

# Fuzzy Q-Learning Based Controller for Cost and Energy Efficient Load Balancing in Cloud Data Center



Subhra Priyadarshini Biswal, Satya Prakash Sahoo,  
and Manas Ranjan Kabat

**Abstract** The goal of cloud controller is to focus on continuous delivery of services to user on demand basis followed by “pay-per-use” model. Due to the increasing demand of cloud services, energy consumption on data center is increasing rapidly which lead to high operational cost. The harmful emission from this energy intensive data center affects our environment badly and cause climate change significantly. So as an alternative we have focused on onsite green power generation to reduce the harmful effects of greenhouse gases. In this paper, we proposed a fuzzy Q-learning based self-learning controller to optimize the load for specific data center. The proposed method also helps to reduce uncertainty and solve the congestion issue efficiently through fuzzy linguistic behavior and membership function. In this proposal, fuzzy output parameter considered as reward value which is used to learn and update the state for each data centre.

**Keywords** Cloud Computing · Q-learning · Data Center · Renewable Energy · Fuzzy Logic

## 1 Introduction

Cloud computing is an IT paradigm which provide various set of services to user on demand basis, that can be accessible at anywhere irrespective of time and place [1–4]. Here resources and services are always available on the internet and it can be released on demand basis without interacting with service provider. Resources are available to the user on pay per use model. User has to pay according to the resource usage by them like amount of space used, number of CPU cycle etc. Due to the increasing demands of cloud service, energy consumption of data center is increasing rapidly.

---

S. P. Biswal (✉)

School of Computer Science and Engineering, National Institute of Science and Technology (Autonomous), Institute Park, Pallur Hills, Berhampur, Odisha 761008, India

S. P. Sahoo · M. R. Kabat

Department of Computer Science and Engineering, Veer Surendra Sai University of Technology, Burla, Sambalpur, India

© The Author(s), under exclusive license to Springer Nature Singapore Pte Ltd. 2021

151

S. K. Das et al. (eds.), *Architectural Wireless Networks Solutions and Security Issues*, Lecture Notes in Networks and Systems 196, [https://doi.org/10.1007/978-981-16-0386-0\\_9](https://doi.org/10.1007/978-981-16-0386-0_9)

Data center are always deals with providing various set of services like web, e-mail, storage, processing etc. to user [5]. It always deals with delivering applications and services over the internet. For this, they requires higher amount of electricity. It will lead to high operational expenses. It adds significant impact on our environment by carbon emission.

Coal, petroleum and gas not only cause climate uncertainty through emissions of greenhouse gases, but also affect other economic, social and environmental area by adding up a dangerous negative balance sheet. This fossil fuel is not renewable. The harmful emission from this energy affects our environment badly and cause climate change significantly. So, the proposed method used fuzzy Q-learning technique to achieve the main goal. Fuzzy Q-learning is a meta-heuristic method which is used to optimize imprecise information efficiently and produce optimal result [6, 7]. There are several work are proposed based on meta-heuristic and fuzzy logic in the area of network based on optimization technique [8–11]. In the proposed method, renewable energy is derived from various natural processes like from sunlight, water, wind, biomass etc. From various study, it was verified that wind and solar energy has lower cost in comparison to other energy.

There are many approached has been proposed based on fuzzy inference system. In fuzzy system the rules are defined at design-time which leads to the following issues like user cannot describe any rule, user may only specify limited rules for some situations, users specify rules are not always effective and it may lead to uncertainty etc. There are several commonly used approaches are proposed like round robin, weighted round robin, ant colony optimization, particle swarm optimization, first come first serve, shortest job first etc. Each approach has its own advantages and disadvantages. But most of them have disadvantages like higher average waiting time, low throughput, requires detail task information, high response time. So to overcome that complexity here we proposed a fuzzy q-learning algorithm which is known as “a self-learning fuzzy cloud controller”. It helps to run and update each state of the fuzzy logic based rules at runtime.

The main purposes of this paper are as follows.

- (a) Design a self-learning cloud controller by combining Q-learning with fuzzy inference system.
- (b) It gives importance on utilization of renewable energy sources and minimizes the consumption of brown energy.
- (c) Deals with uncertainty caused by the incomplete knowledge.

The rest of the paper contain: Sect. 2 gives brief description cloud computing, Sect. 3 describes the related works done on load balancing problem, Sect. 4 describes about reinforcement learning. Section 5 contains proposed approach which consists of algorithm and fuzzy rule base. Section 6 was about experiments and results. Finally Sect. 7 contains the conclusion.

## 2 Cloud Computing

Cloud computing is an IT paradigm which enables user to consume computable resources (like storage, utility or application) on demand basis, that can be accessible at anywhere irrespective of time and place. Here resources and services are always available on the internet and it can be released on demand basis without interacting with service provider. Resources are available to the user on pay per use model. User has to pay according to the resource usage by them like amount of space used, number of CPU cycle etc.

### 2.1 Characteristics of Cloud Computing

- (a) **On demand service:** Resources are provision on demand of the user without interaction of Cloud Service Provider. So it's an automated process.
- (b) **Broad network access:** Provides platform independent access through heterogeneous client platform.
- (c) **Resource pooling:** A pool of resources are provided by the cloud service provider for multiple user to fulfill their demand.
- (d) **Rapid elasticity:** Resources can be rapidly scaled up or down on users demand basis. There are two types of scaling are there given in below
  - (i) **Horizontal scaling:** it deals with lunching and provisioning additional resources.
  - (ii) **Vertical scaling:** it involves with changing the virtual capacity of the server.
- (e) **Measured service:** Cloud computing provides resources to the user as a measured service. It applies pay-per-use model. User has to pay based on the resource usage from the cloud.
- (f) **Performance:** This may be scale up or down based on dynamic application workloads.
- (g) **Security:** It improves security by centralization of data.

### 2.2 Cloud Models

There are various types of cloud models are available to the user like cloud service model and cloud deployment model. Cloud service model are categorize as IaaS (Infrastructure-as-a-service), PaaS (Platform-as-a-service), SaaS (Software-as-a-service) as shown in Fig. 1. Cloud deployment model are categorized as public cloud, private cloud, hybrid cloud and community cloud as shown in Fig. 1.

