Hybrid gradient descent cuckoo search (HGDCS) algorithm for resource scheduling in IaaS cloud computing environment

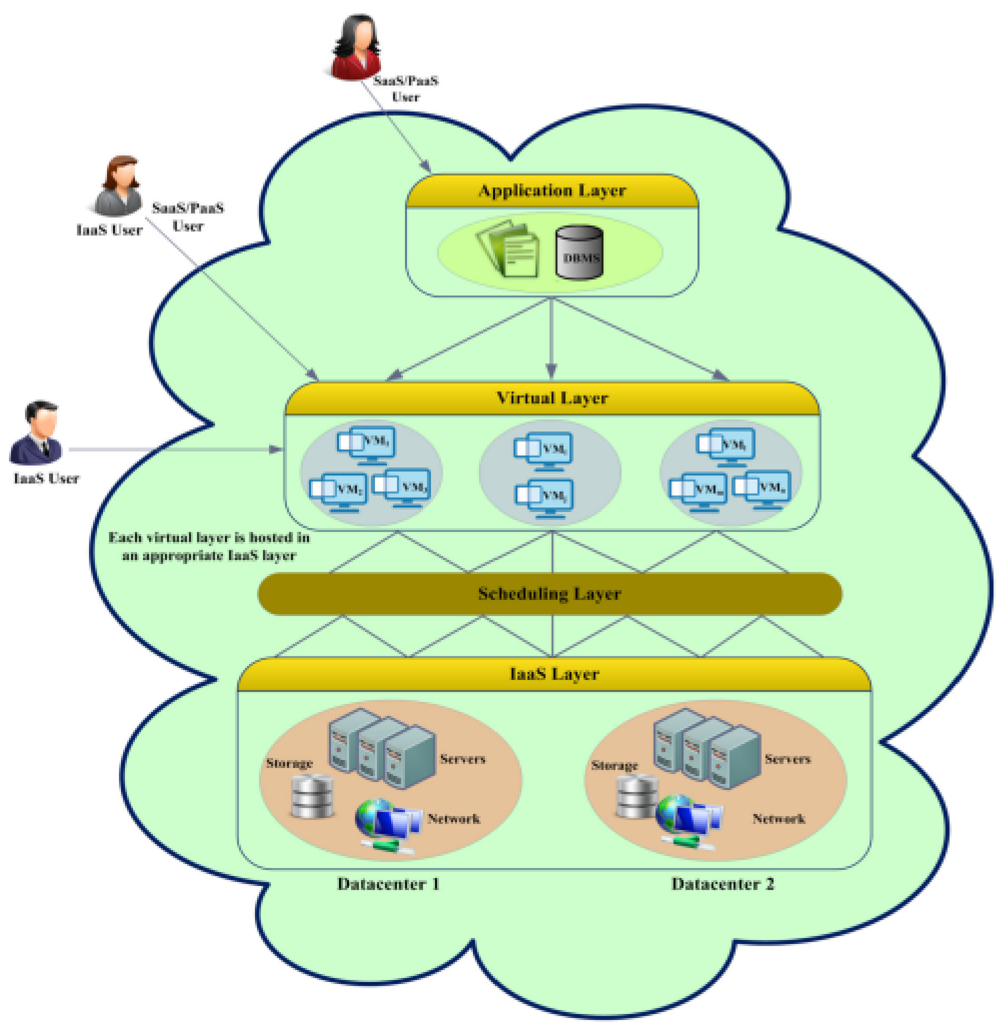
Received: 3 November 2017 / Revised: 11 June 2018 / Accepted: 14 October 2018

