SELECTING OPTIMAL LINK IN FOG INFRASTRUCTURE USING TEACHING LEARNING BASED OPTIMIZATION (TLBO)

Rupinder Kaur
CEC, Landran (Mohali), India

Amanpreet Kaur
Assistant Professor(IT)
CEC, Landran (Mohali), India

Abstract- The optimization work conducted here is based on Teaching Learning Based Optimization (TLBO). This method optimizes multiple competing parameters to obtain an optimal value which reflect the current state of a particular link from which a fog device may interact for either sending or rendering data from a cache. The paper illustrates mesh topology with which the fog infrastructure works closer to the rare life scenarios. The algorithm computed the cost of sending and receiving cache data in the Fog Zone using three parameters (Bandwidth, download rate and distance of the best route) that help us understanding the behavior of the Fog devices.

Keywords- TLBO (Teaching Learning Based Optimization), optimization, caching, fog device.

I. INTRODUCTION

The current business systems should work with multiple network protocols, multiple data formats, and exchange standards. The system should be able to work with the heterogeneous arrangement of devices which may have different levels of energies and constraints. Then it also needs to work with multiple sensing technologies whose output may be analog in nature or it may require modulator boosters, repeaters for signal conditioning and propagation. One of the alternative approaches suggested by [1] is 'Caching' in the context of fog computing devices. Conventionally, content delivery networks have been used in cloud computing. Now, caching technology [2] has now entered in Telco CDN and mobile technologies. All this is mainly due to stream video traffic growth and the emergence of new kind of mobile appliances in business and gaming. These applications are time sensitive and have a high degree of Quality of Service (QoS) specification to meet customer quality of experiences.

Telco CDNs: Content Delivery Networks (CDNs) relieve network segments from overloading, speed up delivery, reduce packet loss and ultimately improve the quality of internet experience for subscribers. CDN providers use telecom operator networks to serve a large number of customers including content providers, large industries and telecom operators themselves. The network of highly distributed caches often referred to as a telco CDN [3], is a network enabler that saves a huge amount of traffic on the links toward the Internet by handling user requests locally without having to retrieve data from the origin server. The caching algorithm [4] decides which objects to cache, and the cache collaboration strategy, at times in combination with preloading, determines how cacheable items are propagated throughout the telco CDN as shown in Figure 1.

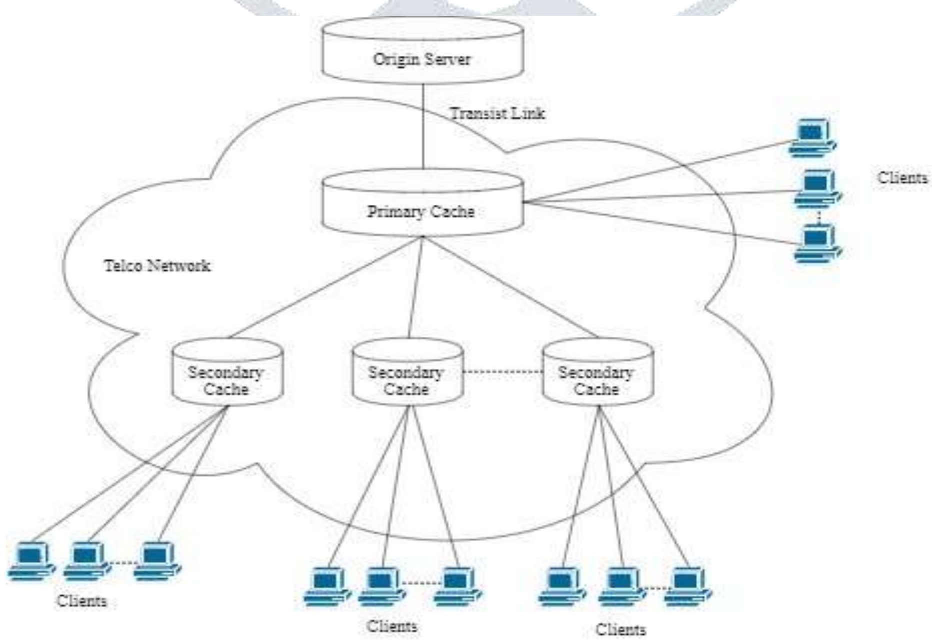

Figure 1: Telco Network [3]

CDN's in Mobile:- A mobile content delivery networks or mobile content distribution networks (Mobile CDN) [5] is a network of servers - systems, computers or devices - that cooperate with each other to maintain the delivery of content to subscribers on any type of wireless or mobile network. Mobile devices such as smartphones, medical/health devices, gaming devices have constraint memory and computing power. Delivery of such content to such devices needs special attention in terms of encoding, a process in and out coming to a live stream of data and performance. For sustaining good turnaround time in such content and response delivery mobile technology operates many follow one of the following strategies:-

