Journal of Positive School Psychology
2022, Vol. 6, No. 2, 4395–4410 http://journalppw.com

Implementation of dynamic resource allocation using Adaptive Fuzzy Multi-Objective Genetic Algorithm for IoT based cloud computing system

J. Mahalakshmi[1], P. Venkata Krishna [2]

[1] Department of Computer Science, Bharathiar University, Tamilnadu, India
[2] Senior Member IEEE, Department of Computing Science, Sri Padmavati Mahila Visvavidyalayam, Tirupati, India
[1] Mahalakshmi1203@gmail.com, [2]pvk@spmvv.ac.in

Abstract

IoT based cloud computing system faces new challenges every day, due to the complex structure of system clusters and high volume of data processed by the systems. The ability of acquiring resources in an elastic manner is considered as the primary rationale for adopting IoT based cloud computing system. Elasticity mainly supports the facility to grow and shrink the virtual resources dynamically according to the requirement of IoT based cloud users. The Resource Allocation proposes Enhanced Genetic Algorithm is proposed to solve these problems, which is used to accomplish better virtual machine allocation across IoT based cloud servers for maintaining vertical elasticity. The resource under utilization problem and high operational cost of IoT based cloud system noticed in the dynamic resource allocation (DRA) technique motivated to propose Resource Allocation Technique. DRA employs the Adaptive Fuzzy Multi-Objective Genetic Algorithm (AFMOGA) for efficient resource allocation using Horizontal Elasticity approach in second contribution. The objectives of the problem are to maximize the Load Distribution and to improve the Resource Utilization, thereby tried to provide the horizontal elasticity for resource allocation in an efficient manner. Further, the proposed technique also reduces the virtual machine migration count. Moreover, the proposed algorithm provides perfect balance between the exploration and exploitation processes by efficiently making use of adaptive genetic operators integrated with the Fuzzy Inference System (FIS). The simulation results shows that the proposed method outperforms compared to the state of art approaches.

Keywords— Dynamic resource allocation, Multi-Objective Genetic Algorithm, Fuzzy Inference System.

INTRODUCTION

In IoT based cloud computing technique [1], the services are abstracted and provided to the users over the internet in a distributed manner and these services are accessed through the networks. Its essential goal is to serve all users with high reliability and better performance. This technique becomes a good choice for several business contexts [2]. Users in a IoT based cloud are allowed to attain the resources by initializing the QoS. IoT based cloud consumers request various services based on their dynamic needs in a IoT based cloud computing environment. Resources are used on a rental basis instead of owning the resources for their business. This saves the cost and reduces risks in managing the resources.

Nowadays any business or organizations [3] are using ―IoT based cloud for their websites. There are many more uses of IoT based cloud in current market for example like popular social networking websites: Facebook, YouTube, Twitter, LinkedIn, Google Plus, ClassMates and many more are hosted on IoT based cloud as well as email providers: Google Apps, Yahoo Zimbra, PanTerra Networks, Microsoft Exchange Online [4]. One can get their computed solution from anywhere at any time. When we surfing on any website or search engine we find IoT based cloud everywhere, Each and every IT magazine describes the new technologies introduced for IoT based cloud. We can upload documents on IoT based cloud for sharing. Online games required very much space but IoT based cloud makes it easy for online games providers, they can host their online games on IoT based cloud with minimum cost. From the survey, nowadays many developers from Microsoft, Amazon, AT&T, Google, GoGrid, RightScale, NetSuite, Enomaly and many more are working on IoT based cloud services [5]. It is a type of computing system which is basically an internet-based that gives access and services that are not on local computers or datacenter but reside on remote location. These services [6] are provided as per the consumption of particular services by the consumer respect to pay-as-you-go. IoT based cloud should ensure that all requested services are available for the end users. Limited resource availability in a IoT based cloud makes IoT based cloud provider‘s job more difficult. Allocation of resources and satisfying the user QoS requirements are important issues in a IoT based cloud computing environment.

This article deals with multi-objective task allocation [7] using Adaptive Fuzzy Multi objective Genetic Algorithm (AFMOGA) Optimization. The main objective of this chapter is to allocate the available resources efficiently in a IoT based cloud environment. Resources are in form of manpower, network, processor and computer nodes. Earlier, researchers used to allocate the tasks to the resources less effectively using standard algorithms like Genetic Algorithm (GA), Ant Colony Optimization (ACO) algorithm [8] and Particle Swarm Optimization (PSO). Allocating the resources to the tasks using hybridization of optimization algorithms can improve the efficiency of the resource allocation. The proposed AFMOGA algorithm helps to optimize the QoS parameters during resource allocation, when compared to traditional algorithms. In a Hybrid AFMOGA Optimization [9] algorithm are combined to attain a better resource allocation. Main aim of this chapter is to improve the IoT based cloud computing system throughput by optimizing the QoS parameters like minimizing the response time, completion time and make span for the user tasks using hybrid model [10]. The benefits of using hybrid resource allocation in IoT based cloud computing environment are

- Utility based task scheduling in terms of memory and execution time
- Efficient resource allocation methods in terms of response time, completion time, make span and throughput.

Literature survey

In [11] authors suggested an algorithm known as Particle Swarm Optimization, which is dependent on the meta-heuristic optimization approach, whose objective is to reduce the total cost of task execution when fulfilling the target limitations. Author heuristic is assessed employing IoT based cloudSim as well as different popular scientific workflows of various sizes. The results indicate that the performance of the approach is much better compared to the present benchmark algorithms.

In [12] authors introduced a robust resource supply approach to IoT based cloud computing, based on reliable supply of resources, and established the adaptive frame for IoT based cloud service (IoT based cloud_RRSSF). This framework renders IoT based cloud service for adaptive adjustment in accordance to conditions which occur during the user/consumer service or supply irregularities.

In [13] authors suggested a novel allocation approach known as Dominant Resource with Bottlenecked Fairness (DRBF) that is a generalization of Bottle- neck-aware Allocation (BAA) to the settings of Dominate Resource Fairness (DRF). The author categorized users into various queues by means of their pre-dominant resources.

In [14] authors studied IoT based cloud Computing to be one among the developing technologies that extend the boundary of internet by making use of the centralized servers for the maintenance of data and resources. It facilitates the users and the consumers to make use of different applications rendered by the IoT based cloud provider.

In [15] authors introduced Dynamic Evaluation Framework for Fairness (DEFF), which helps in evaluating the fair behavior of a resource-based algorithm. In this framework, two forms of sub-models, Dynamic Node Model (DNM) and Dynamic Demand Model (DDM) are presented to define the active features of resource demand as well as identify the number of computing node within IoT based cloud service environment.

