameters	Tools
Artificial Bee Colony[54]	Energy aware Resource Utilization	Minimize Execution time and Energy Efficiency	Workload of Nodes does not be consider	Execution Time and Energy	CloudSim
Ant Colony Framework[55]	QoS aware Resource Allocation	Reduce Energy Consumption	Not Implemented	Throughput and Response Time	CloudSim
Ant Colony System[56]	VM Placement	Better Performance	Fix the power and no of VM	Time and Energy	Not Mention
Ant Colony Optimization[57]	Dynamically Resource Allocation	Enhance Performance and Reduce Response Time	Based on Grid Environment	Time and Cost	GridSim
Ant Colony Optimization[58]	Efficient Resource Allocation	Better Performance	Not Comprihensive Enough	Response Time	CloudSim
Ant Colony Optimization[59]	VM Allocation	Better Performance	Worst performance in terms of Makespan and flow time	Throughput and Response Time	CloudSim
Ant Colony Optimization[60]	VM Placement	Effective Utilization	Compare with traditional algorithm	Utilization	Not Mention
Cuckoo Driven Particle Swarm Optimization[78]	Optimal Resource Allocation	Better Performance	Focus only on profit	Cost and Exection Time	Not Mention
Genetic Algorithm[61]	VM Allocation	Improve Utilization and Reduce Cost	Compare with traditional algorithms	Response Time	Matlab
Genetic Algorithm[62]	Joint allocation of computational and network resources	Minimize makespan and Energy consumption	Do not compare with other algorithms	Task, Time, and Energy	JMetal
Genetic Algorithm[63]	VM Allocation	Enhance Efficiency	Depend on the prediction	Time and Workload	Matlab
Genetic Algorithm[64]	Optimal allocation of computing resources	High Performance	Ignore the makespan and utilization	Computation Time	C++ using GAlib
Immune Algorithm[66]	Resource Allocation with Energy Optimization	Not Clear	Use existing another algorithm instead of IA in experiments	Response Time	Not Mention
Immune Evolution Algorithm[68]	Auction based Resource Allocation	Market efficiency as the optimization objectives	Consider only economically	Not Mentioned	Simjava
Improved Clonal Selection Algorithm[67]	Energy awareResource Allocation	Improve Execution time and Energy Efficiency	Consider only for Energy consumption	Energy and Time	CloudSim
League Championship Algorithm[71]	Winner Determination problem	Improve performance	Consider only economically	Time and Cost	SimJava
Particle Swarm Optimization[75]	VM Allocation	Improve Energy efficiency	Consider only CPU and Disk as resources	Utilization	CloudSim

(*Continued*)

Particle Swarm Optimization[77]	VM Placement	Improve Energy efficiency	Compare with traditional algorithm	Energy	CloudSim
Particle Swarm Optimization[76]	VM Placement	Minimize the resource wastage and reduce energy consumption	Compare with traditional algorithm	Utilization	Eclipse platform version Kepler service Release 2.
Position Balanced Parallel Particle Swarm Optimization[79]	Efficient Resource Allocation	Minimize makespan and total cost	The resources are allocated to learning the rules for new user request	Response time, Cost	CloudSim
Simulated Annealing[81]	Optimal Resource Allocation	Better workload balancing	This approach may fall into a local optimal solution	Workload	CloudSim
Simulated Annealing[82]	Virtual Network Embedding	Efficient Resource Allocation	Cloud Providers perspective only	Cost	CloudSim
Simulated Annealing[83]	VM Placement	Improve Energy efficiency	Divide and Conquer strategy is used for splitting the huge problem into minor problems and at the end combine them into original shape.	Time, Cost n Energy	Not Mention

Utilization: The most important criteria is resource utilization to keep the resource busy as much as possible[88.] It gains significance as a cloud provider want to earn a maximum profit by leasing limited no of resources.

$$Utilization = \frac{\sum_{resource^i} (ExeTime)}{makespan} \tag{4}$$

After the comprehensive review of the meta-heuristic algorithms proposed for the resource allocation for IaaS in cloud computing. It can be observed that the prospects of the meta-heuristic techniques perform better (see Table 2) in this area. The meta-heuristic algorithms work both in favor of cloud user and cloud provider. In the cloud user perspective, the make span, execution time, response time and cost are very important, which are used as significant parameters for resource allocation in cloud computing. Another importance of a cloud provider is where the utilization, energy consumption, and workload play an important part.