SaaS	User has the capability to provision storage & computing
PaaS	User can develop & deploy the application.
IaaS	Provide user interface.

**Fig. 1** Cloud service model

### 2.2.1 Cloud Service Model

(a) **Infrastructure-as-a-Service**

It provides computing and storage resources to the users as virtual machine instances and virtual storage. User can deploy operating system and application of their choice. Cloud service provider maintains the underline infrastructure. Bill will be calculated in pay-per-use model. It means user has to pay for access on number of virtual machine hours and virtual storages. The example of I-a-a-S frame work is Amazon EC2.

(b) **Software-as-a-Service**

It provides user interface to the application itself. CSP maintains all. S-a-a-S applications are platform independent. Here the application software is provided to the user with in the cloud.

### 2.2.2 Cloud Deployment Model

(a) **Public Cloud**

Here cloud services are available to general public. The cloud resources are shared among every individuals, organizations, small and medium enterprises, government etc. The cloud services are provided by the third party provider. The example of public cloud are Amazon EC2 (Elastic Compute Cloud), Google App Engine etc.

(b) **Private Cloud**

Here cloud services are available for only a single organization. A single organization can access the resources from the cloud. So here security is more concern. Here the services are managed internally or by any third party or by the help of any single organization.

(c) **Hybride Cloud**

It combines the services of multiple clouds that may be of private or public. Hybride clouds are best suited for organizations that want their application to be secured applications and want to host their data on a private cloud. It is cost savings with hosting shared applications and data in public cloud.

Public Cloud	Available to general public.
Private Cloud	Available to single organization.
Hybride Cloud	Combine services of multiple clouds.
Community Cloud	Services are share by several users.

Fig. 2 Cloud deployment model

(d) **Community Cloud**

Here cloud services are shared by several organizations that have the same policy. It also supports load balancing techniques. The third party or any organization from the community will manage all the infrastructures (Fig. 2).

### 2.3 Load Balancing

As we know the demand for cloud services increases day by day, as a result it will lead to load balancing as a major problem. When number of user request increases for more resources it becomes difficult for the server to execute their request in less time. So it may lead to lots of complexity with performance degradation. To solve this problem load balancing is efficiently required. This technique helps to distribute the workload among all available servers so that the user can get their resource in less time. Load balancing is an important concept which helps to increase the throughput. As a result, it helps to maximize the user satisfaction level. It helps to enhance the performance of the system by allocating virtual machine for the execution of user’s request in less time. It also tries to minimize the user response time. The below figure will show how the workload will be distributed across all available servers to execute the users request as soon as possible (Fig. 3).

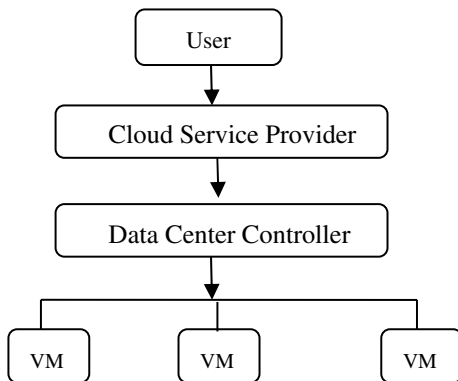
Generally there are so many load balancing algorithms are available. But most of them are either static or dynamic algorithms which are given in below. These algorithms are based on the existing status of the system.

(a) **Static algorithm**

These kinds of load balancing algorithm need earlier knowledge about detail task information to execute the task. It doesn’t consider the present state of the system. This kind of algorithm may face the problem like performance degradation with lots of complexity which may lead to system failure. Some example of this algorithm is Round Robin, max–min load balancing, shortest job scheduling etc.

(b) **Dynamic algorithm**

These algorithms execute the work based on current state of the system. So that there will be no need of any prior knowledge to distribute the workload.

**Fig. 3** Cloud load balance

They give better performance in comparison to static algorithm. The example of dynamic load balancing type of algorithm are fuzzy active monitoring, throttled load balancing etc.

### 3 Related Works

Uma Singhal [12] proposed a new fuzzy logic and GSO based load balancing mechanism for public cloud. Here the cloud was partitioned in to different groups. Each group will chose different load balancing strategy depending on the burstiness of workload. It consists of a cloud controller which will manage the entire load. Then it will forward to the load balancer. At last the most suitable virtual machine was selected based on the information provided by the memory and storage space usage etc. This algorithm has mainly focuses on the several parameters for load balance, i.e. burst detector, fuzzifier and load balancing approach. The fuzzifier is used to enhance the decision of choosing most suitable virtual machine. Here the author did not consider about any energy efficient concept. Pooyan Jamshidi [13] proposed a self-learning cloud controller. It automatically helps to compute the resource at runtime. The proposed algorithm is based on fuzzy inference system and Q-learning. It helps to adjust the rule at runtime. Here the parameter consider for further implementation are workload and response time. The output is change in number of virtual machine. The paper doesn't consider about any energy efficient concept related to the data center. Hamid Arabnejad [14] proposed comparison between different reinforcement learning algorithms. A self-adaptive fuzzy logic controller is combined with two reinforcement learning approaches (Fuzzy SARSA learning and Fuzzy Q-learning). Here both approaches are implemented and compared with their corresponding advantages and disadvantages. Pasha et al. [15] proposed round robin approach was proposed for virtual machine load balancing algorithm in cloud computing environment. It mainly focuses on execution of each and every request arrives at the data center from the user. It increases the throughput efficiently. The algorithm was based on Round Robin

