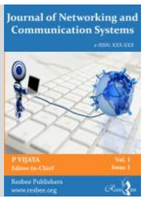
Resbee Publishers

Journal of Networking and Communication Systems

Received 27 August, Revised 28 September, Accepted 23 October



Multi-Objective Task Scheduling in Cloud Computing Using Monkey Search Approach Hybridized with Krill Herd Algorithm

Karnam Sreenu
Sreenidhi Institute of Science and Technology
Hyderabad, Telangana, India
karnamsreenu.27@gmail.com

Sreelatha M.
RVR & JC College of Engineering,
Chowdavaram, Andhra Pradesh, India

Abstract: In cloud computing, optimization of computing resources needs a scheduling method hence the user-requested task is scheduled efficiently. In contrast to the effectiveness, the developed task scheduling methods should attain user needs. Even though there are numerous techniques for task scheduling, the techniques that stated multiple objectives with represented trade-off are infrequent. Here, in a cloud computing environment, a multi-objective optimization technique, named Hybrid Monkey Search with Krill Herd algorithm for Multi-Objective Task Scheduling. The objectives, such as communication cost, resource utilization, execution cost, energy consumption, execution time, and communication time, are calculated exploited penalty cost function and epsilon-constraint. This novel constraint reduces the fitness function, to offer optimal task scheduling. The proposed technique is compared with the conventional Grey Wolf Optimizer, Monkey Search Algorithm (MSA), Krill Herd (KH), and to verify the performance efficiency.

Keywords: Cloud Computing; Task Scheduling; Optimization Model; Cost Functions.

1.Introduction

Generally, cloud computing is considered as a computing technique, that offers the unlimited amount of computing resource to end-user on the basis of their demands, at any time and anywhere in “pay-as-you-go” model whereas users recompense merely for the services they utilize [1]. From cloud because of the centralized management for the cloud infrastructure, users have the ability to ingress several kinds of services namely utility services, resource pooling, throughput, and elastic and flexible, scalability, managed services, performance, high availability, so on. Cloud service providers and users have the ability to leverage the advantage of dynamic resource scheduling methods and virtualization in technology [17] [18]. Efficient scheduling of the resource not merely performs the tasks in minimal time but as well as maximize the ratio of the resource utilization that minimizes consumption of the resource [19]. Because of the increase in workload, the task scheduling becomes a huge concern circumstance, without interruption at the cloud datacenters, which might tend to the cloud resources shortage [2]. Therefore cloud computing is still in its state as well as numerous investigation is needed to plan the tasks with cloud resources effectively as well as to attain the scheduling objective. Scheduling objective is to specify optimal resource in order to perform several tasks hence that scheduling technique can enhance several Quality of Services (QoS) parameters namely reliability, utilization of resource, execution cost, ration of task rejection, consumption of energy, so on. without affecting Service Level Agreement (SLA), taking into consideration of constraint namely priority, deadline, priority, etc as well as evade the load imbalance overexploited, and underexploited issue.

Although increasing the advantages of the economic cloud service providers, the task scheduling is developed for the logical exploitation of cloud resources to attain the user needs. In clouds, the task scheduling schemes are required as a service model in order to attain the QoS needs of user tasks [7], that comprise deadline [8], makespan [9], as well as cost. Simultaneously, for cloud service providers, service profit and energy cost must be completely measured [10]. At a few hardware platforms inappropriately matching applications can mortify the complete cloud performance and might infringe the QoS guarantees, which huge user tasks need.

For clouds, exhaustive-search-based scheduling is unfeasible. Moreover, its intricacy rises exponentially with the number of resources and tasks. To discover an estimated optimal solution numerous scholars exploit intelligent optimization methods. Such methods [2], [8] have the ability to minimize their search space and assure their execution within a possible operational time that offers cooperation among their ensuing schedule's optimality and running time. Particle Swarm Optimization (PSO) [10], Genetic Algorithms (GA) [8], Ant Colony Optimization (ACO) [9] and Cuckoo search [9] were their envoy instances. There are various classical scheduling methods, like Min–Min [11], Max-Min [12], and First-In-First-Out (FIFO) [13]. The first two be unsuccessful to use resources competently, thus foremost to a load imbalance issue.

This paper exploits a method named, Hybrid Monkey Search with Krill Herd algorithm in a cloud environment for task scheduling. This scheduling strategy performs resource allocation for each task offering the scheduling with the consideration of objectives, namely execution cost, communication time, energy consumption, execution time, resource utility and communication cost. Additionally, to handle the scheduling the objective model of the proposed method exploits the penalty cost technique as well as epsilon constraint.

2. Literature Review

In 2019, Mohit Kumar et al [1] worked on the scheduling, which was to distribute tasks between the cloud resources. Hence, the scheduling method evades the issue of inequity. Additionally, the scheduling method must optimize the key performance indicator parameters such as availability, reliability, resource utilization cost, makespan time, energy consumption, response time, and so forth. On the basis of the meta-heuristic, heuristic, as well as hybrid method, to accomplish the aforesaid objective, numerous conventional scheduling methods were presented. Moreover, this article offers a systematic classification and review of the proposed scheduling algorithm besides the benefits and restrictions.

In 2018, Luiz F. Bittencourt et al [2], worked on a classification method for the scheduling issue in distributed systems. In cloud computing, it was done by introducing a classification, which integrates current advancements. The scheduling state-of-arts to confirm the classification and examine the attention in various branches of the proposed classification was reviewed. At last, for distributed systems appropriate future directions was identified in scheduling.

In 2018, PeiYun Zhang et al [3], presented a technique on the basis of a dual-stage scheme. Initially, using a Bayes classifier's model principle a job classifier was motivated to classify tasks on the basis of the past scheduling data. A definite amount of Virtual Machines (VMs) of various kinds were therefore produced. Hence, by producing VMs at some phase in task scheduling time can be saved. With concrete VMs dynamically during the second phase, tasks were coordinated. Accordingly, dynamic task scheduling techniques were presented.