In [16] authors formulated an Energy-aware Resource Allocation technique, known as EnReal. Fundamentally, the author presents leverage on the dynamic implementation of virtual machineries to execute scientific workflow.

In [17] authors introduced a resource allocation scheme for machines on the IoT based cloud, depending on the principles of coalition creation and the uncertainty principle of game theory. The author performed the comparison of the results of using this technique with the available resource allocation techniques, which have their deployment done on the IoT based cloud.

In [18] authors introduced SLA based customer-oriented resource provisioning algorithms for minimizing the cost by reducing the penalty and resource cost and enhancing CSL by reducing SLA offenses. The newly introduced provisioning algorithms take into consideration the consumer profiles and quality parameters of providers (for instance, the response time) to deal with dynamic consumer demands and infrastructure-level heterogeneity for business systems.

In [19] authors introduced a heterogeneous resource allocation technique, known as skewness-avoidance multi-resource allocation (SAMR), for allocating the resource in accordance with the versatile needs on various kinds of resources.

Resource allocation system design

The proposed model helps to maximize the throughput by minimizing the response time, completion time and make span in the IoT based cloud environment. IoT based cloud user tasks can be classified into two types. They are repeated tasks and new tasks. Repeated tasks are already executed in a IoT based cloud environment. These tasks can be assigned to the same type of resources based on priority in the priority queue. So starvation of repeated tasks can be eliminated in the proposed model. Figure 1 explains the IoT based cloud data center selection in the resource manager. Each data center contains the various resources to handle the user tasks. Data center which has less resource is considered unsuitable to offer services to the users. Resource manager evaluates the data centers by checking resource details to find out whether resources are available or not. Resource manager selects the suitable data center which has exact resources to handle the user tasks.

An architectural and deployment paradigm must be designed by the IoT based cloud provider to get maximum benefit from IoT based cloud Computing. Various IoT based cloud service providers employ different IoT based cloud architectures at present. In this work, the IaaS IoT based cloud model with two-tier architecture has been taken into consideration. The conceptual diagram of the Two-Tier IoT based cloud Architecture is shown in Fig. 2. Each IoT based cloud would have a IoT based cloud controller component which is responsible for all the services that it provides. Each IoT based cloud server has a processing unit and memory utilization vector which shows current memory and processing unit utilization status. The main goal of virtualization in IoT based cloud is to run multiple Virtual Machines (VMs) on a single machine and sharing these resources among multiple users. Hence, VM is the most vital element in IoT based cloud Computing. According to the user requests, the VMs are allocated in various IoT based cloud servers as per the command of the IoT based cloud resource manager. The role of the IoT based cloud resource manager is to efficiently handle the entire resources under it.

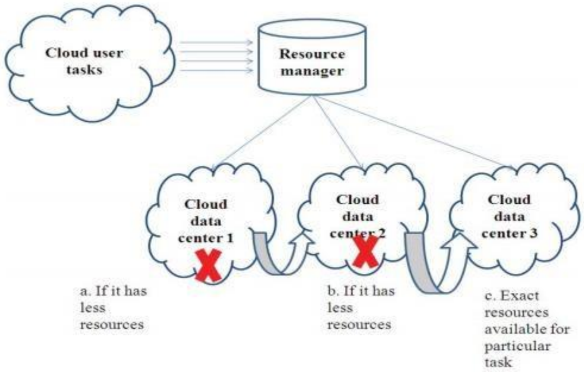

Figure 1 Selection of Optimal Resource from IoT based cloud Data Centre

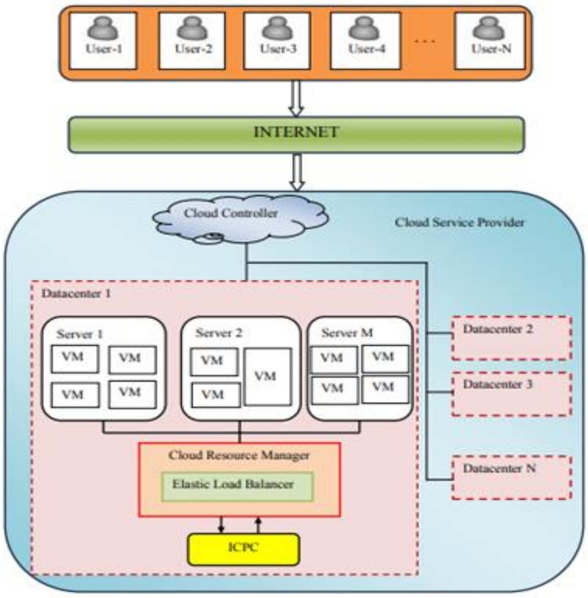

Figure 2 Conceptual diagram of the Two-Tier IoT based cloud Architecture

In this work, the task of the IoT based cloud resource manager has been concentrated with the goal of load balancing. Specifically, the chore of the Mobile IoT based cloud Load Balancer (ELB) component presented in a IoT based cloud Datacenter has been investigated. The ELB component is the one which tries to maximize the benefits of underlying IoT based cloud resources in each Datacenter. It is accountable for the user request handling, monitoring and the scheduling of VMs. The block diagram of the ELB is shown in Figure 2. The task of the Mobile IoT based cloud Load Balancer is to generate the schedule for the assignment of VMs to the available servers with the objective to improve the Load Distribution and also to improve the Resource Utilization. Initially, the user requests are buffered in the VM Queue. The arrival time of user requests are stored in VM status updater. Then the VM requests from VM Queue are given to the Adaptive Fuzzy Multi objective Genetic Scheduler (AFMOGS), at each scheduling round. The AFMOGS employs the AFMOGA to satisfy request of the IoT based cloud user in co-ordination with the horizontal mobile IoT based cloud handler and Inter IoT based cloud Provider Consortium (ICPC). ICPC is used for accessing adequate resources among IoT based cloud providers for satisfying IoT based cloud user demand. Resource allocation and de- allocation process is done by Resource Allocator based on the schedule. User request arrival time, Expected VM Migration Count, Load Distribution, Resource Utilization Rate, Response Time and physical mapping information are updated in the performance evaluator.