policy. Here time slot is allotted to every process. The workload will be assign to the virtual machine only when the virtual machine will be free. So when huge amount of task will be arrive at a time, then user has to wait until the next virtual machine will be available which the main disadvantage of the algorithm. Michael Dale [16] proposed comparative analysis of different types of renewable energy was proposed. Here the author has compared the costs of Photovoltaic, Solar Thermal, and Wind Electricity generation technologies. It describes the importance of renewable energy. The comparison is done basically on three types of renewable energy sources which are wind energy, photovoltaic solar power and concentrating solar power (CSP). The comparison is based on operational cost, capital cost and levelized cost of electricity. Capital cost means manufacture, energy requirement to process the material etc. Operational cost means energy requirements for proper maintenance of the system. For example washing solar systems, replacing worn parts, energy required to build the spare parts, operating the whole system, energy associate with the fuel cycle etc. Here they have describes different renewable technology sources developed in different country in year wise. By studying all these, they give a graph of capital cost of different renewable energy sources. From this they found that wind energy has lower cost in comparison to photovoltaic solar power and concentrating solar power (CSP). Sanyukta Raje [17] proposed different energy efficiency standard of the data center. Here the energy efficiency concept was based on the country India. The author focused on how the rising electricity cost, higher growth of Information Technology industries and higher utilization of fossil fuel in India will lead to high operational cost. It also gives harmful effects on our environment also. By using fossil fuel the carbon emission from the energy intensive data center are increases rapidly. It will leads to higher operational expenses. Due to the explosive growth of smart phone technology, social media apps, internet banking, e-commerce, multimedia etc. will lead to higher increasing demand for data center services. In India, the rising demand for data center reliability, efficiency will lead to some challenges like lake of technical awareness, energy efficient solutions etc. Suman Pandey [18] proposed a perspective study on cloud load balancing. Here the author has considered static, dynamic, genetic, decentralized aware based load balancing algorithms. The most common examples are round robin, shortest job first, max-mean, two phase load balancing, power aware load balancing, throttled, honey bee, active clustering algorithm etc. There are so many challenges are appear in cloud load balancing like virtual machine migration, automated service provisioning, energy management, stored data management. Pratibha Pandey [19] proposed fuzzy logic based job scheduling algorithm for cloud environment. Here this paper mainly focused on selection of virtual machines which are eligible to execute the task properly. The classification of task is done on the basis of quality of services. These QoS parameters are Completion time and Bandwidth. Bheda and Bhatt [20] has proposed an overview of all load balancing techniques in cloud computing environment. Here the author focuses on various load balancing algorithm available in cloud computing. They gives a comparison on the throughput value, overhead, fault tolerance, response time, recourse utilization, scalability, performance etc. For this comparison he considered load balancing algorithm like round robin, dynamic round robin, PLBA, active monitoring, FAMLB, throttled,

active clustering load balancing algorithm etc. Er [21] has proposed online tuning of fuzzy inference system. Here the online tuning was done by using dynamic fuzzy Q-learning. Here an online self-learning algorithm was proposed. It mainly calculates the actions and Q-function in every iteration. If the temporal difference error value is higher than it will start adjusting the membership function and again calculate the action with Q-value. Ding et al. [22] proposed a method for task scheduling which is energy efficient. This is based on cloud computing technique. Basic key element is Q-learning which is used for reduce uncertainty and solve the related issue of cloud computing.

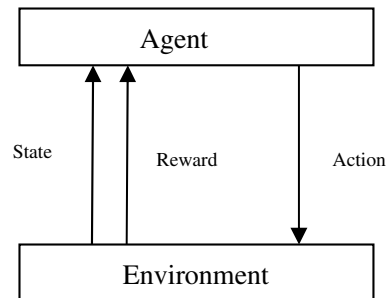
## 4 Reinforcement Learning

The most important features of artificial neural networks (ANN) are the ability to learn from the environment. The process of learning or training is required for making proper parameter adjustment. There are several types of learning process are available like supervised, unsupervised and reinforcement learning. In reinforcement learning process the exact information is not available properly. The learning based on this information is known as reinforcement learning. Here the aims is to find a proper actions, behavior or label which produce reward to maximize long term benefits.

Generally the RL doesn't have any prior knowledge about the environment. It only contains the information about valid set of actions with number of observations. By using repeatedly those actions it gains information about the corresponding environment. It will improve its policies automatically. The most common example of reinforcement learning is chess game. The block diagram of reinforcement learning is given in below (Fig. 4).

Here it performs like a closed loop. The output generated by the agent will affects the environment and similarly the environment will affects the agent. The environments will observe the current state of the system. Then the agent chooses the action from the controller policy depending upon the observations. It will affect to the environment. So, in this way one cycle will be completed and it will be repeated further to affects the next observations. After every cycle the agent will receives a reward value. Here the goal is to always choose the action with respect to the current state to

**Fig. 4** Fuzzy inference system





maximize the reward value. Reward is defines that improvement of the performance after applying the suitable actions. As reinforcement learning doesn't contain any prior knowledge, exploration and exploitation is most important part. Here exploration means the knowledge gained by trying new things. Similarly exploitation me knowledge gained by gathering more information. The most common approaches of reinforcement learning are SARSA and Q-learning.

### 4.1 Elements of Reinforcement Learning

This subsection describe the basic concepts of reinforcement learning which consists of three basic parameters that described below and its algorithm shown in Algorithm 1 and its related notations are shown in Table 1.

- (a) **Action:** The agent has to choose from set of actions to perform. By applying actions it will produce effects on the environment like state change.
- (b) **State:** It is the original scenario where the problem and solutions are carried out.
- (c) **Reward:** It was defined as the agent receives reward after taking some certain actions. Depending on the situation the reward value will be assigns to the system. If the system reach the goal, then the reward value will be higher else lower.