In 2018, Songtao Guo et al [4], proposed an Energy-Efficient Dynamic Offloading and Resource Scheduling (eDors) strategy to minimize energy utilization as well as cut down application accomplishment time. Initially, eDors issue into an Energy-Efficiency Cost (EEC) reduction issue when fulfilling task-dependency obligation and completion time deadline restraint was formulated. A distributed eDors technique comprises of three sub-techniques of computation offloading chosen, transmission power allocation, and clock frequency control was proposed.

In 2018, Li Liu et al [5], modeled the task schedule was as a multi-objective optimization issue, as well as both time deadline constrained as well as unconstrained cases were considered. A Heterogeneous Earliest Finish Time (HEFT) exploiting TOPSIS approach (Technique for Order Preference by Similarity to an Ideal Solution) was presented, which was called as HEFT-T technique to address the aforesaid issue. A three-phase scheme on the basis of the HEFT-T technique was proposed for the unconstrained scenario.

In 2016, Min Chen et al [6], introduced a novel kind of peer-to-peer communication form for mobile cloud computing. For holding computation-intensive tasks, though the example of mobile cloudlet was cost-efficient, the perceptive of its equivalent service form from a theoretic viewpoint was still in it's the early stage in the development. Initially, a novel mobile cloudlet aided service form called Opportunistic task scheduling over Co-located Clouds (OSCC) was proposed, that attains flexible cost-delay tradeoffs among existing mobile cloudlets service as well as remote cloud service mode form. Subsequently, an analytic examination for OSCC form, and resolve the energy reduction issue was performed by cooperating between mobile cloudlets mode, remote cloud mode, and OSCC mode.

3. Cloud Computing Framework

3.1 Objective Model

The main objective of the task scheduling method is resource allocation for the tasks as a result of the subtasks. Hence, for the user request tasks, the parallel execution starts. Each task, indicated as $Y = \{y_1, y_2, \ldots, y_n\}$ is divided into subtasks $Y_j^M = \{y_j^1, y_j^2, \ldots y_j^k\} 1 \le M \le |y_j|$. The resource allocation to the subtasks based upon the P_c indicates processing capacity and the R_c indicates the resource cost of the virtual machine $U = \{u_1, u_2, \ldots, u_k\}$, stated by the physical server $P_s = \{P_{s1}, P_{s2}, \ldots, P_{sn}\}$. Additionally, it calculates the CPU, bandwidth and the memory information, effectiveness for the scheduling of the resources to the tasks. With the appropriate beneficial objectives of energy consumption, resource utilization, communication time, execution time, communication cost, execution cost, it is feasible to attain enhanced task scheduling performance.

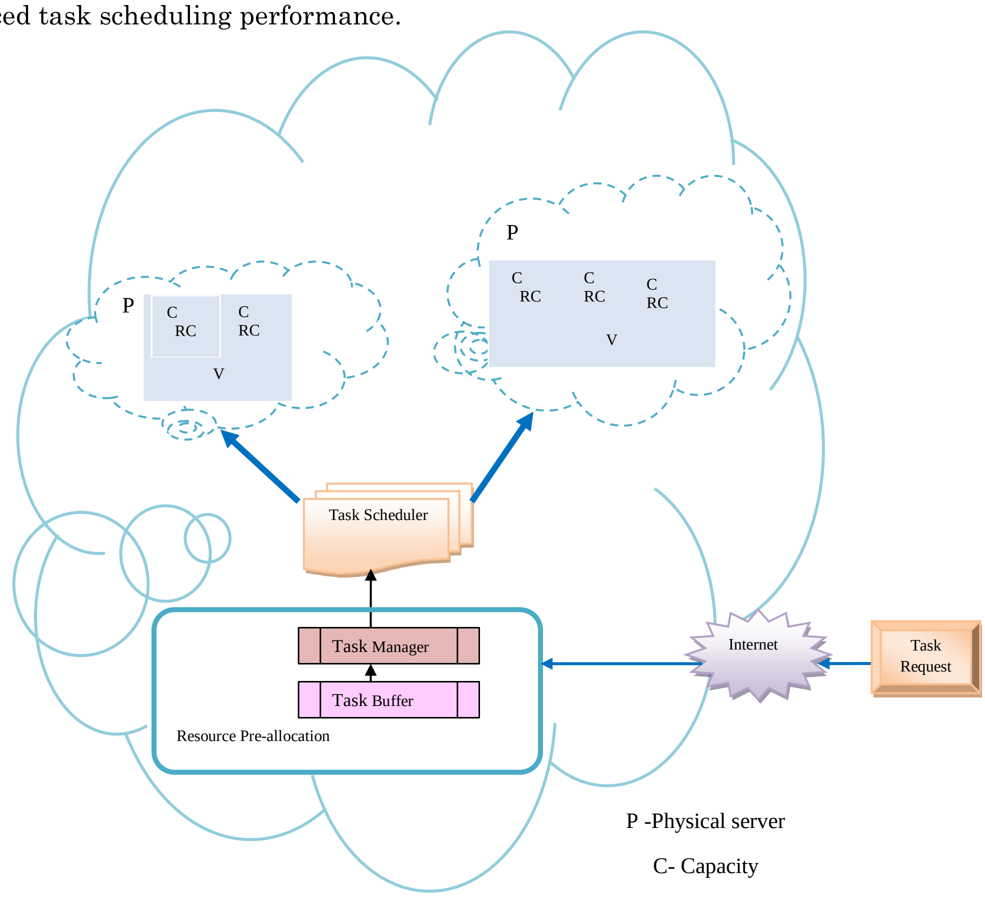


Fig. 1. *Cloud computing Model*

The framework of cloud computing is demonstrated in fig 1. The cloud computing main objective is to perform the resource allocation so that it attains the user demands. Moreover, various physical servers consist of the data centers, which offer allocation of the resource. The physical servers, sequentially, possess various resources or virtual machines. Consider the physical server as $P_s = \{P_{s_1}, P_{s_2}, \ldots, P_{s_n}\}$, each of that comprises of $U = \{u_1, u_2, \ldots, u_k\}$ as virtual machines. Here, in the physical server P_{c_1} two virtual machines are contained, in P_{c_2} three virtual machines hitherto. In order to perform the tasks solicited by the user the virtual machines possess various abilities. Aforesaid abilities of virtual machines comprise a resource cost and a processing capacity, indicated as P_c and R_c correspondingly. For each task request, the servers allocate resources on the basis of the resource cost and processing capacity; hence each task could be allocated merely one virtual resource from a physical server.