Abstract
Resource scheduling is a procedure for the distribution of resources over time to perform a required task and a decision making process in cloud computing. Optimal resource scheduling is a great challenge and considered to be an NP-hard problem due to the fluctuating demand of cloud users and dynamic nature of resources. In this paper, we formulate a new hybrid gradient descent cuckoo search (HGDCS) algorithm based on gradient descent (GD) approach and cuckoo search (CS) algorithm for optimizing and resolving the problems related to resource scheduling in Infrastructure as a Service (IaaS) cloud computing. This work compares the makespan, throughput, load balancing and performance improvement rate of existing meta-heuristic algorithms with proposed HGDCS algorithm applicable for cloud computing. In comparison with existing meta-heuristic algorithms, proposed HGDCS algorithm performs well for almost in both cases (Case-I and CaseII) with all selected datasets and workload archives. HGDCS algorithm is comparatively and statistically more effective than ACO, ABC, GA, LCA, PSO, SA and original CS algorithms in term of problem solving ability in accordance with results obtained from simulation and statistical analysis

Keywords Meta-heuristic algorithms · Resource scheduling · Cuckoo search · Gradient descent · Hybridization · Cloud computing

1 Introduction

Cloud computing is one of the most popular technologies that has become a fundamental part of the computing world nowadays. Its popularity and usage is growing every dayand expected to increase further. Cloud computing provides IT services over the Internet in such a way that cloud user does not have knowledge about where the data or information is being stored, where the infrastructure is located and so on. The cloud users receive services without knowing any of the details about how it's provided [1, 2]. Organization spends time and money to scale up their IT infrastructure, such as hardware, software and services provisioning, to meet the business challenges. However, the scaling up process is slow with unpremises IT infrastructures. Cloud computing paradigm shift provides computing over the Internet. Cloud computing services contain highly optimized virtualized data centers providing hardware, software and information resources for use, whenever they are required [3]. In certain circumstances, cloud computing deals workload fluctuations and provides computation resources to manage large multimedia data and development environments. However, the success of cloud computing, resulting in more and larger datacenters,creates new challenges at the level of infrastructure monitoring and management [4–6]. Resource scheduling is an important procedure in cloud computing that is used to take decision in resource distribution over time. However, there are some challenges in resource scheduling such as NP-hard problem due to fluctuating demand of cloud users and dynamic nature of various resources shown in Fig. 1. In cloud computing, the fitness function of resource scheduling is focused on cloud providers' and cloud users' objectives. Cloud providers want to enhance the utilization of resources by increasing the growth of revenue and profit, on the other side cloud users want to get maximum performance of required service with minimum cost or expenditure [7]. The following objective functions are generally considered for optimum resource scheduling in cloud computing including the availability, cost, energy, fault tolerance, load balancing, makespan, reliability, throughput, etc. An optimization problem is a function f mapping candidate solution to a fitness measure $\frac{1}{4}$ Rn ! R. The optimization solution z 2 Rn achieves the best optimal solution from all feasible solutions by proposed algorithm. Therefore it can be satisfies the minimization .



Hybrid algorithm tries to attain all the factors whereas other algorithms fail to do so. Hybridization intentions gather the benefits of all algorithms in the form of a hybrid algorithm, even though at the same time try to reduce the extensive disadvantages. Generally, the consequences and outcomes of hybrid algorithms are commonly made some improvements in terms of either accuracy or speed. Optimal resource scheduling is considered to be a basic influence of cloud computing. It is only achieved by adopting the specific, enhanced and hybridization of greedy approaches with meta-heuristic algorithms. CS algorithm is a nature-inspired meta-heuristic algorithm in which all entities have identical search behaviour. However, this simple uniform search behaviour is not always optimal to find the feasible solution for the specific problem and trap it into the local region leading to premature convergence. To overcome this weakness in this paper, we combined GD approach in term of local search with CS algorithm for enhancing the convergence rate, in order to assign a task to the specific Virtual Machine (VM) with lowest execution time to fulfill the cloud user's demand and enhance the resource utilization for cloud providers with less delay. In conclusion, a novel HGDCS algorithm with non-identical search strategy is proposed for resource scheduling in IaaS cloud computing, which reduces the makespan, throughput and degree of imbalance. Our major contributions of this paper are as follows: Formulate the makespan, throughput and degree of imbalance through mathematical models for optimal resource scheduling as the objective functions. Hybridize the GD approach with CS algorithm for optimum resource scheduling in IaaS cloud computing. Design the HGDCS algorithm to address the proposed scheduling models. Implementation of the proposed HGDCS algorithm in CloudSim simulation tool. Performance evaluation of the existing meta-heuristic algorithms with HGDCS optimization algorithm by considering the matrices of makespan, throughput, degree of imbalance and performance improvement rate (Comparative and statistical analysis of the HGDCS algorithm with existing meta-heuristic algorithms for resource scheduling in IaaS cloud computing. The remaining sections of this paper are systematically prepared as follows: In Sect. 2, we review the current comparison based studies and related works for resource scheduling algorithms in the area of IaaS cloud computing. Problem formulation is discussed in Sect. 3. We provide the description of local and global search, GD approach, CS algorithm and HGDCS algorithm in Sect. 4. Section 5 defines the parameters for evaluation, while Sect. 6 presents simulation setups for Case-I and Case-II for resource scheduling in cloud computing. Results and discussion show performance evaluation with the help of experimental simulation and statistical analysis in Sect. 7. The last section is Sect. 8, which consists of details of the conclusion, recommendation and future works.