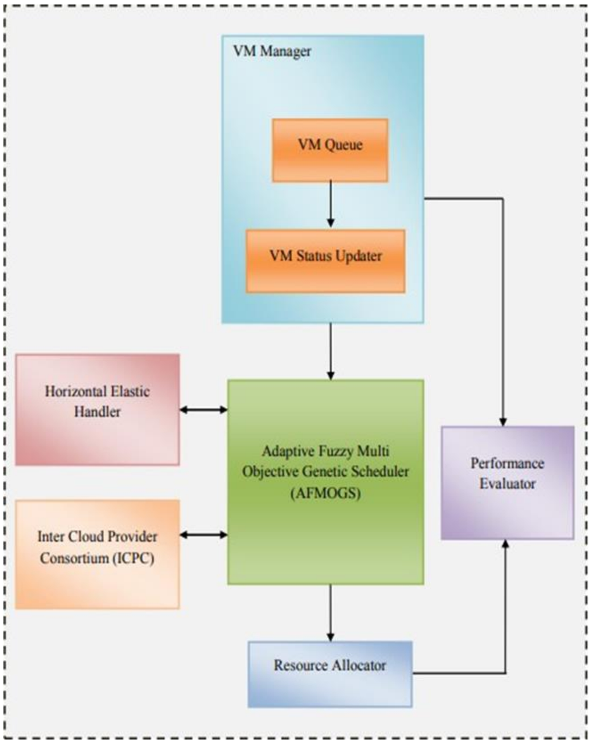

Figure 3 Block diagram of the Mobile IoT based cloud Load Balancer

In IoT based cloud Computing environment, though the arrivals of resource requests are dynamic in nature, at a particular time instance, it can be treated as a definite set of jobs to be executed by the available machines. In this work, the requests for VMs are considered as jobs and the available IoT based cloud servers are treated as machines in this context. Thus, the problem to be solved is to find the best assignment of VM resources on the available servers in order to make the objective values optimal. Each VM resource allocated by the server is not only based on the resource availability of the IoT based cloud provider, but also it adheres to the service level agreements in order to maintain the QoS of the IoT based cloud services. In general, CPU or memory or storage can be considered as a VM resource. Since, the IaaS IoT based cloud model had used in this work, the memory is taken into account as the VM resource in the resource allocation problem.

Proposed Optimization methodology

Multi-Objective Optimization (MOO)

A single objective function is adequate to find the best solution among the set of feasible solutions while solving many optimization problems. Nevertheless, single objective optimization is not enough for solving the real world problems nowadays. Because most of the decision making situations and problems in real-time are complex to resolve as they deal with multiple objectives. As several objectives need to be concentrated in a mutually exclusive manner for multi-objective problems, all objectives cannot be optimized at the same time. There are many ways to address the MOO problems. One way is to use the scalarization technique for solving the multi-objective problems. It applies weightage for each objective and combines them into a single objective in order to solve the problem. However, the problem with this approach is that, it may not get variety of Pareto-set solutions. Also, it cannot explore all the points in the Pareto optimal solution set as the solution set is very much sensitive to the initial weight vector. In order to overcome this problem, each objective is taken separately and carefully tuned in order to obtain the set of compromised solutions. Thus various optimal solutions can be produced by individually optimizing each competing objective. This set of trade-off optimum solutions is collectively called as ―Pareto Optimum set‖ or ―Pareto Front‖. The Pareto Front is based on the concept of non-domination. A solution is said to be non- dominated solution when it satisfies anyone of the following conditions:

i. A solution should be equally good or better than the other solution while comparing it with respect to all objectives, but must be better than the other solution atleast one objective.
ii. If the solution is not able to identify which one is better while comparing it with the other, then these solutions are also called non-nominated solutions.

The problem taken in this chapter is a MOO problem which has two different objectives namely, IoT based cloud Load Distribution Variance and Resource Utilization Rate. Hence, this chapter proposes a multi-objective algorithm in order to address the mobile IoT based cloud resource problem

Algorithms for Multi-Objective Optimization

Various techniques are used to solve MOO problems such as Ɛ-constraints method, Goal programming, Multi-level programming, Dynamic programming and evolutionary algorithms. Among them evolutionary algorithms and particularly, Genetic Algorithm gains more interest for solving the MOO problems. GA is one of the popular evolutionary search algorithms for solving MOO problems. It is a processing based on the feature called genetic and natural selection. The background idea of GA is to begin with randomly generated chromosomes and to follow through strategy of ―survival of the fittest‖ in order to formulate better solutions. GA has been very successful optimization tool, because of its global perspective, simplicity and capability to inherit parallelism. Further, the algorithm provides a potent heuristic search and explores solutions in large search space. Moreover, it can be used for both single and MOO problems.

Work-Flow of GA

The work-flow of GA is comprised of initial population, fitness calculation, selection, crossover, mutation and termination, which is shown in table 1. The algorithm is able to function with set of points simultaneously at each run. The GA initiates its search with a random number of candidate solutions. Each chromosome is assigned with a fitness value which depicts the objective value of the optimization problem. Then, the new population is produced with the support of genetic operators such as selection, crossover and mutation. The algorithm works until the stopping criterion is reached.

Table 1: Work-Flow of Genetic Algorithm

Begin
Create initial populationrepeat
Calculate fitness for each candidate solution
Apply selection operator
Apply reproduction operators
Select individuals for the next generationuntil (stopping criteria reached)

