

## Evolutionary Algorithm Based Task Scheduling in IoT Enabled Cloud Environment

R. Joshua Samuel Raj<sup>1</sup>, M. Varalatchoumy<sup>2</sup>, V. L. Helen Josephine<sup>3</sup>, A. Jegatheesan<sup>4</sup>, Seifedine Kadry<sup>5</sup>, Maytham N. Meqdad<sup>6</sup> and Yunyoung Nam<sup>7,\*</sup>

<sup>1</sup>Department of Information Science & Engineering, CMR Institute of Technology, Bengaluru, 560037, India

<sup>2</sup>Department of Computer Science & Engineering, Cambridge Institute of Technology, Bengaluru, 560036, India

<sup>3</sup>Department of Computer Applications, CMR Institute of Technology, Bengaluru, 560037, India

<sup>4</sup>Department of Computer Science & Engineering, Swarnandhra College of Engineering and Technology, Narasapur, 534280, India

<sup>5</sup>Department of Applied Data Science, Noroff University College, Kristiansand, 4612, Norway

<sup>6</sup>Department of Computer Techniques Engineering, Al-Mustaqbal University College, Hillah, 51001, Iraq

<sup>7</sup>Department of Computer Science and Engineering, Soonchunhyang University, 31538, Korea

\*Corresponding Author: Yunyoung Nam. Email: ynam@sch.ac.kr

Received: 16 July 2021; Accepted: 17 August 2021

**Abstract:** Internet of Things (IoT) is transforming the technical setting of conventional systems and finds applicability in smart cities, smart health-care, smart industry, etc. In addition, the application areas relating to the IoT enabled models are resource-limited and necessitate crisp responses, low latencies, and high bandwidth, which are beyond their abilities. Cloud computing (CC) is treated as a resource-rich solution to the above mentioned challenges. But the intrinsic high latency of CC makes it nonviable. The longer latency degrades the outcome of IoT based smart systems. CC is an emergent dispersed, inexpensive computing pattern with massive assembly of heterogeneous autonomous systems. The effective use of task scheduling minimizes the energy utilization of the cloud infrastructure and rises the income of service providers by the minimization of the processing time of the user job. With this motivation, this paper presents an intelligent Chaotic Artificial Immune Optimization Algorithm for Task Scheduling (CAIOA-RS) in IoT enabled cloud environment. The proposed CAIOA-RS algorithm solves the issue of resource allocation in the IoT enabled cloud environment. It also satisfies the makespan by carrying out the optimum task scheduling process with the distinct strategies of incoming tasks. The design of CAIOA-RS technique incorporates the concept of chaotic maps into the conventional AIOA to enhance its performance. A series of experiments were carried out on the CloudSim platform. The simulation results demonstrate that the CAIOA-RS technique indicates that the proposed model outperforms the original version, as well as other heuristics and metaheuristics.

**Keywords:** Internet of things; cloud computing; task scheduling; meta-heuristics; resource allocation



This work is licensed under a Creative Commons Attribution 4.0 International License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

## 1 Introduction

Internet of Things (IoT) [1] permits to achievement of useful and actionable information, multiple tasks, and glean. It also experienced rapid development of objects driven by a gadget of consumers for accessing multimedia files to achieve various tasks that work independently/individually [2]. The fundamental framework of IoT allows objects to connect and collaborate with Internet in a smart manner without human interventions (viz., smart objects). Smart objects can interact, create, and sense ubiquitous environments. Within this environment, data can be used and collected for supporting further decision makings like manufacturing, emergency handling applications, industrial automation, healthcare, logistics, and scheduling in transportation. Moreover, IoT could be viewed as a kind of largescale distributed system like in Cloud Computing (CC) wherein such objects have a higher amount of heterogeneity regarding software and hardware.

In spite of the present connectivity solution, wide-ranging efforts are required in the scheme of a novel framework IoT reference method. As a result, novel method provision is required for resource allocation (RA) and communication to effectively assist huge IoT placement based on devices and services [3]. The novel IoT reference framework needs to scale economically and operationally with the extension of IoT components and provides smart functionality for autonomous reasoning amongst objects. Order to Implement IoT system in the market would provide greater opportunity and allow various services and devices to interact, stream, and connect in small business/single homes. The IoT could be seen as largescale distributed systems in which the modules have a higher amount of heterogeneity regarding hardware and software based on devices and services. The Quality of Services (QoS) for the user accessing IoT supporting with CC is exponentially increasing.

The most significant challenge in the CC domain is resource scheduling (RS). While carrying out RS, at least fulfilling levels of QoS must be preserved with the help of appropriate hardware algorithms and infrastructure [4]. The components of cloud framework, generally denoted as the broker in recent survey, is accountable to map requested end users task to the accessible virtualized hardware i.e., generally executed by means of virtual machine (VM). The broker carries out mapping by implementing the scheduling process. With the development of several available resources and submitted tasks, it has become highly complex for mapping tasks to the proper VMs for implementation [5]. When an improper scheduling process is utilized, few VMs might be under-utilized/over-utilized, and the inference of this scenario leads to degrading the overall performances of the cloud system. The RS problems belonging to the set of NP (non-deterministic polynomial time) hard optimization problems. It must be pointed out that in the modern computer science literature, the terms cloudlet, and task scheduling are also adapted for the procedure of mapping submitted end-user tasks to accessible VMs.