**Algorithm 1** Pseudocode of Q-learning

```

Step1: Initialize Q (st, at)
Step 2: Repeat
Step 3: Select an action "a" for state s & execute it
Step 4: receive immediate reward "r"
Step 5: Evaluate the value of new state
Q (st, at) ← Q (st, at) + α [ rt + I + γ max Q (st+1, a) - Q (st, at) ]
Step 6: Update value in q-table
s → s'
Step 7: End
    
```

**Table 1** Notation description

Notation	Description
<i>s</i>	State
<i>a</i>	Action
<i>a<sub>t</sub></i>	action taken at time t
<i>s<sub>t</sub></i>	State at time <i>t</i>
<i>r<sub>t+1</sub></i>	Reward at time <i>t</i> + 1
<i>α</i>	Learning rate set in between [0, 1]
<i>γ</i>	Discount factor also lies in [0, 1]

## 5 Proposed Method

In order to make data center scheduling suitable for a cloud QoS aware, a number of requirements have to be met which are given in below:

- (a) **Reliability:** The proposed system must be highly reliable. It has the ability to handle users request as many as possible.
- (b) **Scalability:** The system must be capable of scaling itself when more number of users and applications are arises. It must be stable when the user's requirements change. The user must be access recourse from the system easily.
- (c) **QoS and real time constraint:** The system must have advance QoS mechanism and policies to fulfill users demand.

### 5.1 System Model

This approach always focuses on onsite green power generation. In each time interval, the data center consumes energy from those renewable sources. The proposed approach is based on fuzzy q-learning where the arrival request will be forwarded to the most suitable data center. The selection of data center is done on the basis of information provided by the monitoring agent. Thus the workload will be done in minimum amount of time without any time complexity. Here the aim is to increase the process of load balancing with less time and maximize the utilization of renewable energy sources with lower electricity cost. Such that the throughput and the resource utilization will be maximized and the time complexity will be minimized.

This model consist of set of users transfer their request for the resource demand. The request will be received by the cloud service provider. Then it forwards the request to the cloud controller. Here the controller is based on fuzzy reinforcement learning. Here the monitoring agent will monitors different characteristics of the data center. It always deals with collecting the information and provides the data to both the fuzzy controller and knowledge learning components. For characteristics we have consider here workload, consumption of brown energy, utilization of renewable energy, and processor speed. By seeing the data provided by the monitoring agent, the controller chooses an action.

The fuzzy controller takes the observe data and generate the scaling actions. The learning components learn the appropriate rule and update the knowledge base continuously. After applying the action, system will generate the reward based on reward function. So reward means the performance improvement after applying the action. Then we observe the new state and reward. Here reward is calculated based on cost. Based on this information, we have to update the Q-table value. The important feature of Q-learning is that, it doesn't require any prior knowledge or detailed information about the system. Here if the fuzzy inference doesn't determine the scaling action, then the controller will randomly choose different actions and check whether it will produce the reward or not (Fig. 5).

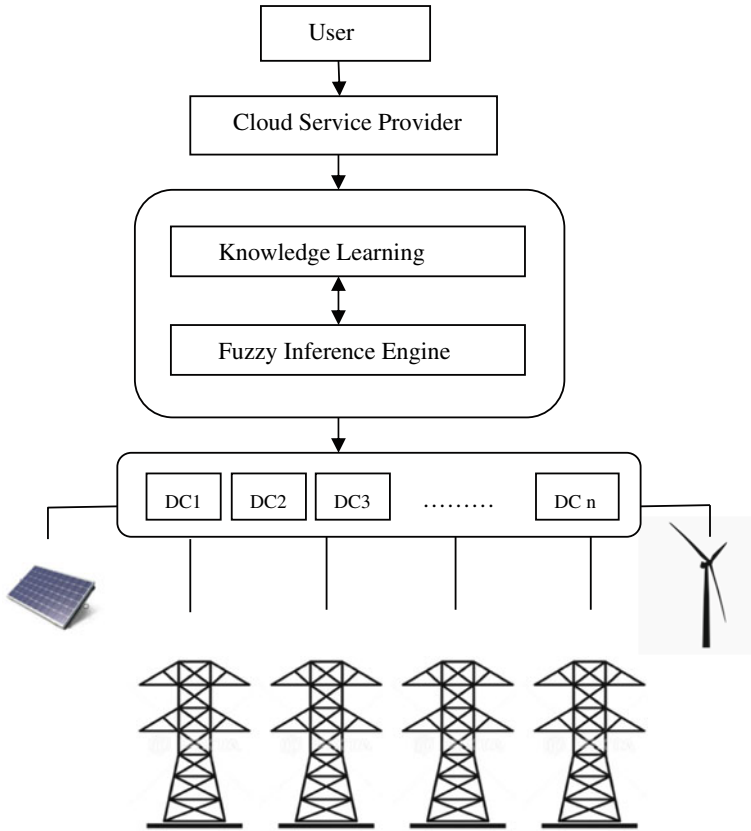


Fig. 5 System model

This model consists of two types of renewable energy sources like wind and solar energy. For wind energy, we have considered the wind turbine and for solar energy we have considered photo voltaic cell to convert it into electrical energy. It mainly focuses on the utilization of renewable energy sources. The fuzzy rule base also consists of several parameters which help to redirect the request to the suitable datacenter. So that utilization of renewable energy sources will increase. As a result the effect of brown energy sources will be minimized and it gives less bad impact on the environment. It also sees the cost of electricity at each data center based on the location. The data center which has less cost with maximum renewable energy sources will accept the request. Based on fuzzy q-learning every state of data center produces the reward. The data center which reward value is highest will accept the request from the user. Here the model consists of cloud controller, service provider, and set of users. For the onsite green power generation, here wind turbine and solar photo voltaic cells are considered. Here cloud controller is based on fuzzy logic.

## 5.2 Fuzzy Q-Learning (FQL)

In proposed approach, we used Q-learning as Reinforcement Learning approaches that we combine with the fuzzy controller. In this schema, a state 's' is modeled by four parameter ( $Ux, Bx, Lx, Sx$ ) for which an RL approach looks for best action 'a' to execute. The combination of the fuzzy logic controller with Q-learning, known as FQL, is explained in the following.