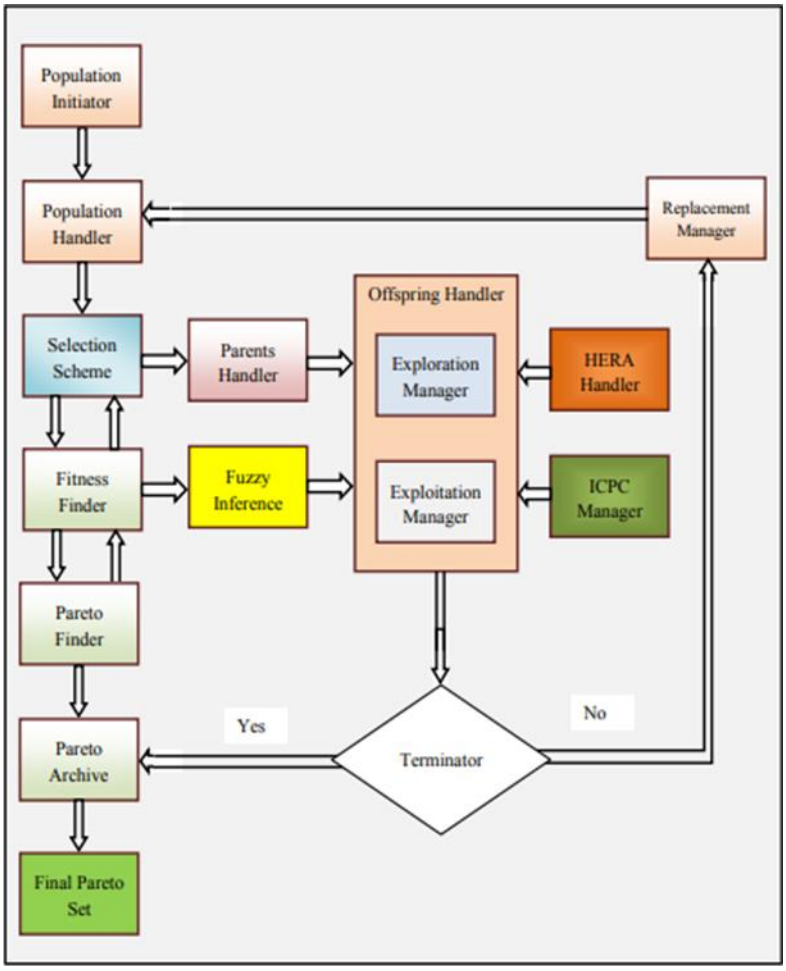

Figure 4: Schematic Representation of the AFMOGA

Variants of GA

Many variants of GA are proposed particularly, for solving multi-objective problems. Some of them are discussed as follows: Vector Evaluated Genetic Algorithm (VEGA) is considered as the first multi objective metaheuristics approach. It separates the entire population into subpopulations and allots different objective value to each one. It adapted the traditional Genetic Algorithm with an improvised selection mechanism. However, the algorithm provides a set of solutions that are not uniformly dispersed in the Pareto front and it is also sensitive to any one particular objective. Multi-Objective Genetic Algorithm (MOGA) employed the sharing of fitness value mechanism in order to improve exploration in the solution space. But, the algorithm had slow convergence rate. Non-dominated Sorting Genetic Algorithm (NSGA-II) is applying niche formation technique and ranking classification procedure to find the non-dominated points. Furthermore, it applied crowding distance in the objective function space. However, various multi-objective GAs were proposed in the literature to find the optimal solution for various problems, there are still needs to tune the genetic parameters to obtain better outcomes. Furthermore, the main pitfall of the above approaches is that, knowledge about the problem domain and related factors are required to find the no dominated solutions. Thus, in order to overcome these issues the AFMOGA is proposed in this work.

AFMOGA

One of the major issues in applying GA for solving optimization problems is the parameter tuning. The fine-tuned parameters of GA often help to balance between the exploration and exploitation during the search process. Further, it also makes the GA to prevent and escape from the problem of local optimum. Hence, the parameters namely, crossover and mutation probability values are tuned adaptively in AFMOGA.

Working Cycle of AFMOGA

The working cycle of AFMOGA starts with the creation of initial population with the support of population initiator. Once the initial population is created, then it is assigned to the population handler as current population. Then, the selection scheme works based on the fitness found by the fitness finder for the current population. Further, the fitness finder provides information regarding the convergence state of the problem to the fuzzy inference system (FIS). Since the problem taken in this chapter is a moo problem, the fitness value is calculated based upon the pareto rank found using the pareto finder. Parent handler is responsible for producing and maintaining the mating pool. The AFMOGA gives chances for all pareto front layers for participating in the mating pool and hence promotes the diversity among chromosomes in the population. All chromosomes from the first pareto front are thrown into the mating pool and thus pave way for preserving the best chromosomes.

The remaining chromosomes are taken randomly from all the other layers based on the expected VM migration count. Hence, the mating pool is prepared and given to the offspring handler. The FIS calculates the crossover and mutation probability values and assigns it to the offspring handler. The offspring handles the reproduction process through the exploration and exploitation managers. Once the offspring population is produced, the terminator checks whether the maximum iterations are reached or not. If the maximum iterations are reached, the terminator stops the execution. Otherwise, it transfers the control to replacement manager which is used to produce the current population by combining offspring and parent.

Hera handler is used to manage the CM resources in horizontal mobile IoT based cloud manner. ICPC manager is used to access the additional resources from other IoT based cloud service providers, if the resources are not sufficient in the primary IoT based cloud provider. Pareto archive is used to store and update the pareto front founded by pareto finder in each iteration. The final pareto set is returned from the pareto archive after the working cycle of AFMOGA stops. The schematic representation of the AFMOGA is shown in figure 4. The respective pseudo code for the AFMOGA is given in table 2 as follows.

Table 2: Pseudo code for AFMOGA The pseudo code for AFMOGA algorithm

Step1: Initialize Population_size, Max_generations, cr_pb, mt_pb, Pareto_archive, Mating_Pool, Parent_Population, Offspring_Population as empty;
Step 2: Call Create_Initial_population( );
Step 3: Assign Current_population=Initial_Population;
Step 4: For i=1 to Max_generations

Step 5: Call RBBW_Selection(Current_population);
Step 6: Parent_Population=Mating_Pool;
Step 7: Find cr_pb, mt_pb values using Fuzzy Inference System;
Step 8: Call Server_Mask_Crossover(Parent_Population) ;
Step 9: Call Repair_Infeasible(Offspring_Population);
Step 10: Call VM_Swap_Mutation(Offspring_Population);
Step 11: Call Repair_Infeasible(Offspring_Population);
Step 12: Replace Current _Population= {Parent_Population} ᴜ {Offspring_Population};
Step 13: End for
Step 14: Call Pareto_ Finder(Pareto_archive);
Step 15 : Output Final_Pareto;

Functional Blocks in AFMOGA

Population initiator and handler

The first step in AFMOGA is the creation of initial population of size 2n, which is done by the POPULATION INITIATOR block. This block is responsible for developing and placing the chromosomes of initial population in the search space. The chromosome represents the schedule for this problem. As chromosome encoding is an important step in GA, the Group based variable length chromosome representation is proposed in this research work for designing the chromosomes. This group based representation reduces the number of bits required as to (n+m) bits whereas the binary encoding requires (n*m) bits. The Figure 5 shows the Group based representation for a sample chromosome. Once the initial population is created, then it is assigned to the Population Handler block which contains the current population. The pseudo code for the initial population creation is shown in Figure 6.

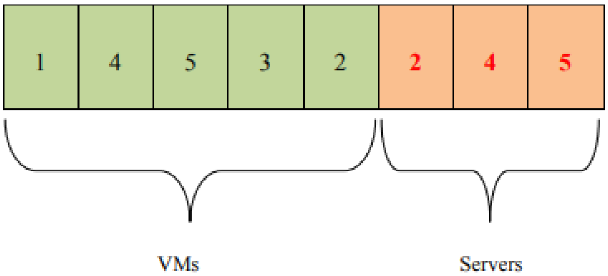

Figure 5 Chromosome with Group based Representation

As the problem taken in this chapter is a constraint optimization problem, the chromosomes are checked for feasibility. A chromosome may have overloaded VM chromosome creation and these chromosomes are treated as infeasible chromosomes. In order to turn the infeasible chromosomes as feasible chromosomes, the repairing procedure is used. If the overloaded VMs are identified, then the overloaded VM list is generated. Then, each overloaded VM is checked in order to be served by any other available server and thereby try to make the chromosome feasible. If the overloaded VMs are not able to accommodate within an available server, new server is added using horizontal mobile IoT based cloudily.