Several IoT application involves various devices attempt to exploit the underutilized assets. The ability of heterogeneous devices and multi-servers for providing different services to the income requests are distributed in a different way. Therefore, the exploitation of heterogeneous server devices and systems should assist in scheduling multiple tasks and share resources for meeting the user QoS needs. Such exploitations are based on the kind of priority of task scheduling, allocation policy selection, and flexibility. There are various kinds of policy selection which have severe limitations to be fulfilled with the potential usage of system resources and applications [6]. Also, almost all the scheduling methods in real world system could not offer an optimum scheduling solution on pre-emptive scheduling when fulfilling their RA needs like tardiness, minimize Makespan, maximum throughput, and utilization ratio in CC when getting

huge requests. Ineffective scheduling could lead to lower throughput, excessive utilization, and undesired long delay of IoT resources. The major issue is how to allocate tasks in an effective manner based on minimizing Makespan when fulfilling deadlines and another limitation [7]. Thus, it is vital to suggest a proper scheduling process for facilitating optimum task scheduling to minimize the Makespan of potential use of IoT RA between different heterogeneous systems of devices and servers.

Several techniques and algorithms for RA in CC platform are now accessible. As, in few schemes, traditional (deterministic) algorithm is utilized. But, traditional optimization methods aren't effective because deterministic algorithms cannot generate fulfilling, or optimum, or best solutions within a moderate computation time for NP-hard problems. Because of the exponential number of potential solutions and search space complexity, traditional methods cannot estimate all possible solutions from the search domain in polynomial time [8]. While addressing NP-hard processes like CC-RS, rather than utilizing traditional optimization methods, approaches which estimate only possible portions of the search space (not the entire search domain) with the help of a smart system while selecting that solution to estimate the following, must be used. The effective way for tackling the RS challenge is the execution of metaheuristics and heuristics based methods which don't assurance detecting optimum solutions, but practically, they have proven, that it is capable of generating fulfilling solutions within a polynomial time.

This paper presents an intelligent Chaotic Artificial Immune Optimization Algorithm for Task Scheduling (CAIOA-RS) in IoT enabled cloud environment. The proposed CAIOA-RS algorithm resolves the problem of resource allocation in the IoT enabled cloud environment. It also fulfills the makespan by carrying out the optimal task scheduling process with the different policies of incoming tasks. The design of CAIOA-RS technique integrates the concept of chaotic maps into the traditional AIOA to improve its performance. An extensive set of simulations take place on the CloudSim platform and examined the outcomes interms of different performance measures.

## 2 Literature Review

This section performs an extensive survey of different task scheduling algorithms for the IoT enabled CC environment. In Praveenchandar et al. [9], an enhanced task scheduling and an optimum power minimization method are projected for an effective dynamic RA procedure. With the prediction and dynamic resource table upgrading mechanism, efficacy of RA based on response time and task completion is realized. This architecture carries effective results based on power reduction as it decreases the power utilization in data centres. The presented method provides precise values for upgrading resource table. An effective RA is attained using an enhanced task scheduling method and decreased power utilization method.

In Rjoub et al. [10], a trust aware scheduling solution named BigTrustScheduling is projected which contains 3 phases: trust-aware scheduling, VMs trust level computation, and tasks priority level determination. An experiment was carried out on real Hadoop cluster environments with real time dataset gathered and Bit brains task. Hasan et al. [11] proposed a task scheduling approach in terms of CPSO method for solving the problems of resource management and RA in heterogeneous and homogeneous IoT CC. The aim is to fulfill the Makespan by carrying out optimum task scheduling when taking into account various policies of incoming tasks. Abd Elaziz et al. [12] proposed another task scheduling technique for IoT requests in a cloud fog platform. Abdelmoneem et al. [13] developed an effective IoT framework, mobility aware allocation, and scheduling techniques for health care sector. The presented method assists the movement of the patient using an adoptive RSS based handoff system. The presented framework enables dynamic

sharing of healthcare tasks between computation nodes either fog devices/cloud devices using execution of a MobMBAR approach.

Ma et al. [14] proposed a deadline and cost aware scheduling approach which decreases the implementation cost of a task under the deadline constraint in the IaaS method. Taking into account the VM acquisition delay and performance variation, firstly they split tasks into various types based on the topological framework thus no dependencies are present among tasks on a similar level. The 3 strings are utilized for coding the genes in the presented method to reflect the resilient and heterogeneous features of cloud environment. Later, HEFT is utilized for generating individuals with minimal cost and completion time. In Basu et al. [15], a cognitive or intelligent method of bioinspired method is utilized for finding an optimum task scheduling for IoT application.

Al-Turjman et al. [16] proposed a cooperative RS in energy constrain application for fault-tolerant and reliable performances. They presented a task scheduling approach in terms of strong CPSO and FIPS approaches for solving the problems of RA in heterogeneous and homogeneous cloud based IoT applications. Xu et al. [17] formulated related task scheduling problems to constrained optimization problems in cloud fog platforms. An LBP-ACS approach is presented for tackling these problems. In this method, an approach of task scheduling is considered the finished deadline and the priority of a task. For handling the sensitivity of task delay, the laxity based priority method is adapted for constructing task scheduling sequences with a moderate priority. While, for minimizing the overall energy utilization, the constrained optimization approach depends on ACO method is utilized for obtaining the accurate optimum scheduling system around the world.

### 3 The Proposed Task Scheduling Technique

Fig. 1 demonstrates the overall working process of proposed method. For solving the issue of resource optimization within the cloud structure, a standard CC technique. This technique utilized in this investigation is same as the approach. The cloud scheme has several geographically distributed data centers (DC) related to internet. Initially, the DC has several compute and store servers and other needed resources. The server in all DCs are linked utilizing consistent and high bandwidth communication networks. So, in this presented technique, communication delay was regarded as negligible. During the presented technique, user accesses the cloud resource with user interfaces. In cloud manager accomplishes every resource utilizing distinct cloud components, for instance, the network modules, image modules, pricing modules, and certification modules, and so on. The user task is allocated amongst distinct DCs obtainable in cloud framework. All DCs separates the user tasks as to various subtasks and allocates them amongst the accessible Processing Elements (PE's) in respective DCs. The presented task scheduling scheme in the structure was responsible for effectively shared of user tasks as to distinct existing servers with objective of optimizing energy utilization and profit of service providers.