3.2 Allocation of Task

In this section, the user solicited tasks, which are partitioned into subtasks; subsequent to that the servers schedule the resources. Consider Y as a task, and it is indicated as $Y = \{y_1, y_2, \ldots, y_j, \ldots, y_n\}$, whereas n indicates the number of tasks. As eq. (2), each task Y consists of subtasks.

$$\text{Subtasks} = \sum_{j=1}^{n} |y_j| \tag{1}$$

$$y_j^M = \{y_j^1, y_j^2, \ldots y_j^k\}; 1 \le M \le |y_j| \tag{2}$$

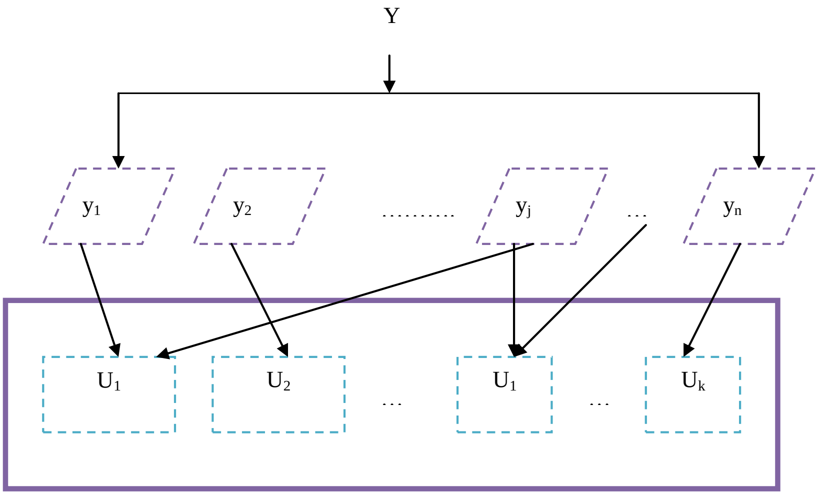


***Fig. 2.** Diagrammatic representation of task scheduling*

Fig. 2 demonstrates the resource allocation for the subtasks of the corresponding tasks. Any two subtasks comprise of a data length DL_j^{st} with bandwidth BW_j^{st} so, $j = \{1, 2, \ldots, U\}$ and $1 \le s, t \le |y_j|$. The parallel execution of the subtasks starts once the subtasks attain the allocated resources. Eq. (3) represents the execution time, which is stated as the subtask's ratio of processing capacity and data length of the virtual machines.

$$t_{jl} = DL_j / R_c \tag{3}$$

In eq. (3), R_c indicates the capacity of the machine and DL_j indicates the data length. This time should be indicated as matrix notation, $T = \{t_{jl}\}$ for $j = \{1, 2, \ldots, n\}$ and $l = \{1, 2, \ldots, k\}$, as the number of user request tasks gathered in the cloud is high. Here, the distribution matrix denotes the correlation among the virtual machine and the task as stated in the eq. (4).

$$D_M = \begin{bmatrix} d_{11} & d_{12} & \ldots & d_{1k} \\ d_{21} & d_{22} & \ldots & d_{2k} \\ \vdots & \vdots & \vdots & \vdots \\ d_{n1} & d_{n2} & \ldots & d_{qk} \end{bmatrix} \tag{4}$$

In the matrix, the element d_{jl} posses a value 1 while the resource allocated to the subtask y_j is u_l. While the subtasks collect the resources from the similar data center, a variable determines a value, either 1 or 0, as eq. (5).

$$A_j = \begin{Bmatrix} 1, \text{if} d_{jk} = d_{jp} = 1 \forall m \ne p \\ 0, \text{otherwise} \end{Bmatrix} \tag{5}$$

The variable sets a value 1 if the server offers resources from the similar data center to the subtasks and for the other scenario the value is set as 0.

3.3 Task Scheduling Objectives

Here, the task scheduling technique exploited is Monkey Search algorithm is hybridized with Krill Herd technique for Multi-Objective Task Scheduling.