Selection scheme and fitness finder:

Once the current population is assigned to the POPULATION HANDLER, the SELECTION SCHEME is invoked. SELECTION SCHEME is one of the important genetic operators, which is used to select the best parent for reproduction based on the fitness value found by the FITNESS FINDER. Based upon the fitness value of each chromosome in the population, the SELECTION SCHEME chooses the parents to fill the mating pool. Generally, from the perspective of evolutionary algorithms, multi objective and single objective problems are differentiated in the selection scheme. In multi objective problems, parent chromosomes are chosen based on the Pareto dominance, rather than the raw fitness value and so the non- dominated solutions are preferred than the dominated ones. In this proposed AFMOGA algorithm, Pareto Dominance Rank based roulette wheel is proposed as the SELECTION SCHEME.

Pareto finder and archive

Before the selection scheme starts working, the PARETO FINDER is invoked to find the Pareto rank for each chromosome in the current population depending upon non-domination count. Then, the PARETO FINDER assigns rank to all chromosomes in various Pareto layers, based on the level of the layer. Further, the chromosomes within the same layer have same rank, but they are differentiated using the expected migration count as the crowding distance. During the selection process, the first Pareto fronts are always copied to mating pool. In this work, the remaining Pareto layers are also allowed to participate in the reproduction process. Hence, the preservation of diversity among chromosomes is improved and the exploration is done in a well versed manner. Once the Pareto solutions are selected from Pareto layers, then the selection operator returns the mating pool. Then, the mating pool is handled over to the OFFSPRING HANDLER for reproduction. Hence, the proposed SELECTION SCHEME maintains good diversity and also provides dynamic selection pressure during the algorithm execution. Further, the proposed SELECTION SCHEME carries the best chromosomes towards successive generations and stores them in the PARETO ARCHIEVE.

Offspring handler

The OFFSPRING HANDLER contains two blocks namely, EXPLORATION and EXPLOITATION MANAGERS for creating offspring‘s for successive generations, based on the probability values given through the FIS. The EXPLORATION MANAGER performs the Server Mask Crossover to generate a new child chromosome from parent chromosome by replicating selected bits from each parent strings. The EXPLOITATION MANAGER is used to randomly mutate the genes in a chromosome for preserving the gene diversity in the population. VM swap mutation operator is used in this work. If the infeasibility occurs after mutation, the mutation operator tries to convert into feasible form by checking the available space in server. If the space is available, VMs are migrated based on the expected VMs migration count. Even after that if feasibility cannot be reached, the proposed algorithm invokes the HERA handler.

Adaptive Parameter Tuning Using FIS

Traditionally, the crossover and mutation probability values are constant. In the proposed work, the AFMOGA tunes the crossover and mutation probability values dynamically using the FIS. The Crossover probability value defines the number of chromosomes to be newly created for consecutive generations, whereas the mutation probability value refers to number of bit changes within the entire chromosomes. The crossover probability usually has value lies between
0.4 and 0.9 while the mutation probability value ranges from 0.01 to 0.1.The concept behind the parameter tuning in this proposed work is to determine the values for crossover and mutation probabilities adaptively. These probability values are computed based on the information obtained from the current convergence state of the algorithm.

To carry the parameter tuning task, the proposed algorithm checks whether it has converged to the local optimum point or not. In order to detect the local optimum, the algorithm obtains the information concerning the fitness improvement. That is, if the difference between the maximum fitness and average fitness is low, almost all chromosomes share the same fitness value and thus the algorithm has converged. But, the convergence may be either local or global optima for the problem. To fix that, the proposed algorithm checks the number of iterations yet to be reached during algorithm execution. If it is low, then it is understood that the algorithm converges to the global optima. But, if it is high then, the algorithm was trapped into the local optima.

Hence, whenever the algorithm traps into the local optimum problem, the AFMOGA adjusts the crossover and mutation probabilities adaptively with respect to the convergence state of the algorithm. Further, the proposed algorithm provides perfect balance between exploration and exploitation. The AFMOGA uses two FIS namely, Crossover Fuzzy Inference System (CFIS) and Mutation Fuzzy Inference System

(MFIS), in order to determine the crossover and mutation probabilities individually for each objective. Both the CFIS and MFIS contain three input variables and two output variables individually.

Input 1 represents the difference between current iteration and maximum iteration whereas input 2 represents the difference between maximum fitness and average fitness of objective 1 IoT based cloud Load Distribution Variance (CLDV). The input 3 symbolizes the difference between maximum fitness and average fitness of objective 2 Resource Utilization Rate (RUR). The output variables Crossover 1 (Cr1) represent the crossover probability of objective 1. Crossover 2 (Cr2) represents the crossover probability of objective 2. The output variables Mutation1 (Mt1) represents the mutation probability of objective 1 and Mutation2 (Mt2) represents the mutation probability of objective 2. Both CFIS and MFIS had used the Centroid method for Defuzzification.

the Input 1 represents the difference between current iteration and maximum iteration whereas the Input 2 represents the difference between maximum fitness and average fitness values. Both the crossover and mutation probabilities are calculated separately, for each objective namely, IoT based cloud Load Distribution Variance (CLDV) and Resource Utilization Rate (RUR). The same rule base is used for finding both output variables Cr1 and Cr2 for both objectives in CFIS. The average of Cr1 and Cr2 produces the final crossover probability. Further, the same rule base is used for finding both output variables Mt1 and Mt2 for both objectives in MFIS. The average of Mt1 and Mt2 produces the final mutation probability