#### 3.1 Problem Formulation

In this technique, the cloud applications are regarded as gathered of user jobs which perform difficult computing tasks with utilizing cloud resources. Assume that  $UserJob = (U_1, U_2, U_3 \dots U_N)$  implies the batch of user applications attained at specific time. All the User jobs ( $U_i$ ) has signified as duplet  $\langle a_i, d_i \rangle$ . Where  $a_j$  refers the arrival time of user job ( $U_i$ ) and  $d_i$  implies the aim of user jobs ( $U_i$ ). When the job cannot finish before goal, afterward it can be regarded as failed job and over queued for fresh scheduling. In the scheduling method, a user job

is allocated to the obtainable DCs (DC's) ( $D_1, D_2, D_3 \dots D_M$ ), where  $N \leq M$ . All DCs ( $D_i$ ) is connected to duplet  $\langle C_i, m_j \rangle$ .  $C_i$ , the price per unit time charged as DC for executing user job,  $m_j$  defines the amount of accessible Processing Elements (PEs) for executing userjobs [18]. All DCs have a group of PEs  $\{PE_1, PE_2 \dots PE_m\}$  for computing assign user jobs. All PEs are connected with duplet  $\langle s, p \rangle$ . ' $s$ ' and ' $p$ ' refers the implementation speed and power utilization of every PE correspondingly. All User Jobs are projected as Directed Acyclic graphs (DAG) represented as  $(V, E)$ . The group of nodes  $V = \{T_1 \dots T_n\}$  signifies the tasks, and the group of arcs illustrates the control or data dependency amongst tasks. The arc is in the procedure of  $\langle T_i, T_j \rangle \in E$ , where  $T_i$  represents the parent task and  $T_j$  implies the child's task.  $T_j$  utilized the data formed by  $T_i$ . The child task could not be applied until every parent task is finished. To provide task graph, the task with no parent is regarded as entry tasks, and the task without some child is named as exit tasks. In this research, only one entry and one exit task node are assumed as zero implementation time. All the vertex E in DAG was connected to value,  $\langle l \rangle$  implies the length of task in Million Instruction (MI). Optimum scheduling of user jobs for obtainable PEs in cloud from various DC is an important objective of this work. Every PE is regarded as homogeneous, independent, and parallel. In scheduling, procedure is regarded as non-preemptive.

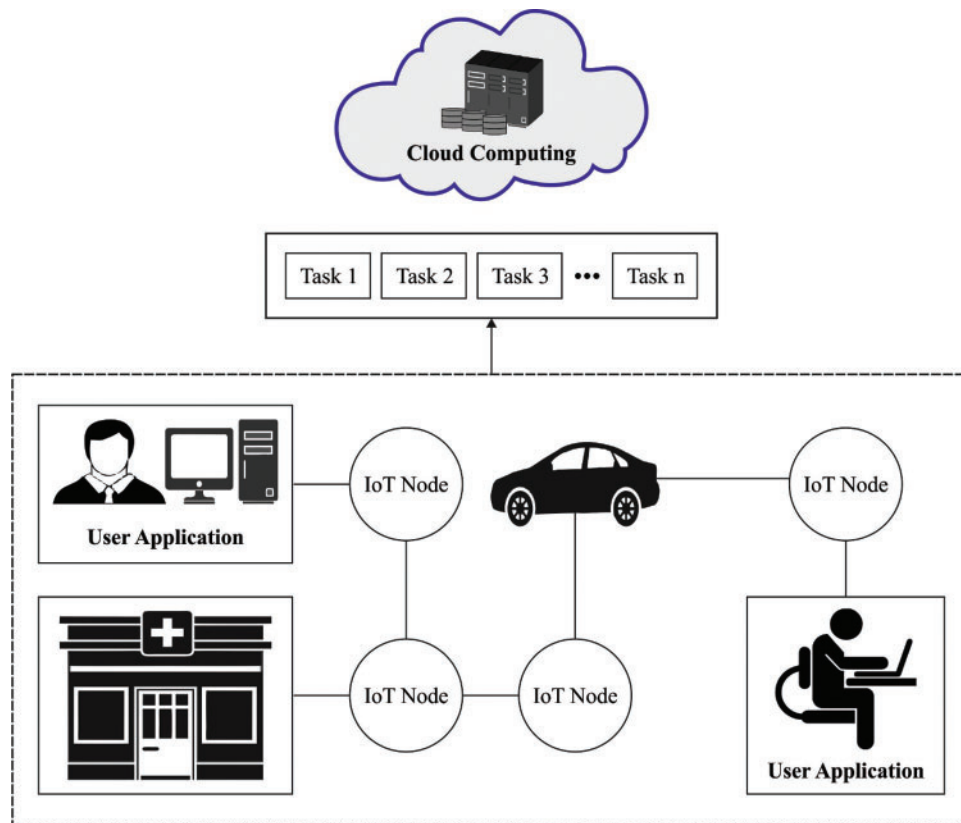


Figure 1: Overall process of proposed method

### 3.2 Objective Function

Supposing user jobs  $U_i$  is allocated to DC ( $D_j$ ).  $T_k$  represents the group of tasks of user jobs ( $U_i$ ) is allocated to PE ( $P_j$ ). When the time need implementing  $T_k$  utilizing  $P_j$  is referred as  $G_j$ . The ending time of  $T_j$  is written as:

$$Finish(T_k) = start(T_k) + \Gamma_j \quad (1)$$

Thus, the entire time essential for completing the user job as  $D_j$  is referred to as *Makespan<sub>j</sub>* and determined as:

$$Makespan_j = \max \{Finish (T_k)\} \quad (2)$$

where  $T_{(k=1..n)}$  the tasks are allocated for  $D_j$

The Energy utilization for computing the user jobs ( $U_i$ ) by DC  $D_j$  has computed as:

$$E_i = \sum_{k=1}^n (\Gamma_k \times p_k) \quad (3)$$

where  $p_k$  refers the power utilized per unit time by PEs ( $P_j$ ) for processing provided task ( $T_k$ ).