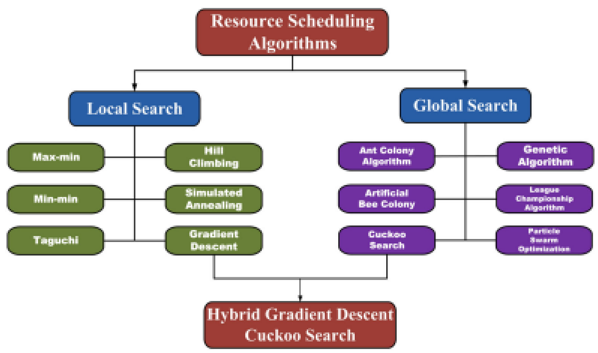
2 Related works

In this section, we review the current comparison and existing studies of various heuristic, meta-heuristic and hybrid algorithms for resource scheduling in IaaS cloud computing. Tsai and Rodrigues [8] provide the systematic explanation of scheduling in cloud computing and also interrelated it with heuristic and meta-heuristic algorithms. Besides it, recommends instructions for the researchers to shift from heuristic algorithms to the meta-heuristic algorithms. In conclusion, meta-heuristic algorithms produce better solutions by modification of operators, alteration of the fitness function and hybridization with heuristic algorithms [9]. Further, Thaman and Singh [10] classify the algorithms as Greedy, Heuristics, Meta-heuristics and Genetic approaches based solutions for task scheduling in cloud computing. The various meta-heuristic algorithms are analyzed for scheduling the cloud computing, including the ant colony optimization (ACO), genetic algorithm (GA), hill climbing, Particle swarm optimization (PSO), simulated annealing (SA) and Tabu Search with a discussion and future direction of a scheduling problem in cloud [8]. Further, Kalra and Singh [11] evaluate the ACO, BAT algorithm, GA, League Championship Algorithm (LCA) and PSO metaheuristic algorithms for scheduling in grid and cloud computing with observations and open issue related to scheduling are also discussed. Similarly, Madni et al. [12] review and analyze the artificial bee colony (ABC), ACO, GA, immune algorithm (IA), LCA, PSO and SA metaheuristic algorithms for assigning of resources in cloud computing. Also, provide the description of selected important parameters are applied for this problematic issue, cloud computing. Besides the good qualities each of them is slow in convergence speed. Also, Singh et al. [15] accomplish the an extensive review of various metaheuristics-based techniques for scheduling in cloud computing including the ACO, bat algorithm (BA), GA, imperialists competitive algorithm (ICA), lion algorithm (LA), PSO and wind-driven optimization (WDO) algorithms with describing the various factors. Hence, these algorithms accomplish them no attention for resource scheduling in cloud computing environment. Cui et al. [16] enhance the GA algorithm to accomplish a relevant task for the effective resource scheduling in cloud computing. Further, Chen et al. [17] merge GA algorithm with knapsack problem for utilization of resources and energy consumption in cloud computing. However, Sindhu and Mukherjee [18] combine the GA algorithm with various heuristic algorithms including the LCFP, SCFP and MCT for enhancing the convergence rate. Similarly, Javanmardi et al. [19] propose a hybrid job scheduling algorithm based on fuzzy theory and GA algorithm for decreasing the number of iteration to generate the population and accurate allocation of resources in order to node capacity. Moreover, Shojafar et al. [20] propose a hybrid algorithm based on fuzzy theory and GA (FUGA) algorithm for assigning the jobs to the optimal resources. For this purpose, fuzzy theory and GA modify with diverse fuzzy based stable state GA in order to minimize the makespan, execution cost, execution cost and average degree of imbalance. Later, Saha et al. [21] merge the GA algorithm and Queuing model as a tool for task scheduling algorithm for minimizing the waiting time and queue length for substantial the cloud users' demand in cloud computing. In cloud computing, Zhang et al. [22] introduce the PSO algorithm in order to achieve the local and global search efficiently by weight inertia strategy and escape the struggling into a local optimum while Netjinda et