Most of the meta-heuristic techniques are used to provide the optimal solution for the more critical problems like dynamically resource allocation (where the user demand is fluctuating), Virtual Machine Allocation and Placement, Computational and Network Resource Allocation, Optimal Resource Allocation, Energy aware Resource Allocation (mostly energy consumption for data centers). And these techniques are implemented in different simulation tools and the environment like Cloudsim[89,90], GridSim[91], Simjava, Matlab and real cloud environment and give the near-optimal solution for the resource allocation for IaaS in cloud computing.

5. Conclusion and Recommendations

In this paper, we have reviewed most significant meta-heuristic algorithms for resource allocation in IaaS Cloud. Numerous techniques to enhance the performance of meta-heuristics have been considered in these algorithms. However, none of the meta-heuristic algorithms is capable to achieve clearly superior performance than other algorithms in resolving resource allocation problems in cloud computing.

6. Acknowledgement

The authors are thankful to Universiti Teknologi Malaysia (UTM) and Ministry of Education, Malaysia for supporting this research through FRGS and RUG grants Vote No. R.J130000.7828.4F324 and Vote No.Q.J130000.2528.05H87 respectively.

7. References

1. Foster I, Zhao Y, Raicu I, Lu S. Cloud computing and grid computing 360-degree compared. Grid Computing Environments Workshop (GCE'08); Austin, TX. 2008. p. 1–10.
2. Mohan NRR, Raj EB. Resource allocation techniques in cloud computing-research challenges for applications. 4th International Conference on Computational Intelligence and Communication Networks (CICN); Mathura. 2012. p. 556–60.
3. Gong Y, Ying Z, Lin M. A survey of cloud computing. Proceedings of the 2nd International Conference on Green Communications and Networks (GCN 2012); 2012. p. 79–84.
4. Parikh SM. A survey on cloud computing resource allocation techniques. Nirma University International Conference on Engineering (NUiCONE); Ahmedabad. 2013. p. 1–5.
5. Manvi SS, Shyam GK. Resource management for Infrastructure as a Service (IaaS) in cloud computing: A survey. Journal of Network and Computer Applications. 2014; 41(1): 424–40.
6. Anuradha VP, Sumathi D. A survey on resource allocation strategies in cloud computing. International Conference on Information Communication and Embedded Systems (ICICES); Chennai. 2014. p. 1–7.
7. Endo PT, de Almeida PAV, Pereira NN, Goncalves GE, Sadok D, Kelner J. Resource allocation for distributed cloud: Concepts and research challenges. IEEE Network. 2011; 25(4):42–6.
8. Wang LY, Liu AM. The study on cloud computing resource allocation method. Applied Mechanics and Materials. 2012; 198-199(1):1506–13.
9. Jayanthi S. Literature review: Dynamic resource allocation mechanism in cloud computing environment. International Conference on Electronics, Communication and Computational Engineering (ICECCE); Hosur. 2014. p. 279–81.
10. Gandomi AH, Yang XS, Talatahari S, Alavi AH. Metaheuristic Applications in Structures and Infrastructures. Newnes, USA: Elesvier; 2013.
11. Beheshti Z, Shamsuddin SMH. A review of population-based meta-heuristic algorithms. International Journal of Advance Soft Computer Application. 2013; 5(1):1–35.
12. Barricelli N. Numerical examples of evolution processes. Methodos. 1954; 1954(1): 45–68.
13. Karnopp DC. Random search techniques for optimization problems. Automatica. 1963; 1(2):111–21.
14. Matyas J. Random optimization. Automation and Remote Control. 1965; 26(2):246–53.
15. Fogel DB. Applying evolutionary programming to selected traveling salesman problems. Cybernetics and systems. 1993, 24(1), pp. 27–36.
16. Kernighan BW, Lin S. An efficient heuristic procedure for partitioning graphs. Bell System Technical Journal. 1970; 49(2):291–307.
17. Goldberg DE, Holland JH. Genetic algorithms and machine learning. Machine Learning. 1988; 3(2):95–9.
18. Glover F. Heuristics for integer programming using surrogate constraints. Decision Sciences. 1977; 8(1):156–66.
19. Mercer RE, Sampson J. Adaptive search using a reproductive meta-plan. Kybernetes. 1978; 7(3):215–28.
20. Kirkpatrick S, Gelatt CD, Vecchi MP. Optimization by simulated annealing. Science. 1983; 220(4598):671–80.
21. Glover F. Tabu search: A tutorial. Interfaces. 1990; 20(4):74–94.
22. Moscato P. On evolution, search, optimization, genetic algorithms and martial arts: Towards memetic algorithms. Caltech Concurrent Computation Program. 1989; 826(1):67.
23. Dorigo M, Blum C. Ant colony optimization theory: A survey. Theoretical Computer Science. 2005; 344(2):243–78.
24. Mori K, Tsukiyama M, Fukuda T. Immune algorithm with searching diversity and its application to resource allocation problem. Transactions-Institute of Electrical Engineers of Japan C. 1993; 113(1):872.
25. Fonseca CM, Fleming PJ. Multiobjective genetic algorithms. IEE Colloquium on Genetic Algorithms for Control Systems Engineering; London. 1993. p. 1–5.
26. Battiti R, Brunato M. Reactive search optimization: Learning while optimizing. Handbook of Metaheuristics. Springer: Business Media, LLC; 2010. p. 543–71.
27. Eberhart RC, Shi Y. Particle swarm optimization: Developments, applications and resources. Proceedings of the Congress on Evolutionary Computation; Seoul. 2001. p. 81–6.
28. Storn R, Price K. Differential evolution–a simple and efficient heuristic for global optimization over continuous spaces. Journal of Global Optimization. 1997; 11(4):341–59.
29. Geem ZW, Kim JH. Loganathan G. A new heuristic optimization algorithm: Harmony search. Simulation. 2001; 76(2):60–8.
30. Karaboga D. An idea based on honey bee swarm for numerical optimization. Technical Report-tr06. Erciyes University, Engineering Faculty, Computer Engineering Department. Turkey. 2005.