The cost for processing the user job as DC  $D_j$  has computed as:

$$C_j = c_j \times Makespan_j \quad (4)$$

where  $c_j$  implies the price per unit time charged by DC  $D_j$  for executing user jobs.

The consumption ( $U_j$ ) of DCs ( $D_j$ ) was computed as:

$$U_j = \frac{Makespan_j}{\max \{Makespan_k\}, k = 1 \dots M} \quad (5)$$

The main functions of this presented technique are written as:

$$\text{Minimize } Makespan_j \quad j = 1..M \quad (6)$$

$$\text{Minimize } E_j \quad j = 1..M \quad (7)$$

$$\text{Minimize } \left\{ \sum_{j=1}^M C_j \right\} \quad (8)$$

$$\text{Maximize } U_j \quad j = 1..M \quad (9)$$

Subjected to:

- The user job necessity end before deadline ( $d_i$ )
- All user jobs are assigned to only one DC.
- The amount of user jobs are lesser than the amount of existing DC at a specific time.

### 3.3 Design of CAIOA for Task Scheduling

In order to flexible task scheduling issues in cloud model, it examined the permutation based encoded technique, i.e., the assignment of integer number to every job and the order of all the jobs signifies the processing order. For instance, there are 5 jobs that exist processing with 3 stages, and in every stage, there are many parallel devices. The processing time to all tasks on every device is 3, 2, and 2 parallel devices in 3 stages correspondingly. To provide one solution  $\{1, 2, 3, 4, 5\}$ , it implies that in the initial stage in cloud modal, the processing order of jobs is  $J_1, J_2, J_3, J_4,$  and  $J_5$ .

Once the scheduling order was defined as subsequent work is for assigning the candidate devices to process it [19]. During this case, it can employ the easy but effectual manner for assigning the device to process the task, and it can be provided as:

- (1) At the beginning stage, all tasks are allocated to device with minimal completion time, i.e., the device existing as initial as feasible.
- (2) Once the task is finished their work is in the present stage, allocate it to subsequent stage directly and elect the devices with simplest idle time.

### 3.4 Solution Representation

AIOA operator includes: operator calculation of individual level, affinity evaluation operator, immune selection operator, incentives meter calculate son, operator, population suppression cloning, mutation operator, and Refresh operator clone (individual multiplication) operator. Using GA and related intelligent optimization methods, artificial immune method to depend on evolution optimization procedure operator to attain. As the encoding method might be a discrete, different, real number coding techniques in the operator would be distinct.

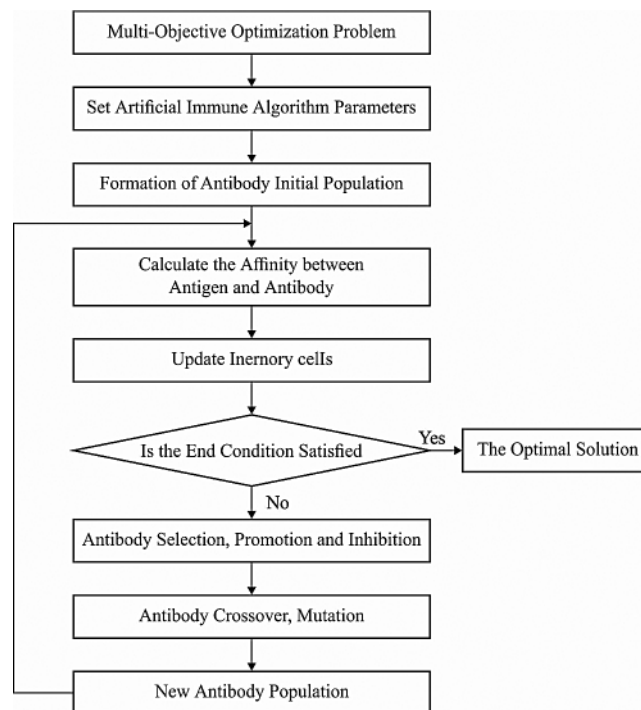
**Antibody Density Operator** (den (x):  $S \rightarrow [0, 1]$ ) classify different types of higher antibodies density, antibodies population quality is more related to the population. Generally, it is determined by

$$\text{den} (A_i) = \frac{1}{m} \sum_{i=1}^m \text{aff}(A_i, B_j), \quad (10)$$

Whereas  $A_i$  denotes the initial  $i$ -species antibodies;  $m$  represents the population size;  $\text{aff} (A_i, A_j)$  indicates the antibody  $i$  &  $j$  in the affinity antibodies. Calculation of antibodies density is an essential condition for antibody determination of antibody affinity,  $\text{aff} (u, v) : S \times S \rightarrow R$ , whereas  $S$  represents the possible solution area. Immunity often referred as the affinity to antigen and antibody affinity, indeed, among the antibody and affinity antibody also present the idea of 2 antibodies that denotes the degree of comparison among individuals. Antibody and antibody affinity mostly depends on antigen affinity evaluation, Euclidean distance is evaluated, information entropy, Hamming distance evaluation, and evaluation of antibody technique. Every individual antibody contain affinity for *anügen*  $\text{aff} (x)$ , hence antibodies - antibody affinity using antigen-antibody affinity to determine [20].

$$\text{aff} (A_i, B_j) = \begin{cases} 1, & \text{aff}(A_i) = \text{aff}(B_j), \\ (1 + (\text{aff}(A_i) - \text{aff}(B_j))^2)^{-1}, & \\ \text{aff} (A_i) \neq \text{aff}(B_j). & \end{cases} \quad (11)$$

The antibody is generally the evaluation of excitation levels that could be utilized antibodies density and affinity in calculating the outcomes of numerical operation. Antibodies incentive amounts (sim  $(x): S \rightarrow R$ ) represent the quality of last antibodies. Calculation outcomes should be taken into account and the amount of antibody density and affinity, generally a huge amount of affinity, lower density of antibody would be higher incentive amounts. Affinity problems and calculation specially interrelated to, for dissimilar optimization issues, must be the real problems in understanding the idea, based on the features of the problems determination of affinity calculation function. Function optimization problems could generally function value of simple processing (like take the reciprocal of opposite number, and so on.) as a pro and calculation, and application of combinational optimization problem, or highly complicated optimization issues, they require certain problems and issues analyses. Affinity calculation operator is generally a function  $aff(x): S \rightarrow R$ , whereas  $S$  denotes the possible solution area,  $R$  indicates real number region. The input is a function of individual antibody (possible solution), the output represents the affinity of calculation outcomes. Fig. 2 illustrates the flowchart of AIOA.