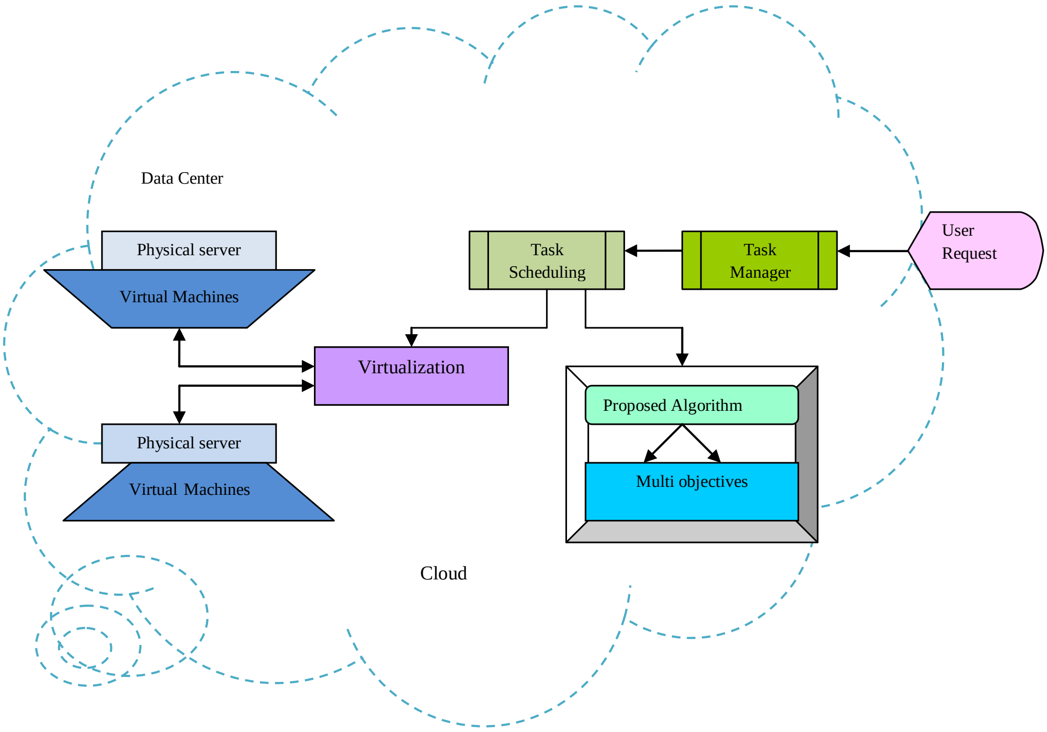


Fig. 3. *Schematic diagrammatic of proposed algorithm*

As demonstrates in fig. 3, through the internet, the user transmits task solicitation to the task scheduler. In the cloud control, the task Manager and task buffer are present it is exploited the tasks to transmit them to the task scheduler. The task buffer permits the tasks request to arrive at the task manager that notices the information with respect to the CPU as well as memory functionality. On the basis of the hybridized proposed algorithm, the submitted tasks arrive at the scheduler, in that the resources are assigned. By the virtual machines, the resources are allocated hence the user tasks needs are fulfilled in the server. With the incorporation of the multiple objectives, it is probable for the server resources to attain the task needs.

3.4 Multi-objective Model Formulation on the Basis of the Epsilon Constraint

Multi-objective optimization contemplates several objective models, which are to be optimized concurrently. In the proposed method, the task scheduling issue exploits a multi-objective optimization technique [14]. Although constructing a fitness model regarding multiple objectives is a hard procedure, the presented method of task scheduling consists of this as a challenge offering an enhanced solution. In the scheduling based optimization issues, the optimization of several targets available uses the cost and processing time. Additionally, the proposed method for scheduling the task incorporates energy utilization as a constraint. In the fitness function of proposed method, the six objectives are considered that incorporates the communication time, execution time, resource utilization energy consumption, communication cost, and execution cost.

In addition, the proposed method incorporates penalty cost functions and epsilon constraint in the computation of the fitness. The augmentation of these models permits the method to carry out the best task scheduling. The variable epsilon integrates into the fitness model analysis increases a solution fitness value, and therefore, it discards those solutions with high fitness. Moreover, the six objectives which are incorporated in the fitness computation are stated with the equivalent formulation.

(a) Execution time: In the task scheduling technique, the execution time of an entire task on the basis of the processing ability and by the distribution matrix element, the execution time of each subtask is multiplied. Eq. (6) represents the execution time needed for the allocation of resource that should be less for feasible task allocation.

$$ET_l = \frac{1}{P_c^{max} * Y} \sum_{j=1}^{n} d_{jl} * t_{jl} \tag{6}$$

In eq. (6), P_c^{max} indicates the utmost processing capacity, d_{jl} indicates the distribution matrix, Y indicates the number of tasks and t_{jl} indicates the execution time for each subtask.

As the virtual machineability is inversely proportional to the execution time, the model is reduced by maximizing its ability. The first objective model is indicated in eq. (7).

$$F_1^{min} = \max(ET_l) \tag{7}$$

(b)Communication Time: The communication time of a task is represented as a second objective model in the allocation of the resource. It is to be reduced in order to carry out efficient scheduling of the task. Maximizing the bandwidth minimizes the communication time, as it is inversely proportional to the bandwidth. Additionally, eq. (8) represents the subtask data length where the conditional variable is incorporated in the calculation of communication time.

$$CT = \frac{1}{BW_{max} * \sum_{j=1}^{n} |y_j|} \sum_{s=1}^{|y_j|-1} \sum_{t=s+1}^{|y_j|} A_j * \left(\frac{DL^{st}}{BW^{st}} \right) \tag{8}$$

In eq. (8), DL^{st} is the data length and BW_{max} indicates the utmost bandwidth. The variable A_j is unity while the server allocates resource from the similar data center to the subtask as well as for other scenario is 0, abandoning the objective is represented in eq. (9).

$$\sum_{\substack{m=1 \\ m \neq p}}^{k} \sum_{p=1}^{k_0} A_j = 1; \forall s, t = \{1, 2, \ldots, |y_j|\} \tag{9}$$

Hence, the objective model is stated as eq. (10).

$$F_{min}^2 = \max(CT) \tag{10}$$

(c)Execution Cost: The cost needed to produce the best scheduling on the basis of the execution time that needs a minimum valued model as eq. (11).

$$EC^Y = \frac{1}{U} \sum_{l=1}^{k} ET_l * R_c \tag{11}$$

In eq. (11), ET_l is the execution time, U indicates the virtual machine, and R_c indicates the resource cost in the server. With a huge number of resources the execution cost can be minimized, as it is inversely proportional. Hence, the third objective is stated in eq. (12).

$$F_{min}^2 = EC^Y \tag{12}$$

(d) Communication Cost: It is computed on the basis of the communication cost and it is stated in eq. (13).

$$M^C = \frac{1}{X} \sum_{j=1}^{q} T_j^C * D_j^T \tag{13}$$

In eq. (13), T_j^C represents the communication time, D_j^T represents the transmitting data per unit time, X represents the number of tasks. The communication cost could be done the minimum in this presented strategy by maximizing the number of tasks. The objective of communication cost is represented as Eq. (14).

$$\underset{min}{F_4} = M^C \tag{14}$$

(e) Energy Consumption: It maximizes with the augmentation in the number of data centers in the cloud. In a virtual machine, the distribution matrix element multiplied using the energy utilized and it is formulated in eq. (15).

$$E^C = \frac{1}{V * X} \sum_{j=1}^{q} \sum_{l=1}^{m} r_{jl} * d_{jl} \tag{15}$$

In eq. (15), d_{jl} indicates the energy devoured by the virtual machine V_l, and r_{jl} indicates an element in the distribution matrix, during the execution of a task X, Eq. (16) indicates the fifth objective model.

$$\underset{min}{F_5} = E^C \tag{16}$$

(f)Resource Utilization: The resource utilization is considered as the last objective and it is calculated on the basis of the subtraction of execution time from the initial objective to offer less value and it is stated in eq. (17).

$$R_U = \frac{\sum_{l=1}^{k} F_l - ET_l}{U * F_l} \quad (17)$$

In eq. (17), U indicates the virtual machine and ET_l indicates the execution time. Contrast to the other objectives, the model inclines to maximize the resource utilization hence to improve the task scheduling procedure that is stated in the eq. (18).