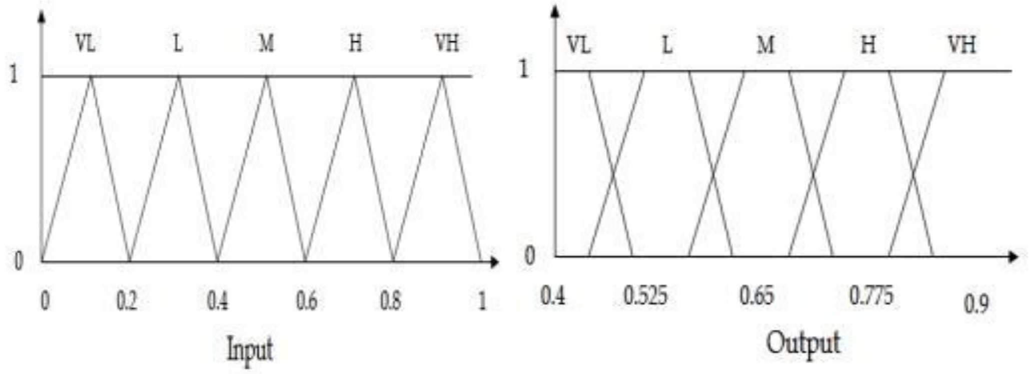

Figure 6: Membership functions used for the fuzzy variables

Experimental Study

Due to the difficulty in testing the proposed system on real systems, simulation based evaluation has been conducted for evaluating the proposed work. In order to investigate the effectiveness of the proposed work, the AFMOGA was implemented and tested with different workloads. The respective results were presented in this section and the performance of AFMOGA was compared with the existing algorithms namely, Enhance Genetic Algorithm (EGA) and Simple Genetic Algorithm (SGA) to reveal the usefulness of proposed work. The IoT based cloudSim 3.0.3 simulator class is extended to simulate the proposed AFMOGA and existing algorithms. The simulation was carried on Intel(R) core (TM) i5 processor 2.6 GHz, 8GB memory and windows 7 platform. The IoT based cloudSim parameter setting is shown in Table 4. The parameters used for the proposed algorithm is shown in Table 3. The metrics taken for analyzing the performance of the proposed algorithm are IoT based cloud Load Distribution Variance, IoT based cloud Resource Utilization Rate, Expected VM Migration Count and Response Time.

Parameter	Value
Number of Datacenter	3
Number of host	20
Type of Manager	Time Shared, Space Shared
Number of PE per Host	2-10
Host Memory (MB)	4096- 10240
Total Number of VMs	25-200
MIPS of PE	1000-2000
Number of PE per VM	1
VM Memory (MB)	512- 4096

Table 3: IoT based cloudSim Parameter

Parameter	Value
Population Size	20
Number of Generations	150
Crossover probability	0.4-0.9
Mutation probability	0.01-0.1
Crossover operator	Server Mask crossover operator
Mutation operator	VM Swap Mutation operator

Table 4: Parameter Setting of AFMOGA algorithm

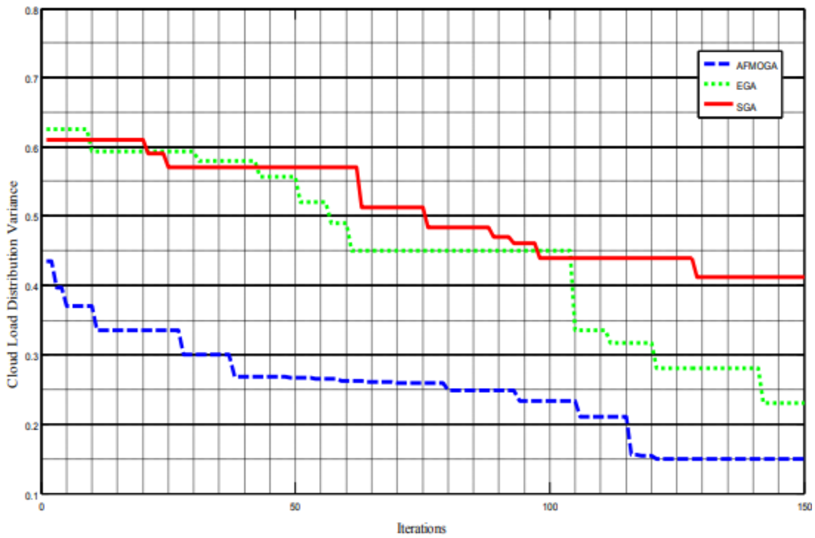

Figure 6: Fitness Evaluation of IoT based cloud Load Distribution Variance

From the Figure 6, it is evident that the AFMOGA provides better IoT based cloud Load Distribution than the Enhance Genetic Algorithm (EGA) and Simple Genetic Algorithm (SGA). To exhibit the efficiency of the proposed algorithm, all the three algorithms namely AFMOGA, EGA and SGA were tested for three different workloads and the results are depicted in Figure 7, where the respective values are tabulated in Table 5.

Types of Load	Cloud Load Distribution Variance		
	AFMOGA	EGA	SGA
Workload1	0.1500	0.2300	0.4119
Workload2	0.1890	0.3500	0.4801
Workload3	0.1892	0.3723	0.5009

Table 5: IoT based cloud Load Distribution Variance for Various Workloads

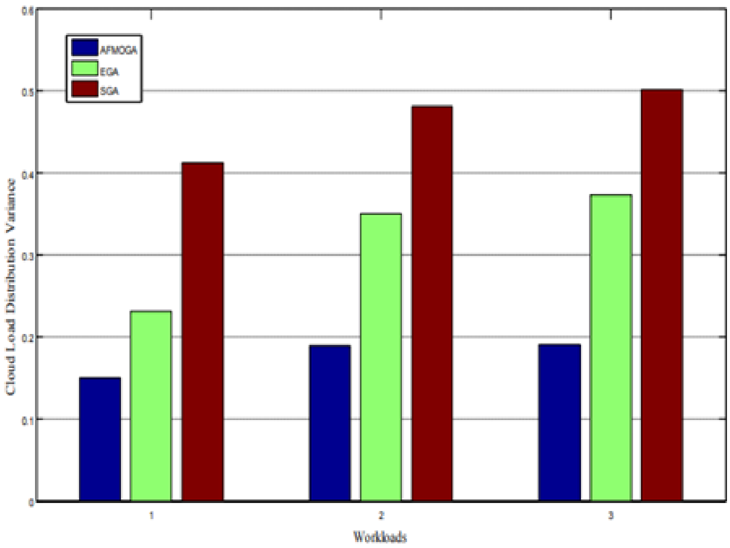

Figure 7: IoT based cloud Load Distribution Variance for Various Workloads

From the Figure 7, it can be clearly observed that the AFMOGA provides better performance in terms of the minimized CLDV value, than the EGA and SGA algorithms for various workloads.

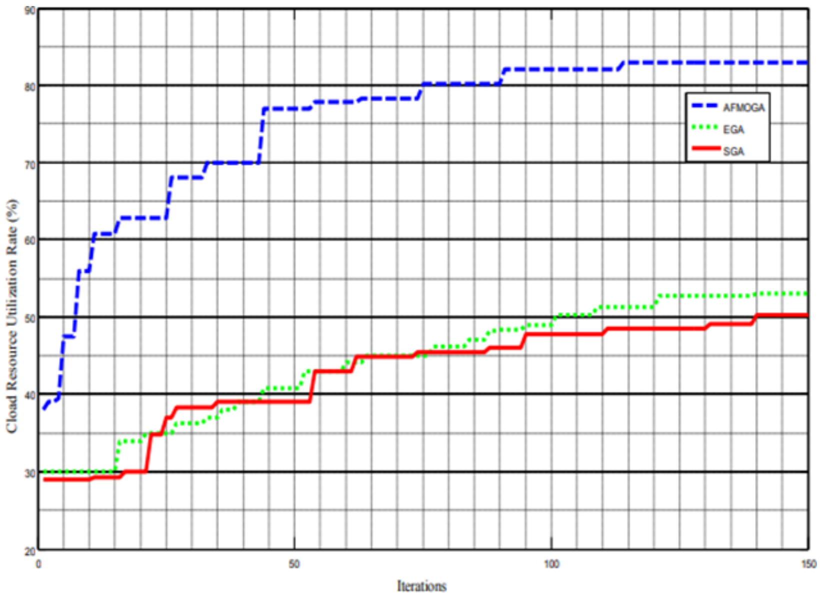

Figure 8 Fitness Evaluation of Resource Utilization Rate

Figure 8 illustrates that the AFMOGA offered remarkable Resource Utilization Rate (RUR) than the other two algorithms and also enhanced the convergence speed of the algorithm. The impact of RUR value for three different workloads is analyzed and it is presented in

Figure 9, where the respective values are tabulated in Table 6.