**Figure 2:** Flowchart of AIOA

In stocks to refresh operator  $(Td: S \rightarrow S)$ , the population of antibodies is lower incentive for refreshing, eliminate from the antibodies population arbitrarily created antibody, and the novel antibody for replacing is helpful to maintain the variety of antibodies, and examine novel possible solution space region.

Incentives in the higher antibody set of antibodies using improved quality individual are highly possible to be chosen for selecting clone operator, highly useful in the search space, the search region for staring local search.  $(Ts: S \rightarrow S)$  for determining the amount of incentives on the basis of antibody election.



( $Tm: S \rightarrow S$ ) on clone, operator attained the outcomes of *anu* body to the clone. Line mutation process for creating affinity mutation, to attain local search. It is an artificial immune process that has the possibility to create efficiency of the algorithm greatly, novel antibodies, and the significance of region search operator. Also, it is interrelated to the real coding algorithms, discrete coding algorithms, and coding algorithms for various mutation operators.

Inhibit in the cloning procedure, the clone part of mutation operator and operator operation antibody using temporary antibodies population attained afterward the creation of a general set of  $X_i$  (antibodies cloning  $A_i$  mutation result using the structure, prevent the actions would save this clone set the maximum affinity antibodies, inhibit of other antibodies. ( $Tr: S \rightarrow S$ ) mutation on via the body to the cloning. Later choose the line, inhibit of lower affinity antibody, and the antibodies retain higher affinity antibody to novel population.

( $Tc: S \rightarrow S$ ), immune selection operator is chosen for replicating the individual antibody.

One of the meta-heuristic techniques is arbitrariness parameters. This parameter is utilizing probability distributions, generally Gaussian or uniform distribution. In recent times, the chaos model has been utilized for enhancing these parameters. It is similar features as arbitrariness, with optimum dynamical as well as statistical features. The chaos is named a phenomenon. Some changes of primary state of chaos can lead to nonlinear modify to the future performance. 3 important properties are explaining the chaos; (i) quasi-stochastic, (ii) ergodicity, and (iii) sensitivity to primary criteria. Mixing of these features guarantees the diversity of created solution. Therefore, this diversity is sufficient for reaching all modes of multi-modal issues. In quasi-stochastic has determined as capability for replacing arbitrary variable with value of chaotic map. In ergodic properties refer to capability of chaotic variables for searching non-repeatedly every state within a particular range. Eventually, the sensitivity to primary state property is determined as some tiny modification, at the beginning initial points, can leads to distinct performance. Relating to every property is considerably boosted the efficiency of meta-heuristic optimization techniques. During this case, ten distinguished non-invertible maps with various features are utilized. During this work, a primary point to every adopted chaotic map has primarily been set to 0.7. This value is similar value utilized to similar ten chaotic maps.

At the CAIOA technique, the arbitrary established is substituted by chaotic initialization, i.e., the initialization of  $i^{\text{th}}$  antibody is completed in Eq. (12).

$$X_{id} = X_{\min,d} + (X_{\max,d} - X_{\min,d}) \cdot \text{chaos}(i + d) \quad d = 1, 2, \dots, n \quad (12)$$

where  $X_{id}$  implies the  $d^{\text{th}}$  dimensional of place of  $i^{\text{th}}$  antibody,  $n$  represents the amount of decision variables,  $X_{\max,d}$  and  $X_{\min,d}$  signifies the upper and lower bounds of  $d^{\text{th}}$  decision variables, and  $\text{chaos}(\cdot)$  refers the standard chaotic map functions.

#### 4 Performance Validation

This section examines the performance of the proposed CAIOA-RS technique with existing techniques. The proposed CAIOA-RS technique is validated on the synthetic datasets and the results are investigated under different numbers/sizes of tasks. There are different types of tasks such as extra-large (800–1000 tasks), large (600–700 tasks), medium (400–500 tasks), and small (100–200 tasks). A comprehensive comparison analysis of the CAIOA-RS technique with existing techniques interms of average turnaround time (ATAT) and average response time (ART) is in Tab. 1.

**Table 1:** Analysis of CAIOA-RS method with other scheduling methods

Models	ATAT (ms)	ART (ms)
RR	0.0420	0.0305
FCFS	0.0419	0.0308
SJF	0.0416	0.0302
GA	0.0266	0.0203
IPSO	0.0577	0.0492
Firefly	0.0555	0.0489
IPSO-firefly algorithm	0.0221	0.0152
FIMPSO	0.0211	0.0136
CAIOA-RS	0.0197	0.0112

Fig. 3 investigates the ATAT analysis of the CAIOA-RS technique with existing techniques. The results demonstrated that the IPSO and Firefly algorithms have demonstrated ineffective outcomes with the ATAT of 0.0577 and 0.055 ms, respectively. In line with, the RR, FCFS, and SJF techniques have obtained moderate outcomes with the ATAT of 0.042, 0.0419, and 0.0416 ms, respectively. Likewise, the GA, IPSO-Firefly algorithm, and FIMPSO techniques have showcased competitive performance with the ATAT of 0.0266, 0.0221, and 0.0211 ms, respectively. However, the proposed CAIOA-RS technique has showcased effectual outcomes with the ATAT of 0.0197 ms.