31. Krishnan KN, Ghose D. Glowworm swarm optimization for simultaneous capture of multiple local optima of multimodal functions. Swarm Intelligence. 2009; 3(2):87–124.
32. Haddad OB, Afshar A. Marino MA. Honey-Bees Mating Optimization (HBMO) algorithm: A new heuristic approach for water resources optimization. Water Resources Management. 2006; 20(5):661–80.
33. Atashpaz-Gargari E, Lucas C. Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition. IEEE Congress on Evolutionary Computation (CEC'07); Singapore. 2007. p. 4661–7.
34. Shah-Hosseini H. Problem solving by intelligent water drops. IEEE Congress on Evolutionary Computation (CEC'07); Singapore. 2007. p. 3226–31.
35. Yang XS. Firefly algorithms for multimodal optimization. Stochastic Algorithms: Foundations and Applications. Berlin Heidelberg: Springer; 2009. p. 169–78.
36. Yang XS, Deb S. Cuckoo search via Levy flights. World Congress on Nature and Biologically Inspired Computing (NaBIC); Coimbatore. 2009. p. 210–4.
37. Rashedi E, Nezamabadi-Pour H, Saryazdi S. GSA: A gravitational search algorithm. Information Sciences. 2009; 179(13):2232–48.
38. Kashan AH. League championship algorithm: A new algorithm for numerical function optimization. International Conference of Soft Computing and Pattern Recognition (SOCPAR'09); Malacca. 2009. p. 43–8.
39. Yang XS. A new metaheuristic bat-inspired algorithm. Nature Inspired Cooperative Strategies for Optimization (NICSO 2010); Verlag-Berlin Heidelberg: Springer; 2010. p. 65–74.
40. Shah-Hosseini H. Principal components analysis by the galaxy-based search algorithm: A novel metaheuristic for continuous optimisation. International Journal of Computational Science and Engineering. 2011; 6(1-2):132–40.
41. Tamura K, Yasuda K. Spiral optimization-A new multipoint search method. IEEE International Conference on Systems, Man and Cybernetics (SMC); Anchorage, AK. 2011. p. 1759–64.
42. Rao RV, Savsani VJ, Vakharia D. Teaching–learning-based optimization: A novel method for constrained mechanical design optimization problems. Computer-Aided Design. 2011; 43(3):303–15.
43. Gandomi AH, Alavi AH. Krill herd: A new bio-inspired optimization algorithm. Communications in Nonlinear Science and Numerical Simulation. 2012; 17(12):4831–45.
44. Salcedo-Sanz S, Del Ser J, Gil-Lopez S, Landa-Torres I, Portilla-Figueras J. The coral reefs optimization algorithm: A new metaheuristic algorithm for hard optimization problems. Proceedings of the 15th International Conference on Applied Stochastic Models and Data Analysis (ASMDA); Mataro, Barcelona. 2013. p. 751–8.
45. Neshat M, Sepidnam G, Sargolzaei M. Swallow swarm optimization algorithm: A new method to optimization. Neural Computing and Applications. 2013; 23(2):429–54.
46. Gandomi AH. Interior Search Algorithm (ISA): A novel approach for global optimization. ISA Transactions. 2014; 53(4):1168–83.
47. Kuo RJ, Zulvia FE. The gradient evolution algorithm: A new metaheuristic. Information Sciences. 2015; 316(1):246–65.
48. Yazdani M, Jolai F. Lion Optimization Algorithm (LOA): A nature-inspired metaheuristic algorithm. Journal of Computational Design and Engineering. 2015. doi:10.1016/j.jcde.2015.06.003.
49. Kashan AH. A new metaheuristic for optimization: Optics Inspired Optimization (OIO). Computers and Operations Research. 2015; 55(1):99–125.
50. Zheng YJ. Water wave optimization: a new nature-inspired metaheuristic. Computers and Operations Research. 2015; 55(1):1–11.
51. Karaboga D, Akay B. A comparative study of artificial bee colony algorithm. Applied Mathematics and Computation. 2009; 214(1):108–32.
52. Karaboga D, Gorkemli B, Ozturk C, Karaboga N. A comprehensive survey: Artificial Bee Colony (ABC) algorithm and applications. Artificial Intelligence Review. 2014; 42(1):21–57.
53. Brownlee J. Clever algorithms: Nature-inspired programming recipes. Berlin Heidelberg: Springer; 2011. p. 97,176,243,275.
54. Kansal NJ, Chana I. Artificial bee colony based energy-aware resource utilization technique for cloud computing. Concurrency and Computation: Practice and Experience. 2015; 27(5):1207–25.
55. Chimakurthi L, Madhu Kumar SD. Power efficient resource allocation for clouds using ant colony framework. Distributed, Parallel and Cluster Computing. 2011:6.
56. 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. 2013; 79(8):1230–42.
57. Hu WX, Zheng J, Hua XY, Yang YQ. A computing capability allocation algorithm for cloud computing environment. Applied Mechanics and Materials. 2013; 347–50(1):2400–6.
58. Liang Y, Rui QP, Xu J. Computing resource allocation for enterprise information management based on cloud platform Ant Colony Optimization Algorithm. Advanced Materials Research. 2013; 791-793(1):1232–7.
59. Pacini E, Mateos C, Garino CG. Balancing throughput and response time in online scientific Clouds via Ant Colony Optimization (SP2013/2013/00006). Advances in Engineering Software. 2015; 84(1):31–47.