al. [23] propose PSO algorithm for enhancing performance from the views of the entire cost and fitness convergence rate. The improvement in solution quality is perceived from the preliminary schedule with the variable neighbour search. Further, to minimize the consumption of energy and enhance the cloud providers' profit, job scheduling model is developed by using the PSO algorithm in cloud environment [24]. In addition, Abdi et al. [25] improve the PSO algorithm for enhancing the performance, merge with SJFP algorithm for improving the local search for reducing the makespan. However, Al-Olimat et al. [26] propose the hybrid scheduling algorithm by using the SA algorithm to enhance the performance of binary PSO algorithm. The proposed hybrid algorithm is used to reduce the makespan and enhance the utilization of resource in cloud computing. For optimal resource scheduling, Wang and Yu [27] improve the Min-min algorithm for enhancing the performance of scheduling in cloud computing by considering the makespan and degree of imbalance. Hence, Li et al. [28] develop a Load Balancing aware ACO algorithm for task scheduling by considering the makespan and degree of imbalance in cloud computing. further, Tawfeek et al. [29] present a cloud task scheduling policy by using the ACO algorithm for reducing the makespan and degree of imbalance. While, Wen et al. [30] and Yang [31] combine the ACO and PSO for improving the local optimization for efficient resource scheduling in cloud computing. Similarly, Cho et al. [32] propose hybrid ACOPS algorithm for VM scheduling through load balancing by enhancing the convergence speed. In the same way, Liu et al. [33] introduce hybrid GA-ACO algorithm for optimal solution quickly by considering convergence speed for task scheduling. Optimized scheduling and VM validation are required for the optimum resource allocation in IaaS cloud computing. For this purpose, Muthulakshmi and Somasundaram [34] propose the Hybrid ABC-SA algorithm for efficient scheduling that improves the efficiency in terms of resource time searching by dynamic and random search. Abdullahi, et al. [35] propose a discrete symbiotic organism search (DSOS) algorithm for an ultimate schedule of tasks to VMs in order to improve the convergence rate in cloud computing. Further, Abdullahi and Ngadi [36] merge SA and symbiotic organisms search (SOS) algorithms in SASOS algorithm for achieving ideal scheduling of tasks in order to attain better convergence ratio and quality of results in cloud computing. Similarly, Tsai et al. [37] improve the differential evaluation algorithm (DEA), by combining the Taguchi method for optimal resource scheduling. Improved DEA shows the ideal results for reducing cost and makespan. While, Guddeti and Buyya [38] propose hybrid bio inspired algorithm combination of modified PSO and cat swarm optimization (CSO) algorithms to assign a task to VMs in order to enhance the reliability, response time and resource utilization. Hence, Gabi et al. [39] put forward a new version of CSO for cloud task scheduling called Orthogonal Taguchi based-cat. The researchers explored the Taguchi approach to optimize task scheduling that returned minimum task makespan as compared to other existing tasks scheduling algorithms. For optimized parameter mapping, Moon et al. [40] propose a slave ants based ACO (SACO) algorithm that schedules tasks to virtual machines (VMs) of cloud users in cloud computing environments in a competent way. The global optimization problem is solved by avoiding long paths with slave ants, whose pheromones are imperfectly accumulated by leading ants and minimal preprocessing. Experimental results SACO algorithm performed better than ACO algo