- (a) **Initialize the q-values:** Here the  $q$ -table contains set of actions with state pair. At the time of learning process each value will be updated by considering reward value. So here we have to set the  $q$  table value to '0' in initial stage.
- (b) **Observe the current state:** After initialization of the  $q$ -values, the current state of each data center will be monitored. The monitoring agent will continuously monitored the characteristics of data center like utilization of renewable energy, consumption of brown energy, available processor speed and assigned load.
- (c) **Select an action:** Here the control action is chosen by fuzzy logic controller. The fuzzy inference engine will observe each state value provided by the monitoring agent. Then it will process the value and generate the scaling actions. Here the scaling action means suitability of each data center. The fuzzy rule base only contains some rules defined by the user for some situations not for all. In that case the controller will automatically choose random actions and check whether it will generate the reward value or not.
- (d) **Calculate the control action inferred by fuzzy controller:** It determines the output values produced by the fuzzy inference engine. Here we have considered Sugeno fuzzy inference engine. The output value is a constant value which lies in between  $\{-2, -1, 0, 1, 2\}$ . The action can be any number of finite set of numbers. But for the simplicity we have consider here 5 possible actions depending on our problem.
- (e) **Calculate the reward value:** The controller receives the current values of state and actions. Here the reward value is calculated based on cost value. If the corresponding data center gives lower cost then it receives the reward otherwise the reward value will be zero.
- (f) **Calculate the value of new state:** After calculating the action value, it will calculate the value of new state i.e.  $V(s')$ .

$$V(s') = Q(st, at) + \alpha[rt + 1 + \gamma \max Q(st + 1, a) - Q(st, at)]$$

- (g) **Update the q-values:** After calculating the value of new state, it will update the corresponding state value in the  $q$ -table.

### Algorithm 2 Fuzzy Q-learning

Step1: Initializes the  $q$ -values to zero

Step2: Observes the current state  $s$

- Step3: Repeat
- Step4: Choose partial action ‘a’ from state s
- Step5: Computes the action ‘a’ inferred by fuzzy controller
- Step6: Receives reward
- Step7: Apply the action and observe the new state s’
- $V(s') = Q(st, at) + \alpha [rt + 1 + \gamma \max Q(st + 1, a) - Q(st, at)]$
- Step8: Updates the q-values
- $s \rightarrow s'$
- Step9: End

### 5.3 Fuzzy Rule Base

We have considered here 12 fuzzy rules for the fuzzy rule base. All the above fuzzy rules are in “if-else” format. Here we have considered four input variables for fuzzy inference which are given in Table 2.

- (a)  **$R_x$ : Utilization of renewable energy sources**  
 It defines the ratio of number of renewable energy sources used from the total number of renewable energy sources. It can be defined as

$$R_x = \frac{\sum R(x)}{R_{total}} \tag{1}$$

Here  $\sum R(x)$  is the total number of renewable energy sources used and  $R_{total}$  is the total renewable energy available.

**Table 2** Fuzzy rules to identify efficient data center

$R_x$	$B_x$	$P_x$	$L_x$	Suitability(Scaling action)
Low	Low	High	Low	Very high(2)
Low	Low	Low	High	Mid (0)
Low	High	High	Low	High (1)
Low	High	Low	High	Mid (0)
Mid	Low	High	Low	High (1)
Mid	Low	Low	High	Mid (0)
Mid	High	High	Low	Mid (0)
Mid	High	Low	High	Low (-1)
High	Low	High	Low	Mid (0)
High	Low	Low	High	Low (-1)
High	High	High	Low	Mid (0)
High	High	Low	High	Very low (-2)

(b)  **$B_X$ : Consumption of brown energy sources**

It is defined as the amount of brown energy consumed from with respect to total available renewable energy.

$$B_x = \frac{\Sigma B(x)}{R_{\text{total}}} \quad (2)$$

Here  $\Sigma B(x)$  is the total amount of brown energy used.

(c)  $P_X$ : Processor speed

(d)  $L_X$ : Assigned load at the data center.

The output value produced by fuzzy inference is defined in terms of “suitability”. But for fuzzy q-learning, we have considered as scaling action. Here we have taken some constant value in between  $\{-2, -1, 0, 1, -1\}$  for easier evaluation. It can be any finite number.

For example- IF  $U_x$  is low,  $B_x$  is low,  $L_x$  is low and  $SX$  is high THEN “sa = + 2”.

#### 5.4 Reward Calculation

After finding proper action given by fuzzy inference system, the next job is to apply the action and evaluate the next state. Then it will checks that whether this action produce the reward or not. The reward calculation is based on cost.

$$\text{Reward} = \mathbf{0}, \text{ if cost} > 0.5$$

$$1, \text{ if cost} < 0.5$$

Here cost is calculated by amount of brown energy consumed by the data center \* electricity price per each unit.

$$\text{i.e. Cost } (F_x) = B_x * \text{electricity price per each unit} \quad (3)$$

#### 5.5 New State Evaluation

The monitoring agent will monitor the state of each data center. If the current state scaling action is greater than 0, the new state will be calculated as the addition of both current state with the corresponding action value. If scaling action is less than 0, the new state will be same as the previous state.

$$\text{New state} = x + u, \text{ if } u > 0 \ \& \ x > 1$$

$$x, \text{ if } u \leq 0 \ \& \ x \leq 1$$

Here  $u$  = action,  $x$  = current state.

## 6 Result and Discussions

For the implementation of fuzzy rules, we used “MATLAB”. It provides a fuzzy tool box system. The tool box contains the list of inputs, output and types of fuzzy inference engine. Here we take four inputs i.e.  $R_X, B_X, P_X, L_X$  and one output is the suitability of each data center in form of scaling actions. For every input and output, we have used triangular membership function. The name of the inference engine is fuzzy load balance.

The inference engine is based on “fuzzy Sugeno inference engine”. It is the most popular inference engine as it has less complexity. In fuzzy inference engine all IF–THEN rules are defined by the fuzzy set. For different input, we have taken different membership function to define whether it is in the range of high, medium, low etc. The below figure will show the design of our fuzzy inference system (Fig. 6).