60. Liu XF, Zhan ZH, Du KJ, Chen WN. Energy aware virtual machine placement scheduling in cloud computing based on ant colony optimization approach. Proceedings of the Conference on Genetic and Evolutionary Computation; 2014. p. 41–8.
61. Arianyan E, Maleki D, Yari A, Arianyan I. Efficient resource allocation in cloud data centers through genetic algorithm. 6th International Symposium on Telecommunications (IST); Tehran. 2012. p. 566–70.
62. Portaluri G, Giordano S, Kliazovich D, Dorronsoro B. A power efficient genetic algorithm for resource allocation in cloud computing data centers. IEEE 3rd International Conference on Cloud Networking (CloudNet); Luxembourg. 2014. p. 58–63.
63. Radhakrishnan A, Kavitha V. Trusted virtual machine allocation in cloud computing IaaS service. Research Journal of Applied Sciences, Engineering and Technology. 2014; 7(14):2921–8.
64. Lin YK, Chong CS. Fast GA-based project scheduling for computing resources allocation in a cloud manufacturing system. Journal of Intelligent Manufacturing. 2015: 1–13.
65. Wang L, Pan J, Jiao L. The immune algorithm. Acta Electronica Sinica. 2000; 28(7):74–8.
66. Dai L, Li JH. An optimal resource allocation algorithm in cloud computing environment. Applied Mechanics and Materials. 2015; 733(1):779–83.
67. Shu W, Wang W, Wang Y. A novel energy-efficient resource allocation algorithm based on immune clonal optimization for green cloud computing. EURASIP Journal on Wireless Communications and Networking. 2014; 2014(1):1–9.
68. Wang X, Sun J, Huang M, Wu C, Wang X. A resource auction based allocation mechanism in the cloud computing environment. IEEE 26th International Parallel and Distributed Processing Symposium Workshops and PhD Forum (IPDPSW); Shanghai. 2012. p. 2111–15.
69. Abdulhamid SM, Latiff MSA, Madni SHH, Oluwafemi O. A survey of league championship algorithm: Prospects and challenges. Indian Journal of Science and Technology. 2015; 8(S3):101–10.
70. Kashan AH, Karimi B. A new algorithm for constrained optimization inspired by the sport league championships. IEEE Congress on Evolutionary Computation (CEC); Barcelona. 2010. p. 1–8.
71. Sun J, Wang X, Li K, Wu C, Huang M, Wang X. An auction and league championship algorithm based resource allocation mechanism for distributed cloud. Advanced Parallel Processing Technologies. Berlin Heidelberg: Springer; 2013. p. 334–46.
72. Xing B, Gao WJ. Innovative computational intelligence: A rough guide to 134 clever algorithms, Intelligent Systems Library. Berlin Heidelberg: Springer; 2014. p. 260–1.
73. Kennedy J. Particle swarm optimization. Encyclopedia of Machine Learning. US: Springer; 2010. p. 760–6.
74. Shi Y, Eberhart R. A modified particle swarm optimizer. IEEE International Conference on Evolutionary Computation Proceedings of World Congress on Computational Intelligence; Anchorage, AK. 1998. p. 69–73.
75. An-Ping X, Chun-Xiang X. Energy efficient multiresource allocation of Virtual Machine based on PSO in cloud data center. Mathematical Problems in Engineering. 2014; p. 8.