$$F_{min}^{6} = R_U \quad (18)$$

The maximum will be resource utilization when R_U the value is small. The fitness model of the proposed technique is stated in eq. (19). In eq. (19), F_w as well as N_w indicates the penalty cost function, u_w represents the weight assigned to the six objectives. It is exploited in the proposed method to offer an optimum resource allocation by exploiting the epsilon constraint that inclines to increase the fitness as stated in eq. (20).

$$F_N = \frac{1}{6}\sum_{w=1}^{6}\left[\left(v_w * F_w\right) + N_w\right] \quad (19)$$

$$N_w = \begin{cases} 0; \text{if} F_w < \varepsilon_w \\ PC; \text{Otherwise} \end{cases}; 1 \le w \le 6 \quad 20)$$

In eq. (20), ε_w indicates the epsilon constraint and indicates the penalty cost, $PC = 1$. The weights assigned w_h to maximize the solutions fitness values are $\varepsilon_1 = 0.5$, $\varepsilon_2 = 0.2$, $\varepsilon_3 = 0.08$, $\varepsilon_4 = 0.1$, $\varepsilon_5 = 0.05$, and $\varepsilon_6 = 0.5$. While the solution fitness is lesser than the specifically allocated constraint, the variable N_w is 0 that else maximizes the value of the fitness by 1 and hence, it abolishes the fitness having utmost values.

4. Proposed Optimization Algorithm

4.1 Conventional Monkey Search Algorithm (MSA)

The Monkey Search algorithm imitates the procedure in that monkeys climb mountains to arrive at the uppermost point [15]. The climbing technique comprises of three important procedures such as (a) in climb procedure, monkeys explore the local optimum solution widely in a close-up range. (b) In the watch-jump procedure, monkeys search for novel solutions with a value of the functional greater than the present ones. It is represented as an exploitation and strengthening approach. (c) In somersault, the process is used for exploration as well as it averts attainment trapped in a local optimum. In other search domains, monkeys search for novel points. Each monkey attempts to arrive at the uppermost mountaintop in nature that equivalent to the utmost objective model value. The objective model fitness imitates the elevation of the mountaintop when the decision variable vector is represented to comprise the locations of the monkeys. Varying the indication of the objective models permits the method to search the global least amount in place of the global utmost.

(i) Random production of β from the somersault interval is $[a,b]$ whereas *a* and *b* rules the utmost distance that the monkey can somersault.

(ii) Using eq. (21) produce a pivot N, $N_i = (N_1, N_2, ..., N_D)$, Y indicates the position of monkey and N_p indicates the population number.

$$N_i = \frac{1}{N_p - 1}\sum_{l=1}^{N_P}\sum_{i=1}^{N_p} Y_{lj} - Y_{ij} \quad (21)$$

(iii) Get Z i.e., Monkey new position from eq. (22)

$$Z_i = Y_i + \beta\left|N_i - Y_{ij}\right| \quad (22)$$

(iv) Update Y_i with Z_i if possible (within boundary limits) or continue until possible.

4.2 Krill Herd Algorithm (KHA)

This bio-inspired approach [16] imitates the grouping krill individual's behavior. The objective model values communicate to the movements of the krill that indicate the least distances of each individual krill from food as well as from the maximum herd density. The motion of krill engages three most important methods such as the movement persuades using the attendance of other individuals, random diffusion and the foraging enterprises. Additionally, two adaptive genetic operators are exploited such as crossover and mutation approaches. While the predation action is done by predators in nature, like sea birds or penguins, seals, they evade krill individuals out coming in minimizing the density of krill. Subsequently, the krill individuals maximize their density as well as search for food. Consequently, the individual krill goes to the optimal optimum solution as it explores the maximum food and density. The nearer the distance to the maximum food and density, the minimum objective model value is attained. The value of the objective model for each individual krill is theoretical to be an imaginary distance as well as comprises an amalgamation of the distance from food as well as from the maximum density of the krill swarm. The variables of the individuals' for the model are contemplated to be time-dependent positions of an individual krill that are ruled using the three aforesaid characteristics beside with the genetic operator.

4.3 Hybridization of Monkey Search and Krill Herd Algorithm

This algorithm that integrates a few of the schemes and procedures of MSA and KHA to obtain a consistent method for scheduling of a task in cloud computing. Here, the procedure for both the methods comprises diversification/exploration and intensification/exploitation features are described. The diversification/exploration features of MSA are the watch-jump and somersault procedure when for KHA, they are the physical arbitrary diffusion as well as the operators of the genetic algorithm. Conversely, the intensification/exploitation characteristics of MSA are the watch-jump and the climbing procedure, when for KHA; they are the persuade activity and the motion of foraging. Both the methods try to balance among diversification/exploration and intensification/exploitation characteristics. MSA has two exploitation and exploration operators. The watch-jump procedure performs as both exploitation and an exploration operator. The operator of the somersault is a maximum-executing diversification operator, which makes a better employ of the pivot model. As MSA is an exploration-superior method, the exploitation balance is carried to the method by running the climbing procedure twice per iteration. The MSA approach employs a huge number of cycles, which arrives at up to 2000 cycles in a few issues in each process. Maximizing the number of cycles minimizes the computational effectual due to it raises the "Number of Function Evaluations (NFE)".

Although KHA also has dual exploitation and exploration operators, its exploration module is not governing due to the physical arbitrary diffusion is a minimum competent operator of the exploration than the operator of the somersault. Hence, in local minima, the entrapment is high possible in KHA than in MSA. In the KHA, the confine issue can be identified using the two genetic operators such as mutation and crossover. As the movement of foraging is considered as a maximum-executing operator of the exploitation and KHA can be contemplated as an exploitation-superior method.

An equivalent amount of exploitation and exploration operators do not require a balance among exploitation and exploration. The operator performance is considered as a significant factor. Evaluating the operator performance can be performed by substitutes the exploitation or operator of the exploration in one technique with a similar kind of operator in the other technique. To enhance the performance of the technique such that the modified technique performs better than the two conventional techniques, here the optimal performing exploitation and exploration operators from the two techniques are used. The proposed hybrid technique, MSA-KHA, five procedures are used such as physical random diffusion procedure, foraging activity procedure, watch-jump procedure, genetic mutation, as well as crossover procedure and somersault procedure.