3 Problem formulation

Scheduling consists of the assignment of starting and completion times for the various operations to be performed. Like other scheduling problems, resource scheduling in cloud computing is a method that applies to the distribution of valuable cloud resources, generally processors, networks, storage and VMs to fulfill the demands of cloud users by the cloud providers. It is applied for balancing the load, ensure equal distribution of resources according to the demand and give some prioritization according to set rules. It also ensures that a cloud computing is able to serve all the cloud users' requests, with a certain quality of service. Resource scheduling problem can clarify with the help of Eq. (2) and Fig. 1. RS $\frac{1}{4}$ Xm;n x$\frac{1}{4}$1 Rx Sx...Nx Tx ! UZ x 2 where it assigns m required numbers of cloudlets/task T $\frac{1}{4}$ T1; T2; T3; ...; Tm onto n available physical resources to virtual resources in cloud data centers R $\frac{1}{4}$ R1; R2; R3; ...; Rn , S $\frac{1}{4}$ S1; S2; S3; ...; Sn up to N $\frac{1}{4}$ N1; N2; N3; ...; Nn . The Fitness of particular objective F ¼ F1; F2; F3; ...; Fz may be enhanced for the cloud users U ¼ U1; U2; U3; ...; Un . When Z=1, the fitness function F1 is assigned to the cloud users, when Z= 2, the fitness function F2 is assigned to the cloud users and so on, according to the their demand. Cloud computing consists of various datacenters and all datacenters are interrelated with VMs with different specification. Suppose there is a set of cloudlet/task Ti ¼ T1; T2; T3; ...; Tn that are originated from the cloud users as their required demands. Cloud broker is responsible for assigning the cloudlet/task to requisite virtual resources Vj ¼ V1; V2; V3; ...; Vm as virtual resources with minimum completion time. The expected time to completion (ETC) is described as the expected time of all cloudlets/tasks are execute on a definite virtual resource acquired by using ETC matrix as shown in Eq. (3). Total number of cloudlets/tasks multiply by the total number of resources gives the ETC matrix's dimension and their elements are characterized as an ETC Ti; Vj

Fig. 3 Composition of hybrid gradient descent cuckoo search (HGDCS) algorithm



4 Methodology

his section presents the description of the local and global search, the basic structure of GD approach, standard CS algorithm and proposed HGDCS algorithm in detail

4.1 Local search versus global search

A search technique which always reaches the same locally optimal solution from the same starting point is probably a local search technique. Equally, the performance of a global search technique should be less dependent on its initial position(s). Whereas a local search technique will target nearby local optima, global search techniques should be able to find (local) optima anywhere in the search space. Both search techniques attempt to find a solution that optimizes a cost criterion. Local search algorithms start exploring the state space in a certain point of the state space (this point can be selected using a huge variety of techniques. These techniques are highly dependent on the problem domain and the local search algorithm), and iteratively try to find a better solution in terms of the cost function. In general, these algorithms are faster than other global search techniques and they can provide quite good solutions if the initialization step is adequated to the problem. Also, these algorithms are iterative, and we always know the best found solution at the current iteration. This leaves us total freedom to select the stop condition. As other answers point, these algorithms only provide local optimas, which may have a much higher cost than the global one, and which also depend on the initial solution in which the exploration started. Ideally speaking, a global searching technique is promised to make sure to find the best global formation but this is achieved mostly at the cost of a long time searching. But then again in reality, they are run and stop when the stopping criterion comes across. Examples of this search include the PSO, SA, GA, etc. Whereas, local search algorithms do not totally focus on search and but it attempts to move from a current formation to a neighboring refining formation. This is much depending on the initial search space and initial formation. An example of a local search is hill climbing algorithm, which is an iterative algorithm which can start with a random solution and then after the algorithm tries to find a better solution by incrementally altering a solution of a single element. If this alteration harvests a better solution, an incremental alteration is made to a new solution, this process can be repeated in anticipation of no more enhancement is identified. There are NP problems where finding one optimal and the definite solution are not possible. From a classification