76. Kumar D, Raza Z. A PSO based VM resource scheduling model for cloud computing. IEEE International Conference on Computational Intelligence and Communication Technology (CICT); 2015. p. 213–9.
77. Dashti SE, Rahmani AM. Dynamic VMs placement for energy efficiency by PSO in cloud computing. Journal of Experimental and Theoretical Artificial Intelligence. 2015:1–16.
78. Banu AS, Helen W. Scheduling deadline constrained task in hybrid IaaS cloud using cuckoo driven particle swarm optimization. Indian Journal of Science and Technology. 2015; 8(16):6.
79. Mohana RS. A position balanced parallel particle swarm optimization method for resource allocation in cloud. Indian Journal of Science and Technology. 2015; 8(S3):182–8.
80. Ingber L. Simulated annealing: Practice versus theory. Mathematical and Computer Modelling. 1993; 18(11):29–57.
81. Fan Z, Shen H, Wu Y, Li Y. Simulated-annealing load balancing for resource allocation in cloud environments. International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT); Taipei. 2013. p. 1–6.
82. Wang W, Chang X, Liu J, Wang B. Simulated annealing based resource allocation for cloud data centers. Proceedings of the 15th Annual Conference Companion on Genetic and Evolutionary Computation; 2013. p. 81–2.
83 Wu Y, Tang M, Fraser W. A simulated annealing algorithm for energy efficient virtual machine placement. IEEE International Conference on Systems, Man and Cybernetics (SMC); Seoul. 2012. p. 1245–50.
84. Shyamala K, Rani TS. an analysis on efficient resource allocation mechanisms in cloud computing. Indian Journal of Science and Technology. 2015; 8(9):814–21.
85. Sarma VAK, Rajendera R, Dheepan P, Kumar KS. An optimal ant colony algorithm for efficient VM placement. Indian Journal of Science and Technology. 2015; 8(S2):156–9.
86. Abdulhamid SM, Latiff MSA, Idris I. Tasks scheduling technique using league championship algorithm for makespan minimization in IaaS cloud. ARPN Journal of Engineering and Applied Sciences. 2015; 9(12):2528–33.
87. Bashir MB, Latiff MSA, Ahmed AA, Yousif A, Eltayeeb ME. Content-based information retrieval techniques based on

grid computing: A review. IETE Technical Review. 2013; 30(3):223–32.

88 Zhang Q, Cheng L, Boutaba R. Cloud computing: Sate-of-the-art and research challenges. Journal of Internet Services and Applications. 2010; 1(1):7–18.

89. Calheiros RN, Ranjan R, Beloglazov A, De Rose CA, Buyya R. CloudSim: A toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms. Software: Practice and Experience. 2011; 41(1):23–50.

90. Buyya R, Ranjan R, Calheiros RN. Modeling and simulation of scalable Cloud computing environments and the CloudSim toolkit: Challenges and opportunities. International Conference on High Performance Computing and Simulation (HPCS'09); Leipzig. 2009. p. 1–11.

91. Buyya R, Murshed M. Gridsim: A toolkit for the modeling and simulation of distributed resource management and scheduling for grid computing. Concurrency and Computation: Practice and Experience. 2002; 14(13-15):1175–220.