In the initialization process, the random generation of the population in that the positions of the hybrid agent both the krill and monkey are produced arbitrarily, $Y_i = (Y_{i1}, Y_{i2}, \ldots Y_{i(D)})$ where $i = 1$ to N_P that indicates the number of hybrids, when D indicates the decision variable vector dimension.

The evaluation of the fitness and sorting are defined as $F_i = f(Y_i)$, whereas F_i indicates for hybrid fitness as well as f indicates the objective model exploited. In the watch-jump procedure, the arbitrary generation of Y_i from $(Y_{ij} - a, Y_{ij} + a)$, whereas a indicates the eyesight of the hybrid (monkey in MSA) that represents the maximum distance the hybrid can observe and $Z_i = (Z_{i1}, Z_{i2}, \ldots Z_{i(D)})$ that are the novel hybrid positions.

If $-f(Z_i) \geq -f(Y_i)$ subsequently update Y_i with Z_i if possible that is within limitations. The Foraging motion depends on the food position and the preceding experience regarding the position. Moreover, computing attractive food, α_i^{food} and the effect of optimal fitness hitherto α_i^{Best}.

$$\alpha_i^{food} = D^{food} \hat{F}_{i,food} \hat{Y}_{i,food} \tag{23}$$

$$\alpha_i^{Best} = \hat{F}_{i,ibest}\hat{Y}_{i,ibest} \tag{24}$$

In eq. (23) D^{food} represents the coefficient of food that minimizes with time and it is computed from eq. (24). In eq. (24), $I_{N_{max}}$ represents the maximum number of iterations and I_N represents the iteration number.

$$D^{food} = 2\left(\frac{1-I_N}{I_{N_{max}}}\right) \tag{25}$$

Eq. (25), indicates calculation of the center of food density and F_{ibest} is the optimal formerly visited position.

$$Y^{food} = \frac{\sum_{i=1}^{N_P}\frac{1}{F_i}Y_i}{\sum_{i=1}^{N_P}\frac{1}{F_i}} \tag{26}$$

.In eq. (26) $\hat{Y}$ and $\hat{F}$ represents unit normalized values attained from this common form.

$$\hat{Y}_{i,j} = \frac{Y_j - Y_i}{\left\|Y_j - Y_i\right\| + \varepsilon} \tag{27}$$

$$\widehat{F}_{i,j} = \frac{F_j - F_i}{F_{worst} - F_{best}} \tag{28}$$

In eq. (27), ε indicates a minute positive number which is augmented to evade singularities. F_{best} indicates the best fitness values and F_{worst} indicates the worst fitness values, correspondingly, of the hybrid agents hitherto. F indicates the hybrid fitness and exploited as K_H variable in the krill herd technique. In eq. (28), the foraging motion is evaluated U_f indicate the foraging speed, FP_i^{old} indicate the final foraging motion and IW_f indicate the inertia weight of the foraging motion in the range [0, 1].

$$FM_i = U_f\alpha_i + IW_f + FP_i^{old} \tag{29}$$

The physical diffusion is an exploration step, which is exploited at high dimensional issue, after that

$$HD_i = HD_{max}\left(1 - I_N \big/ I_{N_{max}}\right)\gamma \tag{30}$$

In eq. (30) γ indicate the arbitrary direction vector and HD_{max} indicate the maximum diffusion speed. Compute the time interval Δt using eq. (31), where A_t indicate constant.

.

$$\Delta t = A_t \sum_{L=1}^{D}\left(U_{B_L} - L_{B_L}\right) \tag{31}$$

The position step is computed using eq. (32) and (33). Here, $\frac{dY_i}{dt}$ indicates the velocity of the proposed hybrid algorithm.

$$\frac{dY_i}{dt} = FM_i + HD_i \tag{32}$$

$$Y(t+\Delta t) = Y_i(t) + \Delta t\frac{dY_i}{dt} \tag{33}$$

・The genetic operators can be implemented by the following steps:

(a) Crossover

$$Y_{i,m} = \begin{Bmatrix} Y_{r,m}, & \text{random} < C_r \\ Y_{i,m}, & \text{O.W} \end{Bmatrix} \tag{34}$$

In eq. (34) $r \in \{1,2,....,i-1,i+1,.....N_P\}$ and C_r indicate the crossover probability

$$C_r = 0.8 + 0.2\hat{K}_{H_{i,best}} \tag{35}$$

$$Y_{i,m} = \begin{Bmatrix} Y_{gbest,m}, + \eta\left(Y_{p,m} - Y_{q,m}\right) & \text{random} < M_P \\ Y_{i,m}, & \text{O.W} \end{Bmatrix} \tag{36}$$

In eq. (36) η indicate a arbitrary number, $p, n \in \{1, 2, ..., i-1, i+1, ..., N_P\}$ and M_P indicate the probability of the mutation and it is represented in eq. (37).

$$M_P = 0.8 + 0.05\hat{F}_{i,best} \tag{37}$$

$$\hat{F}_{i,best} = \frac{(F_i - F_{gd})}{F_{worst} - F_{gd}} \tag{38}$$

In eq. (38), F_{gd} indicates the optimal global fitness of the hybrid hitherto and Y_{gbest} is its position.