Fig. 4 Pseudo-code of hybrid gradient descent cuckoo search (HGDCS) algorithm

Pseudo-Code of Hybrid Gradient Descent Cuckoo Search Algorithm

```
Input: (Pα, λ).
Output: S_best
    f(x), x = (x_1, ..., x_d)^T;                  // Objective function
    Initial x_i (i = 1, 2, ..., n),                // population of n host nests
    Pα ∈ [0,1] and Max_itr;
    while : ((t < Max_itr) or (Stop Condition)) do
        Get a cuckoo (say i) randomly by Eq 8 and Eq 9 or Eq 11;
                                                   // new solution x_i^(t+1)
        Check F_i = f(x_i^(t+1))                   // evaluate its quality/fitness F_i
        Choose a nest among n (say j) randomly; // old solution x_i^t
        if (F_i > F_j) then                        // x_i^(t+1) > x_i^t
            F_j ← F_i;                             // Replace old solution x_i^t with new solution x_i^(t+1)
        end if
        if (rand [0,1] < Pα) then                  // abandon a fraction (Pα) of worse nests and
            init_nest (x_worst);                   // build new ones at new locations
        end if
        if (F_i < F_min)then                       // rank the solutions and find the current best
            x_best = x_i;
            F_i = F_min;                           // keep the best solutions or nest with quality solutions
        end if
        t←t+1;
    end while
    return S_best ;
    end
```

point of view, this differs from the method used to find the solution. Any method that searches in the vicinity of a starting point (incremental methods) and has the potential to get stuck the moment it sees extrema is a local method. GD approach is the classic example for this. Global methods will usually treat the whole feature space as one when finding the best solution. A very slow and a primitive example would be the exhaustive search. 4.2 Gradient descent approach Gradient descent (GD) is a principal order iterative optimization method. It is used for finding a local minimum of a function, one takes steps proportional to the negative of the gradient (or of the approximate gradient) of the function at the current point. If instead one takes steps proportional to the positive of the gradient, one approaches a local maximum of that function, the procedure is then known as a "Sleepest Ascent Method" [42, 43]. Gradient descent is founded on the observation that if the multi-variable function F(x) is distinct and differentiable in a neighborhood of a point a, then F(x) decreases fastest if one goes from a in the direction of the negative gradient of F at a, rFa. It follows that, if b ¼ a crFa 7 where a is the current position, b is the next position, is the weight factor, rFa is the direction of sleepest ascent. For c minor sufficient, then F a Fb. In other words, the term crFa is subtracted from a because we want to move against the gradient, namely down toward the minimum. With this observation in mind, one starts with a guess x0 for a local minimum of F, and considers the sequence x0; x1; x2... such that: xn1 ¼ xn cnrF x n ; n[1 8 then y ¼i axi yi 2 byi yi1 2 byi yi1 2 9 or y 0 i ¼ yi 2axi yi 2byi1 yi 2yi 2 10 So we have x0 F x 0 F x 1 F x 2 ...: So hopefully the sequence xn converges to the desired local minimum. Note that the value of the step size c is allowed to change with each iteration. With certain assumptions on the function F and particular choices of c, convergence to a local minimum can be guaranteed. The behaviour of GD approach is illustrated in Fig. 2. Here F is assumed to be defined on the plane, and that its graph has oval shapes. The blue curves lines are the contour lines, that is, the regions on which the value of F is constant. A red dotted line creating at a point shows the direction of the negative gradient at that point x0 to xn. Note that the (negative) gradient at a point is orthogonal to the contour line going through that point. We see that gradient descent indicates to the center of the oval shape, that is, the point where the value of the function F is minimal. 4.3 Cuckoo search algorithm Cuckoo search (CS) is created by "Xin-She Yang" and "Suash Deb". CS has been originated to contribute in global optimization problems [44]. Cuckoo in real is a captivating bird, not only because of the sweet sound but al