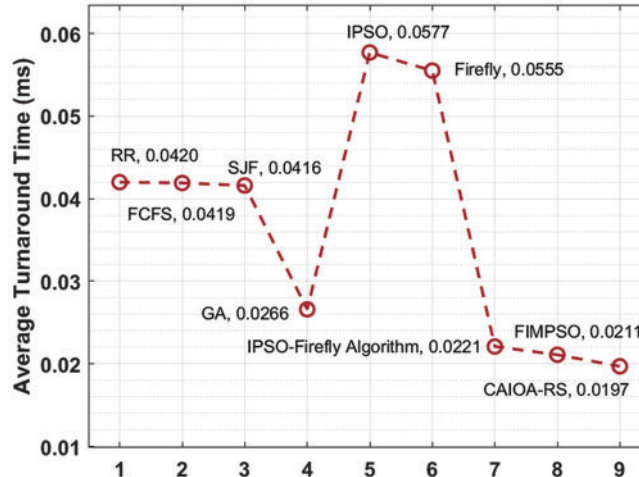
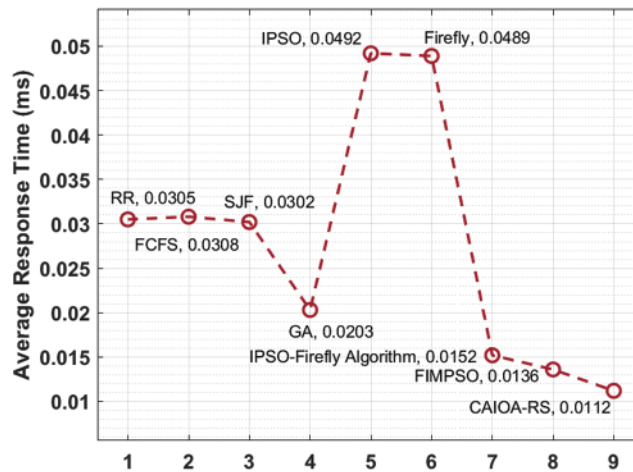
**Figure 3:** Result analysis of CAIOA-RS method interms of ATAT

Fig. 4 examines the ART analysis of the CAIOA-RS approach with recent methods. The outcomes exhibited that the IPSO and Firefly manners have portrayed ineffective results with the ART of 0.0492 and 0.0489 ms, correspondingly. Also, the FCFS, RR, and SJF algorithms have reached moderate outcomes with the ART of 0.0308, 0.0305, and 0.0302 ms, correspondingly. In line with, the GA, IPSO-Firefly algorithm, and FIMPSO manners have depicted competitive performance with the ART of 0.0203, 0.0152, and 0.0136 ms, correspondingly. Finally, the projected CAIOA-RS methodology has outperformed effectual results with the ART of 0.0112 ms.



**Figure 4:** Result analysis of CAIOA-RS method interms of ART

Tab. 2 demonstrates the comparative analysis of CAIOA-RS method with existing techniques interms of CPU utilization, memory utilization, makespan, and average throughput. A CPU utilization of the CAIOA-RS technique with existing techniques in Fig. 5. The figure demonstrated that the CAIOA-RS technique has accomplished improved performance over the other techniques. Meanwhile, the RD and WRR methods have obtained poor CPU utilization over the other techniques. Likewise, the DLB and LB-BC techniques have demonstrated effectual outcome slightly improved CPU utilization over the earlier methods. In the same way, the LB-RC and IPSOF techniques have portrayed moderate CPU utilization performance. Meanwhile, the IPSOF and FIMPSO algorithms have accomplished competitive CPU utilization performance. Finally, the CAIOA-RS technique has resulted in a higher CPU utilization over the other techniques.

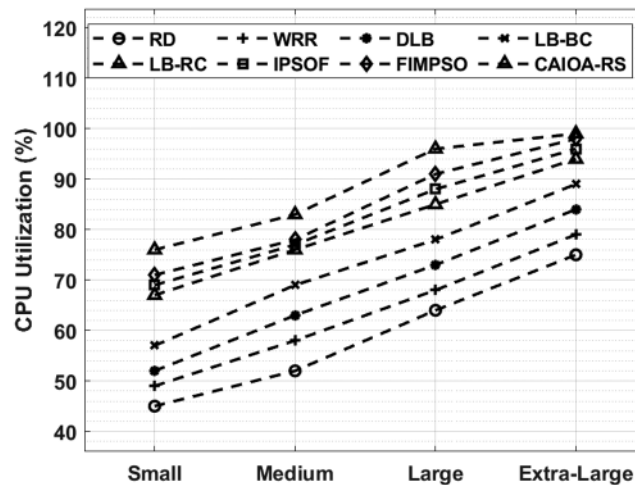
**Table 2:** Comparison study of CAIOA-RS in terms of various measures

Methods	CPU utilization (%)				Memory utilization (%)			
	Small	Medium	Large	Extra-large	Small	Medium	Large	Extra-large
RD	45	52	64	75	40	50	61	72
WRR	49	58	68	79	44	54	64	76
DLB	52	63	73	84	49	59	69	81
LB-BC	57	69	78	89	53	63	72	84
LB-RC	67	76	85	94	57	67	79	89
IPSOF	69	77	88	96	58	69	83	91
FIMPSO	71	78	91	98	60	70	85	93
CAIOA-RS	76	83	96	99	68	74	89	95

(Continued)

**Table 2:** Continued

Methods	Makespan (s)				Average throughput (%)			
	Small	Medium	Large	Extra-large	Small	Medium	Large	Extra-large
RD	65	120	201	288	65	54	42	30
WRR	60	110	195	280	72	63	47	36
DLB	55	103	188	273	81	70	54	44
LB-BC	52	98	180	261	90	76	64	53
LB-RC	48	92	175	153	96	84	72	65
IPSO	47	89	172	150	98	84	71	68
FIMPSO	45	86	169	148	99	87	76	72
CAIOA-RS	42	81	163	142	99	92	86	79