5. Results and Discussions

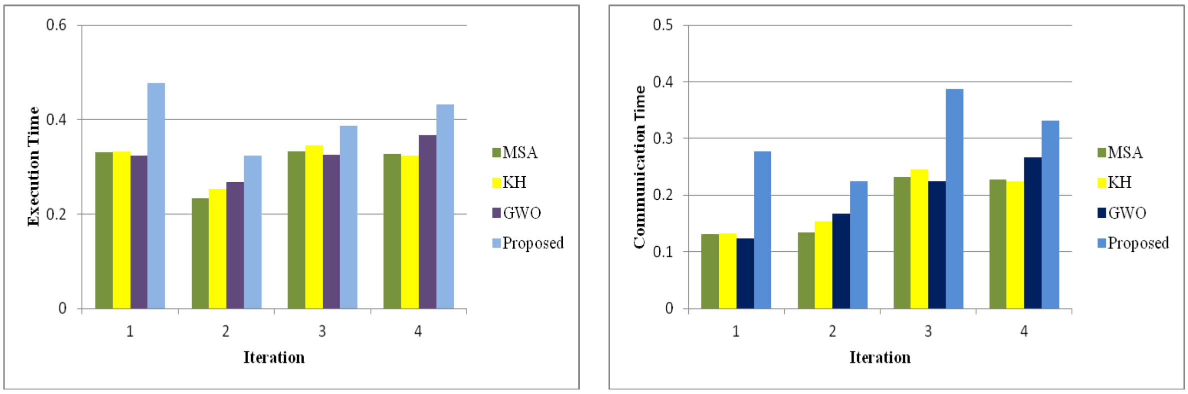
5.1 Experimental Set up

The experimentation is performed in cloud platforms, described as a cloud setup. The description of the cloud setup comprises of 5 physical servers, which possess 14 virtual machines by that the tasks attain the resources. The user request is contemplated to have 5 tasks, each of that is sub-partitioned to perform a total of 17 subtasks. The multiple objectives analysis the proposed method performance by increasing or decreasing the metrics based on the requirement. For the performance evaluation, the six measures are considered like execution time, energy consumption, resource utilization, communication time, communication cost, and execution cost.

5.2 Performance Analysis

Fig. 4 demonstrates the performance analysis of the proposed technique regarding the execution time and communication time. Fig 4 (a) states the performance analysis of the proposed algorithms against conventional algorithms such as MSA, KH and GWO algorithms regarding the execution time. Here, the analysis states that the proposed technique attains the least execution time compared to the conventional techniques.

Fig 4 (b) demonstrates that the analysis of the proposed method with respect to the communication time. For efficient scheduling, it is necessary that the proposed method must utilize minimum communication time.

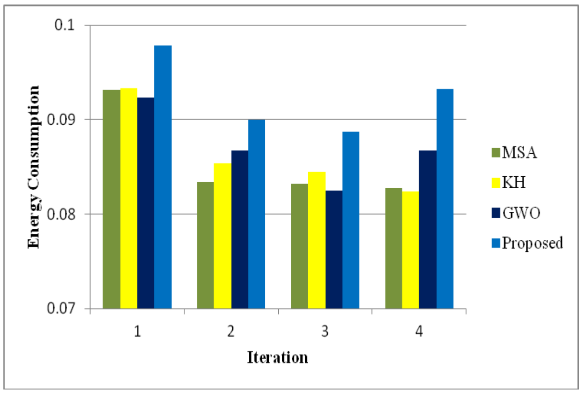


(a) (b)

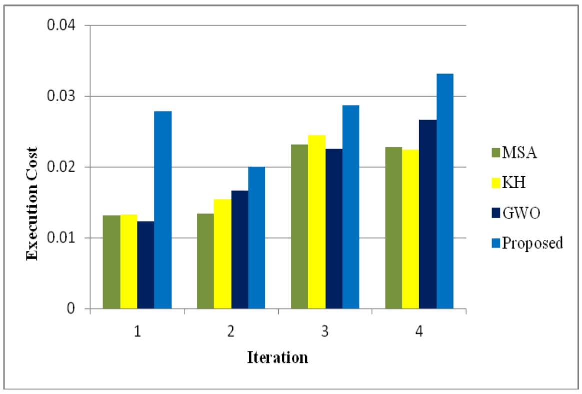
Fig. 4. *Performance analysis of proposed algorithm regarding (a) Execution time (b) Communication Time*

Fig. 5 demonstrates the performance analysis of the proposed method regarding energy consumption and execution cost. Fig 5 (a) states the performance analysis of the proposed algorithms over conventional algorithms such as MSA, KH and GWO algorithms regarding the energy consumption. Here, the analysis states that the proposed enhances its effectiveness by minimizing energy consumption.

Fig 5 (b) demonstrates that the analysis of the proposed method with respect to the execution cost. The performance analysis states that the proposed algorithm has minimized the execution cost than the conventional MSA, KH and GWO algorithms.



(a)

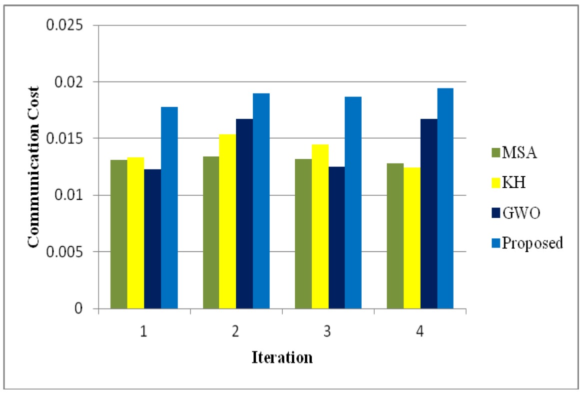


(b)

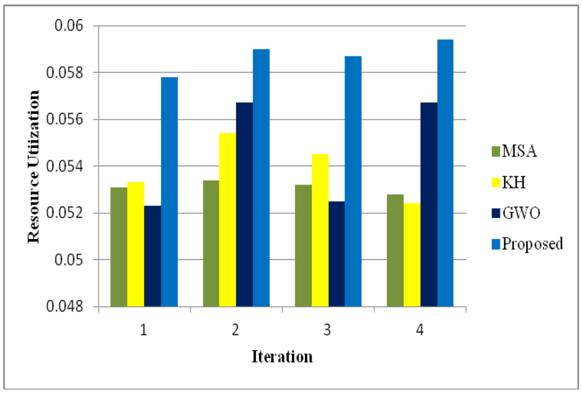
Fig. 5. *Performance analysis of the proposed technique regarding (a) Energy Consumption (b) Execution Cost*

Fig. 6 demonstrates the performance analysis of the proposed method regarding communication cost and resource utilization. Fig 6 (a) states the performance analysis of the proposed algorithms over conventional algorithms such as MSA, KH and GWO algorithms regarding the communication cost.

Fig 6 (b) demonstrates that the analysis of the proposed method with respect to the resource utilization. For enhanced performance, the proposed method must use the resources efficiently, offering maximum resource utilization. The proposed method objective is to choose a reduced value for the utmost utilization. The overall analysis states that the proposed technique has the ability to offer utmost utilization with the minimum value.