- **a)** Local Cache- When a device stores frequently used files in its local storage medium for improving the response time is called local caching. Local storage medium might be a file system or maybe a compressed file. The local storage may also support data aggregation [6] and data buffering. Some devices specialized DBMS [7] such as pouch DB.
- **b)** Peer-To-Peer Caching [8]- When caching is done in mutually agreeing peer storage medium.
- **c)** Edge Caching- The amount of content required by service providers is growing significantly, so it is not possible to cache all the content in the cloud server, which also increases the end to end delay. To solve this problem some periphery traffic is stored in edge device can significantly reduce burden. In edge devices, the edge caching is applied to reduce the traffic burden on the cloud server and provide the fast content access and also improve performance.
- **d)** Server Caching: When caching is done on cloud servers or network having a client-server architecture, the network may have primary and secondary servers [9].

In the case of video content, these services include device detection operations, video transcoding, image rendering, and bit-rate adaptation. For the maintenance of the delivery of other content, mobile CDNs use technologies such as caching. With currently available mobile CDN solutions, nodes are deployed at the edge of the network and in multiple locations, often over multiple backbones directly connected with Mobile Network Operators (MNOs) [5]. These nodes cooperate with each other to satisfy requests for content by subscribers. The benefits of maintaining such form are reduced bandwidth usage, improved end-user performance, or increased global availability of content over a mobile network.

Optimization methods: It is the process of updating the system for increasing the efficiency or for using fewer resources.

1. Single objective optimization [10]: It is a standard optimization with a single objective function.
2. Multi objective optimization [11]: When mathematical optimization procedural problems are required for optimizing more than one objective function at the same time then it is called multi objective optimization. It is also known as multi-attribute optimization.
3. Heuristic optimization [12]: It is a technique which is designed for solving a problem when classic methods seem to be too slow.
4. Stochastic optimization [13] : It is the process of increasing or decreasing the value of a mathematical function in a case when the input parameters are random.
5. Parametric optimization [14]: It is a type of mathematical optimization, in which problem is solved as a function of one or multiple parameters.
6. Non parametric optimization [15]: It gives a method to define a design space in regions without the process of defining the problem in design parameters.
7. Linear optimization [16] : It is a method to attain the best results in a mathematical model whose requirements are represented by linear relationships.
8. Non Linear optimization: It is used to find the maximum or minimum value of a function.

II. RELATED WORK

Siming Wang et al. [1] Presented a technique named as 'CachinMobile' that can considerably improve energy efficiency, service strength and reduce latency. Traditional cloud computing gradually fails to fully meet the requirements of ever-growing mobile services due to potential problems, such as high service latency, lack of location-awareness. The implemented caching strategy considered genetic algorithm for optimization which taken $\lambda_{i,j}$(contact rate), R_n(Popularity distribution of contents), L_i (geographic location of u_i), L_j(geographic locations of u_j), M (Input Population size), T(Termination generation), P_{mut} (probability of mutation), P_{cro}(probability of crossover) as inputs and produced $X \in \{0,1\}^{K\times N}$ (Total transmitting energy optimization) output which shows significant improvement to reduce transmitting energy consumption.

Nidhi Jain Kansal et al. [17] conducted a review of all the methods, techniques used in the cloud, small and large scale distributed networks for minimization of resource utilization for increasing energy efficiency, service response time, performance and reduce carbon emissions.

Xiao Chen et al. [18] designed layer network architecture for enhancement of vehicular network. Two dynamic scheduling algorithms were proposed in the fog computing scheme for the data scheduling in vehicular networks. The first algorithm was Dynamic Scheduling with Queue Length taken J (set of tasks) and S (set of vehicular servers) as inputs and produced $A_{j,s}$ (schedule a task $j \in J$ to a server $s \in S$) as output. Second algorithm was Dynamic Scheduling with Response Time also taken J (set of tasks), S (set of vehicular servers), $N_{s,t}$ (number of servers at terminals) and K (number of vehicles) as inputs and produced $A_{j,s}$ (schedule a task $j \in J$ to a server $s \in S$) as output. Vehicular data scheduling policy used for better efficiency. For performance analysis, a compositional formal method, named PEPA (Performance Evaluation Process Algebra), was applied to model scheduling algorithm in a fog-based vehicular network and findings revealed that the algorithm was numerically stable and helped in increasing efficiency dynamically in terms of better scheduling of workloads.