**Figure 5:** Comparative CPU utilization analysis of CAIOA-RS techniques

Memory utilization of the CAIOA-RS approach with recent methods in Fig. 6. The figure outperformed that the CAIOA-RS manner has accomplished enhanced performance over the other approaches. Also, the RD and WRR approaches have gained worse memory utilization over the other methods. Similarly, the DLB and LB-BC techniques have outperformed effectual outcome somewhat increased memory utilization over the earlier techniques. In addition, the LB-RC and IPSOF approaches have showcased moderate memory utilization performance. In the meantime, the IPSOF and FIMPSO methodology have accomplished competitive memory utilization performance. Eventually, the CAIOA-RS approach has resulted in superior memory utilization over the other algorithms.

Fig. 7 examines the performance of the proposed CAIOA-RS technique with existing techniques that take place interms of makespan. The figure showcased that the RD and WRR models have accomplished ineffective performance with the higher makespan. In addition, the DLB and LB-BC techniques have obtained somewhat performance with the moderate makespan. Followed by, the IPSOF and FIMPSO techniques have accomplished near optimal performance with the

competitive makespan. At last, the CAIOA-RS technique has resulted in a lower makespan under different types of tasks.

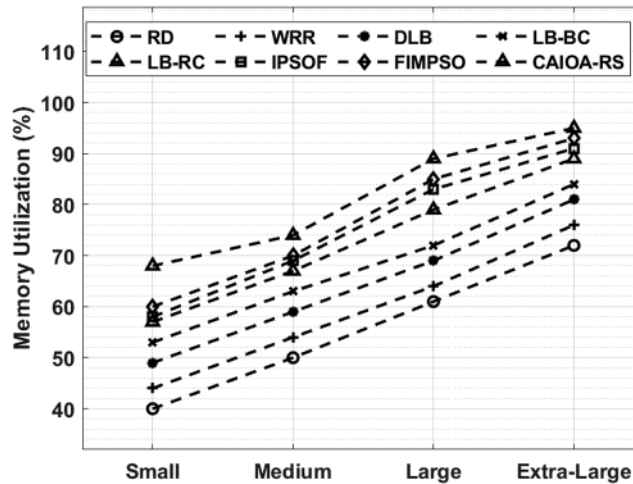


Figure 6: Memory utilization analysis of CAIOA-RS with existing techniques

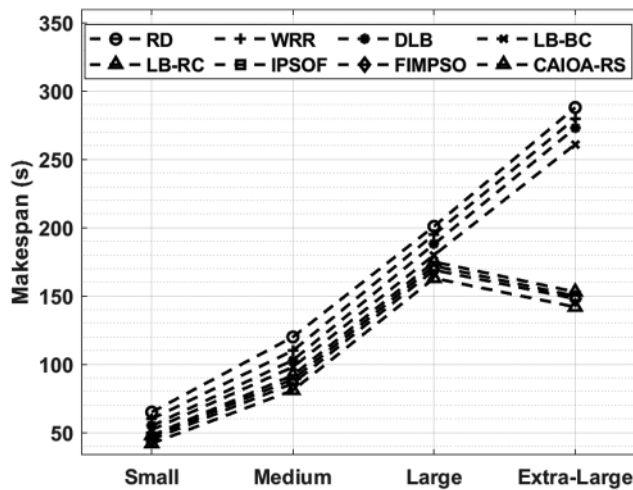
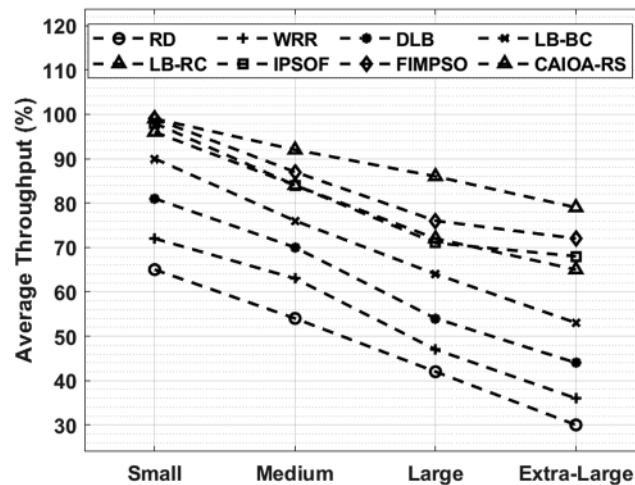


Figure 7: Makespan analysis of CAIOA-RS model with other approaches

Finally, an average throughput of the CAIOA-RS approach with existing algorithms in Fig. 8. The figure outperformed that the CAIOA-RS technique has accomplished higher efficiency over the other manners. Afterward, the RD and WRR algorithms have achieved least average throughput over the other methods. Concurrently, the DLB and LB-BC techniques have displayed effectual results slightly maximum average throughput over the earlier approaches. Besides, the LB-RC and IPSOF algorithms have demonstrated moderate average throughput performance. Meanwhile, the IPSOF and FIMPSO technique have accomplished nearly optimum average throughput performance. Lastly, the CAIOA-RS methodology has resulted in a superior average throughput over the other algorithms.



**Figure 8:** Average throughput analysis of CAIOA-RS model with existing techniques

## 5 Conclusion

This paper has developed an effective CAIOA-RS technique in the IoT enabled cloud environment. The proposed CAIOA-RS algorithm resolved the problem of resource allocation in the IoT enabled cloud environment. It also satisfies the makespan by carrying out the optimum task scheduling process with the distinct strategies of incoming tasks. The design of CAIOA-RS technique incorporates the concept of chaotic maps into the conventional AIOA to enhance its performance. An extensive simulation analysis is performed on the CloudSim platform and assessed the outcomes interms of different performance measures. The simulation results demonstrate that the CAIOA-RS technique indicates that the proposed model is superior to other techniques. As a part of future scope, data offloading mechanism can be designed to accomplish optimum resource allocation in the IoT enabled cloud environment.