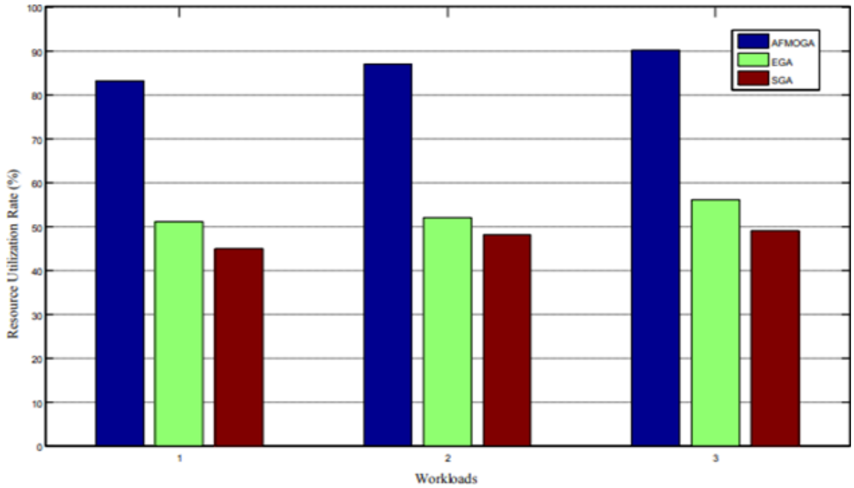

Figure 9: Resource Utilization Rate for Various Workloads

From the Figure 9, it is understood that the AFMOGA provides significant improvement in Resource Utilization Rate (RUR) than the other two algorithms.

Table 6: Resource Utilization Rate for Various Workloads

Types of Load	Resource Utilization Rate (%)		
	AFMOGA	EGA	SGA
Workload1	83	51	45
Workload2	87	52	48
Workload3	90	56	49

From that figure, it is explicitly proved that the AFMOGA reduces the Expected VM Migration Count heavily than the other two algorithms. The Expected VM Migration Count for various workloads founded by the existing algorithms and AFMOGA algorithm are shown in Figure 10. Finally, the Figure 5.11 represents the Response Time calculated for three different workloads by the three different algorithms, where the AFMOGA algorithm minimizes the Response Time in all the three cases of workloads. The Response Time values for various workloads are depicted in Table 7 and Table 8.

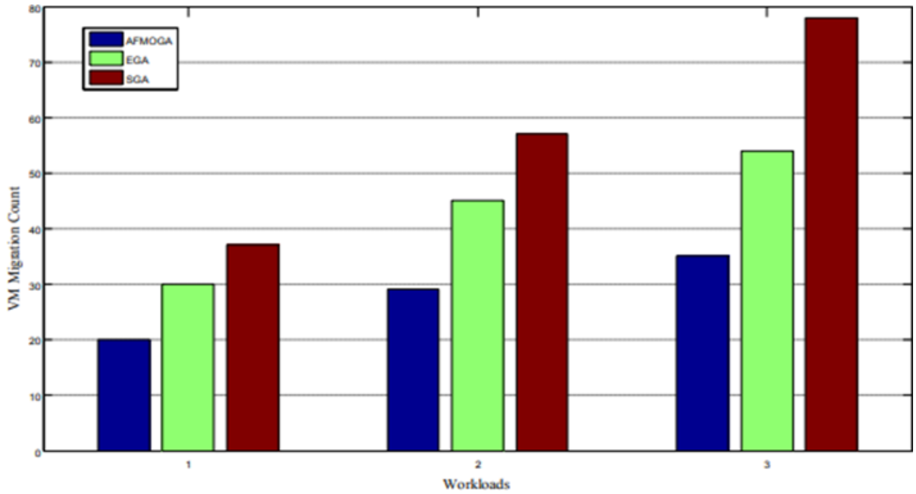

Figure 10: Expected VM Migration Count for Various Workloads

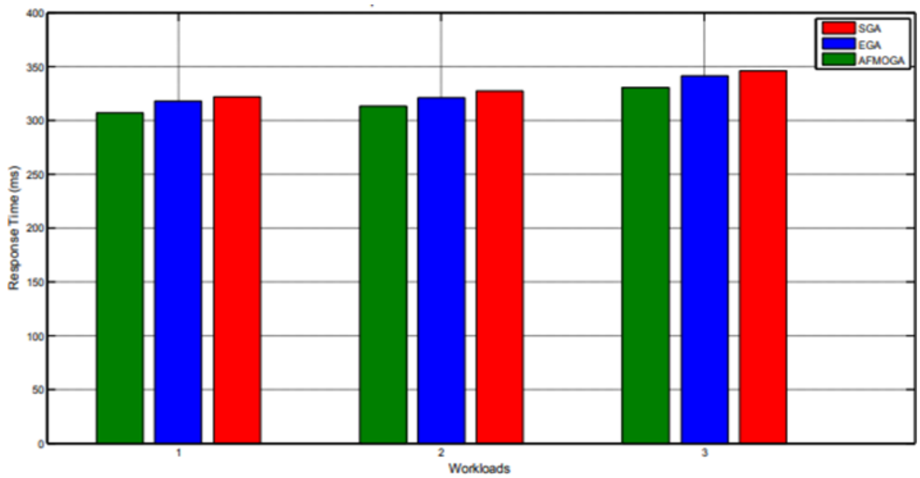

Figure 11: Response Time for Various Workloads

Types of Load	Expected VM Migration Count		
	AFMOGA	EGA	SGA
Workload1	20	30	37
Workload2	29	45	57
Workload3	35	54	78

Table 7: Expected VM Migration Count for Various Workloads

Types of Load	Response Time (ms)		
	AFMOGA	EGA	SGA
Workload1	306.42	317.89	321.56
Workload2	312.64	320.49	327.26
Workload3	329.84	341.34	345.52