Victor Shnayder et al. [19] built Power TOSSIM which is an extension of TOSSIM, a scalable simulation environment for wireless sensor networks that provides an accurate, per-node estimate of power consumption and obtains an accurate estimate of CPU cycle counts for each mote by measuring basic block execution counts and also mapping each basic block to microcontroller instructions. Further, they did CPU profiling and power state of sensors and other devices. Trace-based simulation experimentation was done and exhibits little overhead.

Amanpreet Kaur et al. [20] used 'Particle Swarm Optimization' for load balancing in cloud or fog. The authors had used cloudSim simulator for demonstrating their work and optimization had been carried out for throughput and energy. Finally, results revealed that work was compared with the centralized mechanism of load balancing and showed lower energy consumption and increased throughput capacity with PSO.

Dolly et al. [21] describe the study which involves the surveying of various task scheduling algorithm and resource allocation algorithm for cloud. By making comparison among various algorithms, author concludes that scheduling algorithm depends on type of task to be scheduled. Depending on surveying, Time driven based resource allocation gives better response time and increase resource utilization. Finally, results revealed that, the various algorithms make span can be reduced by grouping the task. Since cloud computing systems have a high degree of unpredictability with respect to resource availability.

Gilsoo Lee et al. [22] introduced the framework of online optimization as a powerful tool for operating distributed fog computing networks in dynamic and uncertain environments. In particular, author discussed three key applications of online optimization for fog computing. In conclusion, online optimization tools are expected to play a key role in future fog computing networks primarily due to the need for low latency operation and the online models that capture the dynamic and uncertain environments.

Dilip Kumar et al. [23] Used Teaching Learning Based Optimization (TLBO) to solve the MOVMP problem. In this paper, author represent the optimized virtual machine placement optimization approach for a cloud data center to reduce the total energy consumptions with better VMP algorithm using TLBO. The simulation based result validates the effectiveness of their approach compared to the other approaches.

R. Venkata Rao [24] proposed specific parameter-less algorithm named as 'Jaya algorithm'. TLBO algorithm has two phases (i.e. teacher phase and the learner phase) but the proposed algorithm has only one phase and it is simpler to apply. Jaya algorithm is implemented on 24 well defined constrained optimization problems having different characteristics. Finally the results obtained by Jaya algorithm are compared with the results of well-known optimization algorithms. Results revealed the satisfactory performance of Jaya algorithm for the constrained Optimization problems.

Dr. Anantkumar J. Umbarkkar et al. [25] performed the experiment on the programming parallel TLBO (OpenMP TLBO) on multi-core system for unconstrained optimization. Finally results revealed that, the OpenMP TLBO gives linear speedup and optimizes the CPU utilization against the sequential implementation of TLBO (STLBO).

III. THE SCOPE OF WORK

In a typical scenario where a fog device sending or receiving data. There are always multiple parameters playing a role which can be which can be represented in a mathematical form and mathematical relations can be inferred from the behavior of these parameters. It is abundantly clear that higher the payload, higher is the energy consumption and lower the bandwidth, lower is the response time. So this culminates into a scenario where multiple parameters compete with each other and there is no best metric value based on which decision can be taken. Therefore the role of optimization algorithm becomes important. Therefore the scope of this paper is to build an algorithm that optimizes these competing parameters to obtain best metric value.

IV. METHODOLOGY

This section explains the architecture of optimization model for identification of the routing path for getting most optimal values for resource management.

Experimental setup and tools:

For this research work, CloudSim [26], iFogsim [27] and **weka customize API** used to simulate the conditions im fog environment.

The purpose of implementation of TLBO is to find an optimal value as per following conditions:

1. If the value find by the TLBO is close to the Target QoS then, ssend the request to that particular link.

$$\text{Target QoS}=\begin{bmatrix} TH: & 10 & 10\;11 & 10 & 9\;\;12 \\ \text{DL}: & 0.1 & 0.10.1 & 0.2 & 0.20.2 \\ \text{PW}: & 50 & 52\;50 & 55 & 45\;48 \end{bmatrix}$$

2. The Quality or health of the link is represented with the mathematical relationship between the design variables.

$$\text{Resource Monotoring} = \begin{bmatrix} \text{TH}_1 & \text{TH}_2 & \text{TH}_3\,\text{TH}_4 & \text{TH} & \text{TH}_6 \\ \text{DL}_1 & \text{DL} & \text{DL}_3\,\text{DL}_4 & \text{DL}_5 & \text{DL}_6 \\ \text{PW}_1 & \text{PW}_2 & \text{PW}_3\,\text{PW}_4 & \text{PW}_5 & \text{PW}_6 \end{bmatrix}$$