**Funding Statement:** This research was supported by Korea Institute for Advancement of Technology(KIAT) grant funded by the Korea Government(MOTIE) (P0012724, The Competency Development Program for Industry Specialist) and the Soonchunhyang University Research Fund.

**Conflicts of Interest:** The authors declare that they have no conflicts of interest to report regarding the present study.

## References

- [1] K. Shankar, E. Perumal, M. Elhoseny and P. Thanh Nguyen, "An iot-cloud based intelligent computer-aided diagnosis of diabetic retinopathy stage classification using deep learning approach," *Computers, Materials & Continua*, vol. 66, no. 2, pp. 1665–1680, 2021.
- [2] M. Z. Hasan and F. Al-Turjman, "Optimizing multipath routing with guaranteed fault tolerance in internet of things," *IEEE Sensors Journal*, vol. 17, no. 19, pp. 6463–6473, 2017.
- [3] N. A. B. Namboothiri and R. J. S. Raj, "A comparative study on job scheduling algorithm augmenting load balancing in cloud," in *2016 Second Int. Conf. on Science Technology Engineering and Management (ICONSTEM)*, Chennai, India, pp. 582–588, 2016.
- [4] R. J. S. Raj and S. V. M. Prasad, "Survey on variants of heuristic algorithms for scheduling workflow of tasks," in *2016 Int. Conf. on Circuit, Power and Computing Technologies (ICCPCT)*, Nagercoil, India, pp. 1–4, 2016.

- [5] A. F. S. Devaraj, M. Elhoseny, S. Dhanasekaran, E. L. Lydia and K. Shankar, "Hybridization of firefly and improved multi-objective particle swarm optimization algorithm for energy efficient load balancing in cloud computing environments," *Journal of Parallel and Distributed Computing*, vol. 142, pp. 36–45, 2020.
- [6] Y. Zhang and J. Sun, "Novel efficient particle swarm optimization algorithms for solving QoS-demanded bag-of-tasks scheduling problems with profit maximization on hybrid cloud," *Concurrency and Computation: Practice and Experience*, vol. 29, no. 21, pp. e4249, 2017.
- [7] K. Shankar, "Iot with cloud based end to end secured disease diagnosis model using light weight cryptography and gradient boosting tree," *Recent Advances in Computer Science and Communications*, vol. 14, no. 8, pp. 1–8, 2021.
- [8] T. Vaiyapuri, V. S. Parvathy, V. Manikandan, N. Krishnaraj, D. Gupta *et al.*, "A novel hybrid optimization for cluster-based routing protocol in information-centric wireless sensor networks for IoT based mobile edge computing," *Wireless Pers. Commun.*, Ahead of Print, 2021. <https://doi.org/10.1007/s11277-021-08088-w>.
- [9] J. Praveenchandar and A. Tamilarasi, "Dynamic resource allocation with optimized task scheduling and improved power management in cloud computing," *Journal of Ambient Intelligence and Humanized Computing*, vol. 12, no. 3, pp. 4147–4159, 2021.
- [10] G. Rjoub, J. Bentahar and O. A. Wahab, "Bigtrustscheduling: Trust-aware big data task scheduling approach in cloud computing environments," *Future Generation Computer Systems*, vol. 110, pp. 1079–1097, 2020.
- [11] M. Z. Hasan and H. Al-Rizzo, "Task scheduling in internet of things cloud environment using a robust particle swarm optimization," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 2, pp. e5442, 2020.
- [12] M. Abd Elaziz, L. Abualigah and I. Attiya, "Advanced optimization technique for scheduling IoT tasks in cloud-fog computing environments," *Future Generation Computer Systems*, vol. 124, pp. 142–154, 2021.
- [13] R. M. Abdelmoneem, A. Benslimane and E. Shaaban, "Mobility-aware task scheduling in cloud-Fog IoT-based healthcare architectures," *Computer Networks*, vol. 179, pp. 107348, 2020.
- [14] X. Ma, H. Gao, H. Xu and M. Bian, "An IoT-based task scheduling optimization scheme considering the deadline and cost-aware scientific workflow for cloud computing," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, pp. 249, 2019.
- [15] S. Basu, M. Karuppiah, K. Selvakumar, K. C. Li, S. K. HafizulIslam *et al.*, "An intelligent/cognitive model of task scheduling for IoT applications in cloud computing environment," *Future Generation Computer Systems*, vol. 88, pp. 254–261, 2018.
- [16] F. Al-Turjman, M. Z. Hasan and H. Al-Rizzo, "Task scheduling in cloud-based survivability applications using swarm optimization in IoT," *Transactions on Emerging Telecommunications Technologies*, vol. 30, no. 8, pp. 1–20, 2019.
- [17] J. Xu, Z. Hao, R. Zhang and X. Sun, "A method based on the combination of laxity and ant colony system for cloud-fog task scheduling," *IEEE Access*, vol. 7, pp. 116218–116226, 2019.
- [18] R. K. Jena, "Task scheduling in cloud environment: A multi-objective ABC framework," *Journal of Information and Optimization Sciences*, vol. 38, no. 1, pp. 1–19, 2017.
- [19] J. Li and Y. Han, "A hybrid multi-objective artificial bee colony algorithm for flexible task scheduling problems in cloud computing system," *Cluster Computing*, vol. 23, no. 4, pp. 2483–2499, 2020.
- [20] M. Mobini, Z. Mobini and M. Rabbani, "An artificial immune algorithm for the project scheduling problem under resource constraints," *Applied Soft Computing*, vol. 11, no. 2, pp. 1975–1982, 2011.