### 6.1 Fuzzy Sugeno Inference Engine

It is mostly suitable for mathematical analysis. Sugeno inference always gives output that is either constant or a linear (weighted) mathematical expression. Sugeno-type

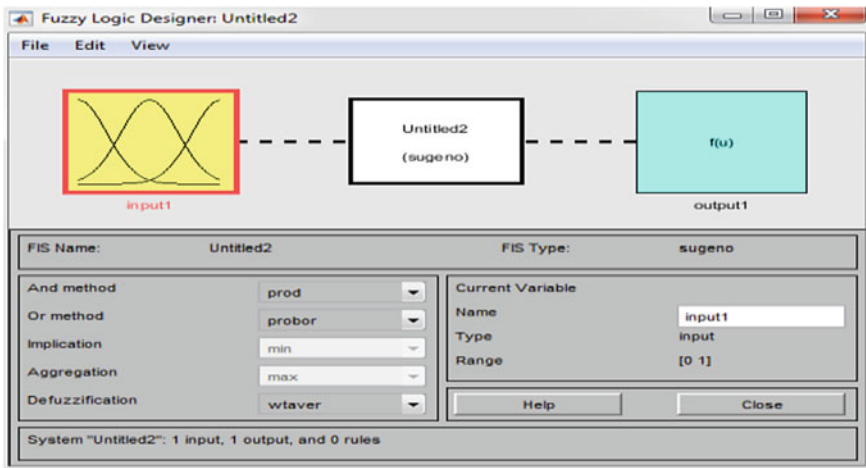
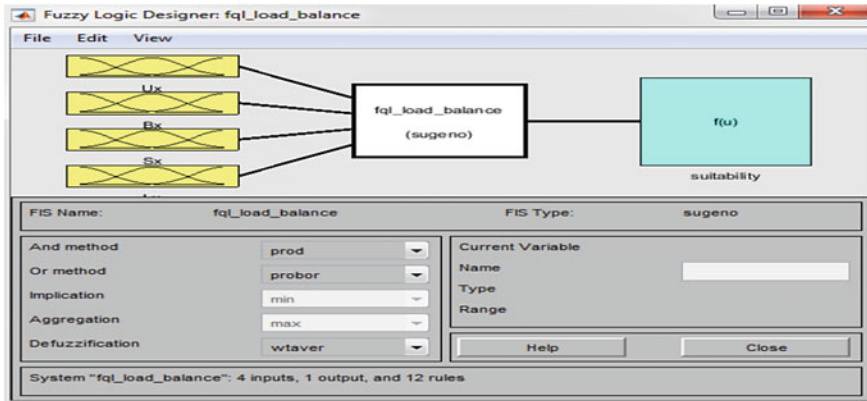


Fig. 6 Fuzzy logic toolbox



**Fig. 7** Fuzzy Sugeno Inference

FIS uses weighted average to compute the crisp output. This is mostly helpful in optimization and adaptive techniques. Sugeno FIS has some advantage over Mamdani FIS which are given in below:

- (1) It is computationally more efficient.
- (2) It works well with optimization and adaptive techniques.
- (3) It is so convenient to mathematical analysis.

Example- If  $A$  is  $X_1$  and  $B$  is  $X_2$  then  $C = ax_1 + bx_2 + c$  (linear expression) where  $a$ ,  $b$  and  $c$  are constants (Fig. 7).

### 6.1.1 Membership Function for Input and Output

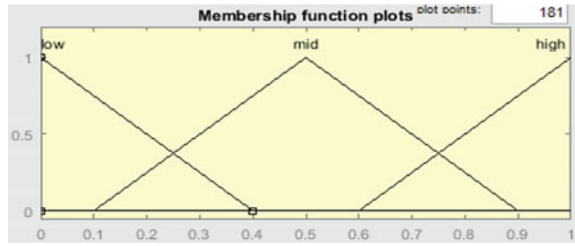
For the design of inference engine we have used mat lab toolbox which is given in below. Here the membership functions are in the form of triangular shape function.

This function is defined by a lower limit  $x$ , an upper limit  $y$ , and a value  $m$ , where  $x < m < y$ . It is easier to evaluate the membership value in compare to other membership function.

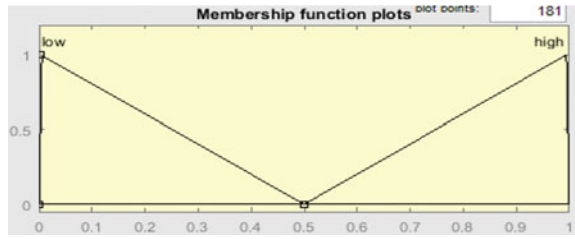
- (a) Utilization of Renewable Energy  
This input variable is denoted by  $U_x$ . It defines the ratio of the number of renewable energy sources used to the total number of renewable energy sources. Here the value of membership functions lies in between  $[0, 1]$ . This membership function has divided in to three types i.e. low, mid, high (Fig. 8).
- (b) Consumption of Brown Energy  
It is denoted by  $B_x$ . It defines how much brown energy sources will consumed for the electricity. Here the values of membership functions lie in between  $[0, 1]$ . This membership function has divided in to two types i.e. low and high (Fig. 9).



**Fig. 8** Membership function of utilization of renewable energy

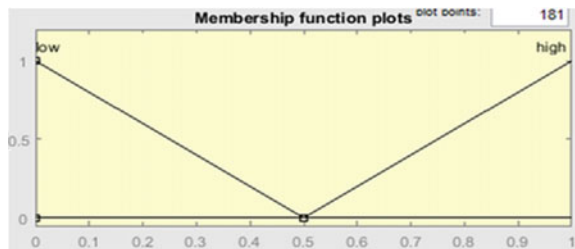


**Fig. 9** Membership function of consumption of brown energy

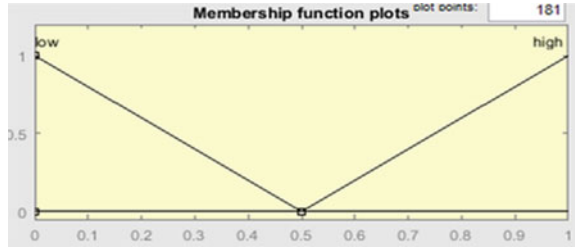


- (c) Processor Speed  
This input variable is denoted by  $Sx$ . It defines the speed of the processor. Here the values of membership functions lie in between  $[0, 1]$ . This membership function has divided in to three types i.e. low, mid, high (Fig. 10).
- (d) Assigned Load  
This input variable is denoted by  $Lx$ . It defines the assigned load of the given data center. Here the values of membership functions lie in between  $[0, 1]$ . This membership function has divided in to two types i.e. low, high (Fig. 11).
- (e) Suitability of the Data Center (Scaling Actions)  
This input variable is denoted by “suitability”. It is the output function produced by the fuzzy inference system. Here the values of membership functions lie in between  $[-2, -1, 0, 1, 2]$ . This membership function has divided in to four types i.e. very low, low, mid, high, very high. If the output values lies in between  $[-2, -1, 0, 1, 2]$ , then it will be considered as in given format (Fig. 12).