(a)



(b)

Fig. 6. *Performance analysis of proposed algorithm regarding (a) Communication Cost (b) Resource Utilization*

6. Conclusion

In cloud computing, task scheduling was considered as one of the main issue that minimizes the performance of the system. To enhance the performance of the system, there was necessitate of a competent task-scheduling algorithm. Conventional task-scheduling techniques spotlight on CPU memory, task-resource requirements, execution time and execution cost. Nevertheless, these do not consider energy consumption. In this paper, a hybridization method, named Monkey Search with Krill Herd algorithm for multi-objective task scheduling is presented. The proposed method follows the multiple objectives, namely execution time, communication cost, resource utilization, communication time, execution cost, and, energy consumption on the basis of the penalty cost technique and epsilon constraint. The resources are allocated to the tasks by exploiting the hybridization of algorithmic process. The multi-objective evaluation evaluates the proposed method performance with three conventional techniques such as MSA, KH, and GWO. The performance analysis outcomes attained states that the proposed approach can offer enhanced as well as stable performance for task scheduling, resolving the multiobjective optimization issues.

References

[1] Mohit Kumar, S. C. Sharma, Anubhav Goel, S. P. Singh, "A comprehensive survey for scheduling techniques in cloud computing" Journal of Network and Computer Applications, Volume 143, 1 October 2019, Pages 1-33.

[2] Luiz F. Bittencourt, Alfredo Goldman, Edmundo R. M. Madeira, Nelson L. S. da Fonseca, Rizos Sakellariou, "Scheduling in distributed systems: A cloud computing perspective", Computer Science Review, Volume 30, November 2018, Pages 31-54.

[3] P. Zhang and M. Zhou, "Dynamic Cloud Task Scheduling Based on a Two-Stage Strategy," IEEE Transactions on Automation Science and Engineering, vol. 15, no. 2, pp. 772-783, April 2018.

[4] S. Guo, J. Liu, Y. Yang, B. Xiao and Z. Li, "Energy-Efficient Dynamic Computation Offloading and Cooperative Task Scheduling in Mobile Cloud Computing," IEEE Transactions on Mobile Computing, vol. 18, no. 2, pp. 319-333, 1 Feb. 2019.

[5] L. Liu, Q. Fan and R. Buyya, "A Deadline-Constrained Multi-Objective Task Scheduling Algorithm in Mobile Cloud Environments," IEEE Access, vol. 6, pp. 52982-52996, 2018.

[6] M. Chen, Y. Hao, C. Lai, D. Wu, Y. Li and K. Hwang, "Opportunistic Task Scheduling over Co-Located Clouds in Mobile Environment," IEEE Transactions on Services Computing, vol. 11, no. 3, pp. 549-561, 1 May-June 2018.

[7] P. Zhang and M. Zhou, "Dynamic Cloud Task Scheduling Based on a Two-Stage Strategy," in IEEE Transactions on Automation Science and Engineering, vol. 15, no. 2, pp. 772-783, April 2018.

[8] Gan, G.N.; Huang, T.L.; Gao, S. Genetic Simulated Annealing Algorithm for Task Scheduling based on Cloud Computing Environment. In Proceedings of the 2010 International Conference on Intelligent Computing and Integrated Systems, Guilin, China, 22–24 October 2010; pp. 60–63.

[9] Babukarthik, R.G.; Raju, R.; Dhavachelvan, P. Hybrid Algorithm for Job Scheduling: Combining the Benefits of ACO and Cuckoo Search. Adv. Intell. Syst. Comput. 2013, 177, 479–490.

[10] Tong, Z.; Chen, H.; Chen, M.; Mei, J.; Liu, H. A hybrid biogeography-based optimization algorithm for task scheduling in cloud computing. J. Comput. Eng. Sci. 2018, 40, 765–772.

[11] J. Li, M. Qiu, J. Niu, Z. Zong, W. Gao, and X. Qin, “Feedback dynamic algorithms for preemptable job scheduling in cloud systems,” in Proc. WI/IAT, Aug. 2010, pp. 561–564.

[12] S. Devipriya and C. Ramesh, “Improved Max-min heuristic model for task scheduling in cloud,” in Proc. ICGCE, Chennal, India, Dec. 2013, pp. 883–888.

[13] E. Santos-Neto, W. Cirne, F. Brasileiro, and A. Lima, “Exploiting replication and data reuse to efficiently schedule data-intensive applications on grids,” in Job Scheduling Strategies for Parallel Processing (Lecture Notes in Computer Science), Berlin, Germany: Springer-Verlag, vol. 3277, Jun. 2005, pp. 210–232.

[14] H. He, G. Xu, S. Pang, and Z. Zhao, “AMTS: Adaptive multi-objective task scheduling strategy in cloud computing,” China Commun., Vol. 13, no. 4, pp. 162–71, Apr. 2016.

[15] M. Zein, A. E. Hassanien, A. Badr and T. Kim, "Human Activity Classification Approach on Smartphone Using Monkey Search Algorithm," 2015 Seventh International Conference on Advanced Communication and Networking (ACN), Kota Kinabalu, 2015, pp. 84-88.

[16] E. Fattahi, M. Bidar and H. R. Kanan, "Fuzzy Krill Herd optimization algorithm," 2014 First International Conference on Networks & Soft Computing (ICNSC2014), Guntur, 2014, pp. 423-426.

[17] S Vinusha and J.S. Abinaya,"Performance analysis of the Adaptive Cuckoo Search Rate Optimization Scheme for the Congestion Control in the WSN",Journal of Networking and Communication Systems (JNACS),Volume 1, Issue 1, October 2018.

[18] Amit Kelotra and Prateek Pandey,"Energy-aware Cluster Head Selection in WSN using HPSOCS Algorithm", Journal of Networking and Communication Systems (JNACS),Volume 2, Issue 2, October 2018.

[19] Vijayakumar Polepally, K Shahu Chatrapati,"DEGSA-VMM: Dragonfly-based exponential gravitational search algorithm to VMM strategy for load balancing in cloud computing";Kybernetes, vol.67, no.6;pp.1138-1157;2018.