Table 8: Response Time for Various Workloads

Conclusion

In this article, the DRA is discussed in detail along with the novel proposed algorithm AFMOGA for figuring out the mobile IoT based cloud resource allocation problem in IoT based cloud computing environment. The proposed algorithm affords optimal VM schedule for the IoT based cloud IaaS model. Further, the proposed algorithm produced good diversity among the Pareto solutions and preserved good solutions simultaneously. In order to expose the usefulness of the AFMOG Algorithm, it was compared with the EGA and SGA. It can be realized that the proposed algorithm expresses better performance in terms of minimized IoT based cloud Load Distribution Variance and maximized the Resource Utilization Rate with better convergence speed. Furthermore, the AFMOG Algorithm also reduced the Expected VM Migration Count and supports horizontal mobile IoT based cloud resource allocation with the help of ICPC. Moreover, the AFMOG Algorithm has been tested with various workloads and it is proved that AFMOGA provides better performance than the SGA and EGA. Hence, it is clearly observed that among the three different algorithms discussed in this chapter, the AFMOGA provides more benefit to IoT based cloud providers as well as to IoT based cloud users. In this contribution, the performance of proposed resource allocation technique in normal IoT based cloud computing environment is analyzed and the work can be further extend to examine the resource allocation in mobile IoT based cloud environment.

References:

1. Di, Sheng, & Cho-Li Wang, 2013, ‗Dynamic optimization of multi attribute resource allocation in self-organizing IoT based clouds.' IEEE Transactions on parallel and distributed systems ,vol . 24, no. 3 ,pp. 464-478.
2. Di, Sheng, Cho-Li Wang, & Franck Cappello,2014, ‗Adaptive algorithm for minimizing IoT based cloud task length with prediction errors.' ,IEEE Transactions on IoT based cloud Computing ,vol. 2, no. 2 ,pp. 194-207.
3. Egwutuoha, Ifeanyi P., Daniel Schragl, & Rafael Calvo,2013, ‗A brief review of IoT based cloud computing, challenges and potential solutions', Parallel & IoT based cloud Computing, vol.2, no.1.
4. Endo, P.T., de Almeida Palhares, A.V., Pereira, N.N., Goncalves, G.E., Sadok, D., Kelner, J., Melander, B. & Mangs, J.E, 2011, Resource allocation for distributed IoT based cloud: concepts and research challenges. IEEE network, vol. 25, no. 4.
5. EshaSarkar, & C.H Sekhar, ‗AOrganizing Data in IoT based cloud using Clustering Approach', 2014, IEEE International Journal of Computer and Electrical Engineering, International Journal of Scientific & Engineering Research, vol.5, no.5.
6. Fargo, Farah, CihanTunc, Youssif Al-Nashif, Ali Akoglu, & Salim Hariri, 2014,Autonomic workload and resources management of IoT based cloud computing services.' Proceedings of International Conference on IoT based cloud and Autonomic Computing (ICCAC), pp. 101-110. 112
7. Feng, Guofu, SaurabhGarg, RajkumarBuyya, & Wenzhong Li,2012, ‗Revenue maximization using adaptive resource provisioning in IoT based cloud computing environments.',In Proceedings of the 2012 ACM/IEEE 13th International Conference on Grid Computing, pp. 192-200.
8. Feng, Hao, Jaime Llorca, Antonia M. Tulino, & Andreas F. Molisch, 2016, ‗Dynamic network service optimization in distributed IoT based cloud networks.' Proceedings of Computer Communications Workshops (INFOCOM WKSHPS), pp. 300-305.
9. Firuzbakht, Alireza, & Asgarali Bouyer, 2013, ‗An optimal Algorithm based on gridding and clustering for decrease energy consumption in WSN', IEEE International Convention on Information & Communication Technology Electronics & Microelectronics, pp. 474- 478.
10. Gouda, K. C., Radhika,T,V & Akshatha, 2013, ‗Priority based resource allocation model for IoT based cloud computing', International Journal of Science, Engineering and Technology Research ,vol. 2, no. 1, pp. 215-219.
11. Goyal, & Sumit, 2014, ‗Public vs private vs hybrid vs communityIoT based cloud computing: A critical review', International Journal of Computer Network and Information Security,vol.6, no.3 .
12. Granmo, Ole-Christoffer, B. John Oommen, SveinArildMyrer, & Morten Goodwin Olsen, 2007,‗Learning automata-based solutions to the nonlinear fractional knapsack problem with applications to optimal resource allocation.' ,IEEE Transactions on Systems, Man, and Cybernetics ,vol. 37, no. 1 ,pp. 166-175.
13. Grati, Rima, KhouloudBoukadi, & Hanêne Ben-Abdallah, 2014, ‗QoS based resource allocation and service selection in the IoT based cloud.' Proceedings of 11th

International Conference on e-Business (ICE-B), pp. 249-256.

14. Gupta, Ajay Kumar, & ShailendraPrakash, 2016, ‗SERVICE ENHANCEMENT USING IOT BASED CLOUD COMPUTING.', International journal of advance research in science and engineering, vol. 5 ,no. 8.
15. Hadji, Makhlouf, & DjamalZeghlache, 2012, ‗Minimum cost maximum flow algorithm for dynamic resource allocation in IoT based clouds.' Proceedings of International Conference In IoT based cloud Computing (IOT BASED CLOUD), pp. 876-882. 113
16. Hao Zheng, Yixiong Feng, & Jianrong Tan,2017, ‗A hybrid energyaware resource allocation approach in IoT based cloud manufacturing environment', IEEE Access ,vol. 5, pp. 12648-12656.
17. .Hasan, Monowar, EkramHossain, & DusitNiyato, 2016, ‗Random access for machine- to-machine communication in LTE-advanced networks: issues and approaches.' ,IEEE Communications Magazine vol 51, no. 86-93.
18. Hassan, Mohammad Mehedi, & Ahmed Alsanad, 2016, ‗Resource Provisioning for IoT based cloud-Assisted Software Defined Wireless Sensor Network.' ,IEEE Sensors Journal ,vol 16, no. 20 ,pp. 7401-7408.
19. .Hassan, Mohammad Mehedi, Biao Song, & Eui-Nam Huh,2011, ‗Distributed resource allocation games in horizontal dynamic IoT based cloud federation platform.',Proceedings of 13th International Conference on High Performance Computing and Communications (HPCC), pp. 822- 827.
20. Helda Mercy, C.Anand, & T.S. Suganya , 2011, ‗Resource Overbooking: Using Aggregation Profiling in large scale Resource Discovery', International Journal of Engineering Trends and Technology, pp. 52-54.