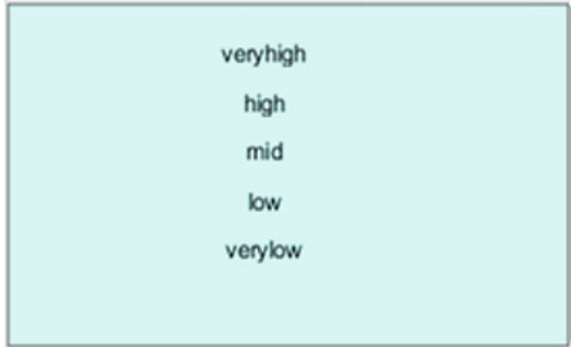
**Fig. 10** Membership function of processor speed



**Fig. 11** Membership function of assigned load



**Fig. 12** Linguistic variables for output parameter



- [−2] – Very low
- [−1] – Low
- [0] – Mid
- [1] – High
- [2] – Very high

### 6.1.2 Rule Base

The rule base consists of 12 rules by using the input variables to determine the output value or the scaling actions (Fig. 13).

### 6.1.3 Output of Fuzzy Inference Engine

We have taken an example to produce the output. The values of every parameter are given in the figure. The data center which consists of higher suitability value will accept request from the user for execution (Fig. 14).

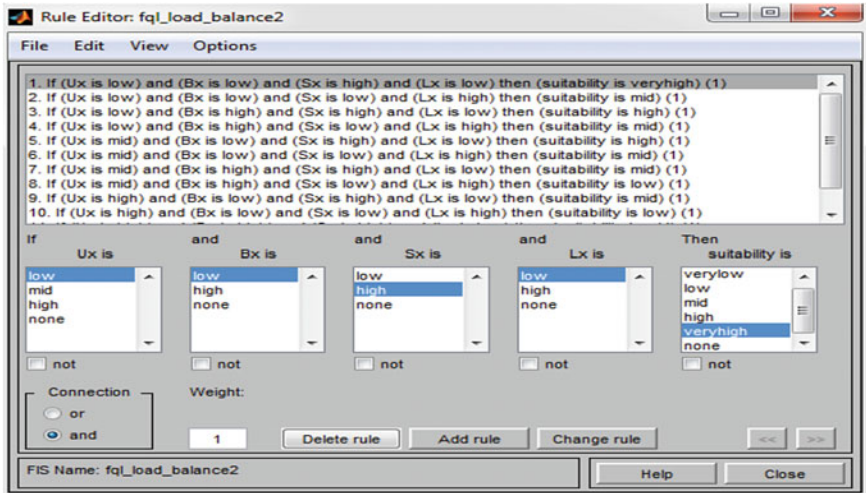


Fig. 13 Rule base for the proposed method

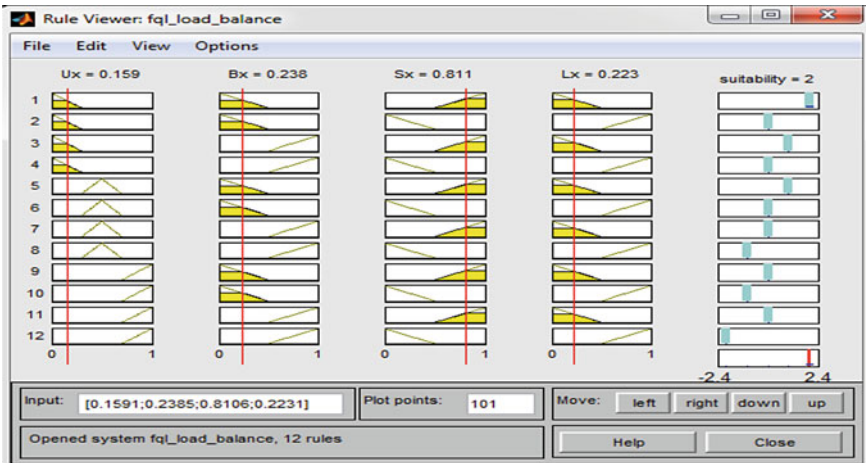


Fig. 14 Validation of fuzzy inference system

### 6.2 Result Comparison of Fuzzy and Fuzzy Q-Learning

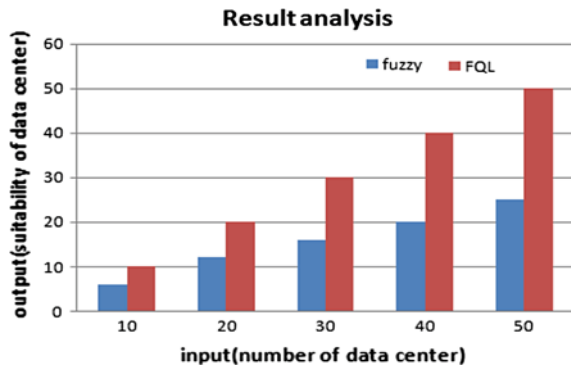
In Table 3, comparison is done in between fuzzy and fuzzy Q-learning. The inference system is based on Mamdani and Sugeno inference engine.

From the below comparison fuzzy q-learning gives batter result in compare to simple fuzzy inference system (Fig. 15).

**Table 3** Comparison between fuzzy and fuzzy Q-learning

	FIS	Input	MFs	Reward calculation
Fuzzy	Mamdani	RX	3	N/A
		BX	2	
		SX	2	
		LX	2	
		FX	3	
FQL	Sugeno	RX	3	Based on cost
		BX	2	
		PX	2	
		LX	2	

**Fig. 15** Result analysis of fuzzy and FQL

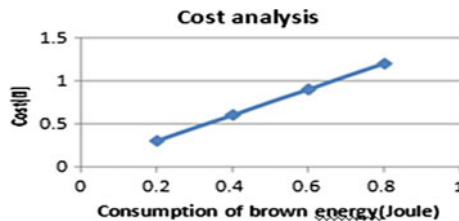


### 6.2.1 Comparison of Cost with Respect to Utilization of Renewable Energy

Here the comparison is done on the basis of cost value. Cost value is calculated based on consumption of brown energy from each data center as given above on Eq. (3) (Fig. 16).

As a result if the consumption of brown energy will increases then the cost of that corresponding data center will be also increased. So that it is important to choose the alternate of brown energy which is known as renewable energy sources.

**Fig. 16** Result analysis of Fuzzy and FQL



## 7 Conclusion

Fuzzy q-learning based algorithm for knowledge evolution is proposed here. This is based on fuzzy inference and Q-learning. This approach provides a non-linear mapping from the inputs like processor utilization of renewable energy, consumption of brown energy, electricity cost, speed of the processor, assigned load on data center to an output showing the appropriateness of the data center for the request redirection. This can be done by using some fuzzy rule which make this approach simple with out of any complexity. The Q-learning approach helps to learn each rule and update it in knowledge base at runtime. In future work we will try to implement this using cloud analyst software.

## References

1. Khayer A, Talukder MS, Bao Y, Hossain MN (2020) Cloud computing adoption and its impact on SMEs' performance for cloud supported operations: a dual-stage analytical approach. *Technol Soc* 60:101225
2. Gill SS, Tuli S, Xu M, Singh I, Singh KV, Lindsay D, Tuli S, Smirnova D, Singh M, Jain U, Pervaiz H (2019) Transformative effects of IoT, Blockchain and artificial intelligence on cloud computing: evolution, vision, trends and open challenges. *Internet of Things: 100118*
3. Lin HC, Kuo YC, Liu MY (2020) A health informatics transformation model based on intelligent cloud computing—exemplified by type 2 diabetes mellitus with related cardiovascular diseases. *Comput Methods Programs Biomed* 191:105409
4. Kim T, Min H, Choi E, Jung J (2020) Optimal job partitioning and allocation for vehicular cloud computing. *Future Gener Comput Syst* 108:82–96
5. Toosi AN, Buyya R (2015) A fuzzy logic-based controller for cost and energy efficient load balancing in geo-distributed data centers. In: 2015 IEEE/ACM 8th international conference on utility and cloud computing (UCC). IEEE, pp 186–194
6. Dey N (ed) (2017) *Advancements in applied metaheuristic computing*. IGI Global
7. Dey N, Ashour AS (2016) Antenna design and direction of arrival estimation in meta-heuristic paradigm: a review. *Int J Serv Sci Manage Eng Technol (IJSSMET)* 7(3):1–18
8. Das SK, Tripathi S (2019) Energy efficient routing formation algorithm for hybrid ad-hoc network: a geometric programming approach. *Peer-To-Peer Netw Appl* 12(1):102–128
9. Das SK, Tripathi S (2018) Adaptive and intelligent energy efficient routing for transparent heterogeneous ad-hoc network by fusion of game theory and linear programming. *Appl Intell* 48(7):1825–1845
10. Das SK, Tripathi S (2018) Intelligent energy-aware efficient routing for MANET. *Wirel Netw* 24(4):1139–1159
11. Chatterjee S, Sarkar S, Dey N, Ashour AS, Sen S, Hassanien AE (2017) Application of cuckoo search in water quality prediction using artificial neural network. *Int J Comput Intell Stud* 6(2–3):229–244
12. Singhal U, Jain S (2014) A new fuzzy logic and GSO based load balancing mechanism for public cloud. *Int J Grid Distrib Comput* 7(5):97–110
13. Jamshidi P, Sharifloo AM, Pahl C, Metzger A, Estrada G (2015) Self-learning cloud controllers: fuzzy q-learning for knowledge evolution. In: 2015 international conference on cloud and autonomic computing (ICCAC). IEEE, pp 208–211
14. Arabnejad H, Pahl C, Jamshidi P, Estrada G (2017) A comparison of reinforcement learning techniques for fuzzy cloud auto-scaling. In: 2017 17th IEEE/ACM international symposium on cluster, cloud and grid computing (CCGRID). IEEE, pp 64–73

15. Pasha N, Agarwal A, Rastogi R (2014) Round robin approach for VM load balancing algorithm in cloud computing environment. *Int J* 4(5):34–39
16. Dale M (2013) A comparative analysis of energy costs of photovoltaic, solar thermal, and wind electricity generation technologies. *Appl Sci* 3(2):325–337
17. Raje S, Maan H, Ganguly S, Singh T, Jayaram N, Ghatikar G, Greenberg S, Kumar S, Sartor D (2015) Data center energy efficiency standards in India. In: *Proceedings of the 2015 ACM 6th international conference on future energy systems*. ACM, pp 233–240
18. Pandey S (2017) Cloud load balancing: a perspective study. *Int J Eng Comput Sci* 6(6)
19. Pandey P, Singh S (2017) Fuzzy logic based job scheduling algorithm in cloud environment. *Comput Model NEW Technol* 21(3):25–30
20. Bheda H, Bhatt H (2015) An overview of load balancing techniques in cloud computing environments. *Int J Eng Comput Sci* 4:9874–9881
21. Er MJ, Deng C (2004) Online tuning of fuzzy inference systems using dynamic fuzzy Q-learning. *IEEE Trans Syst Man Cybern B (Cybern)* 34(3):1478–1489
22. Ding D, Fan X, Zhao Y, Kang K, Yin Q, Zeng J (2020) Q-learning based dynamic task scheduling for energy-efficient cloud computing. *Future Gener Comput Syst* 108:361–371