3. Data representation of possible solution

4. How much values can be changed, Learning rate=0.4

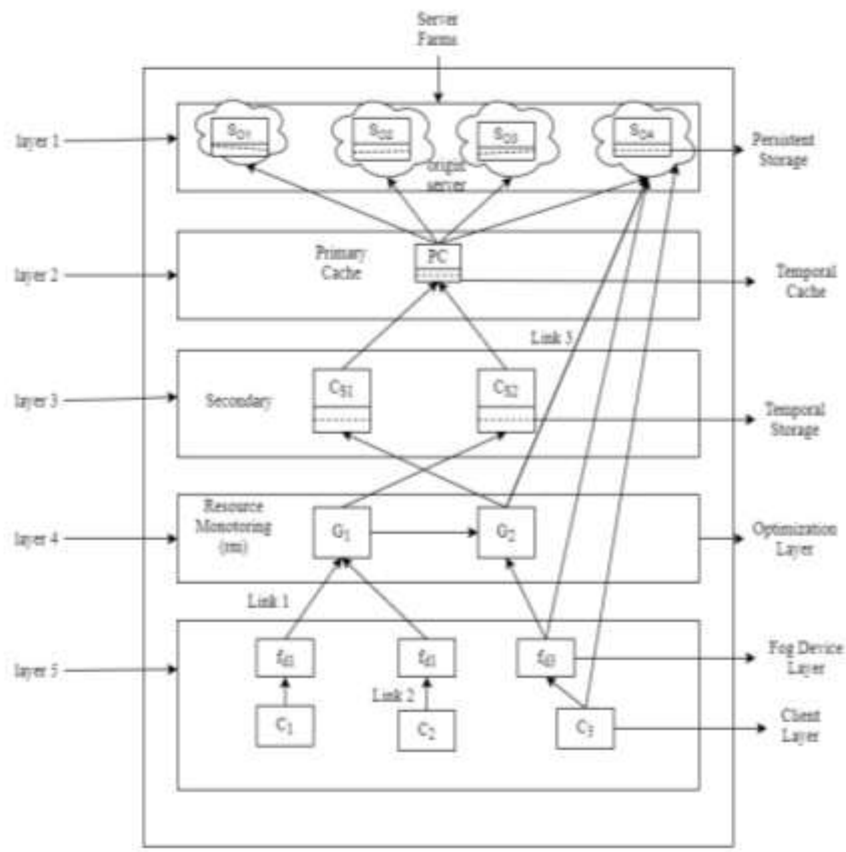

Figure 2: Network Layout

EXPLANATION OF NETWORK LAYOUT

(a) **Layer 1:** This layer represents the original data location. It consists of server farms (A server farm consists of computer servers which supply server functionality more than the capability of a single machine. Server farms have backup servers, which can take the place of primary servers in case of a primary-server failure) which has multiple servers having multiple applications running in virtualized environment. It is from the layer a fog device may try to retrieve data or send for persistent storage (Persistent storage is data storage device which holds the data after the device is shut off. It is also referred to as non-volatile storage).

(b) Layer 2: The second layer is created for doing temporal storage (temporal storage is a part of computer memory which is utilized to store the data for temporary purpose that waits processing. The perfect example is buffer). It is also called as interim storage to act as primary cache (A Primary cache is built right into the CPU. Primary cache is very faster cache, but it is smaller than secondary cache, whereas secondary cache is much faster than main memory. It is also called as internal cache, level 1 cache, and on-chip cache.) for cluster of servers in data farm. The primary cache has a data which represents most frequently used data for the clients.

(c) Layer 3: As the network size increases and also the span of fog devices increases. The primary cache may further be divided into secondary caches. So that, there is little jitter/delay.

(d) Layer 4: This layer consist of routing servers that have inbuilt resource monitoring (Resource Monitoring displays information about the usage of hardware i.e CPU usage, memory usage; disk space occupied, and network usage and software resources in real time) and optimization algorithms(Optimization algorithms are used to find the minimum values of mathematical functions in engineering. They are also used to evaluate design tradeoffs, and to find patterns in data) for guiding the fog device clients.

Table1: Parameters Used in Implementation

Application models	Sense → process → actuate, streaming
Resource management	Resource provisioning, operation management, scheduling
Fog infrastructure	Monitoring, performance, power monitoring
Data	HTTP Live request/response
Fog devices	Routers, virtual LANs

Network topology	Edge (Json standard based file)
IOT Device (e.g 4 G enabled Kindle)	Content /Data Aggregation/ View

(e) **Layer 5:** Fog computing is an architecture that uses one or more collaborative end-user clients devices to carry out a huge amount of storage rather than stored on primarily in cloud data centers. Gateways: Gateway is a computer program and a link between two computer programs that allow each other to share information and bypass certain protocols on a host computer. TLBO (Teaching Learning Based Optimization)

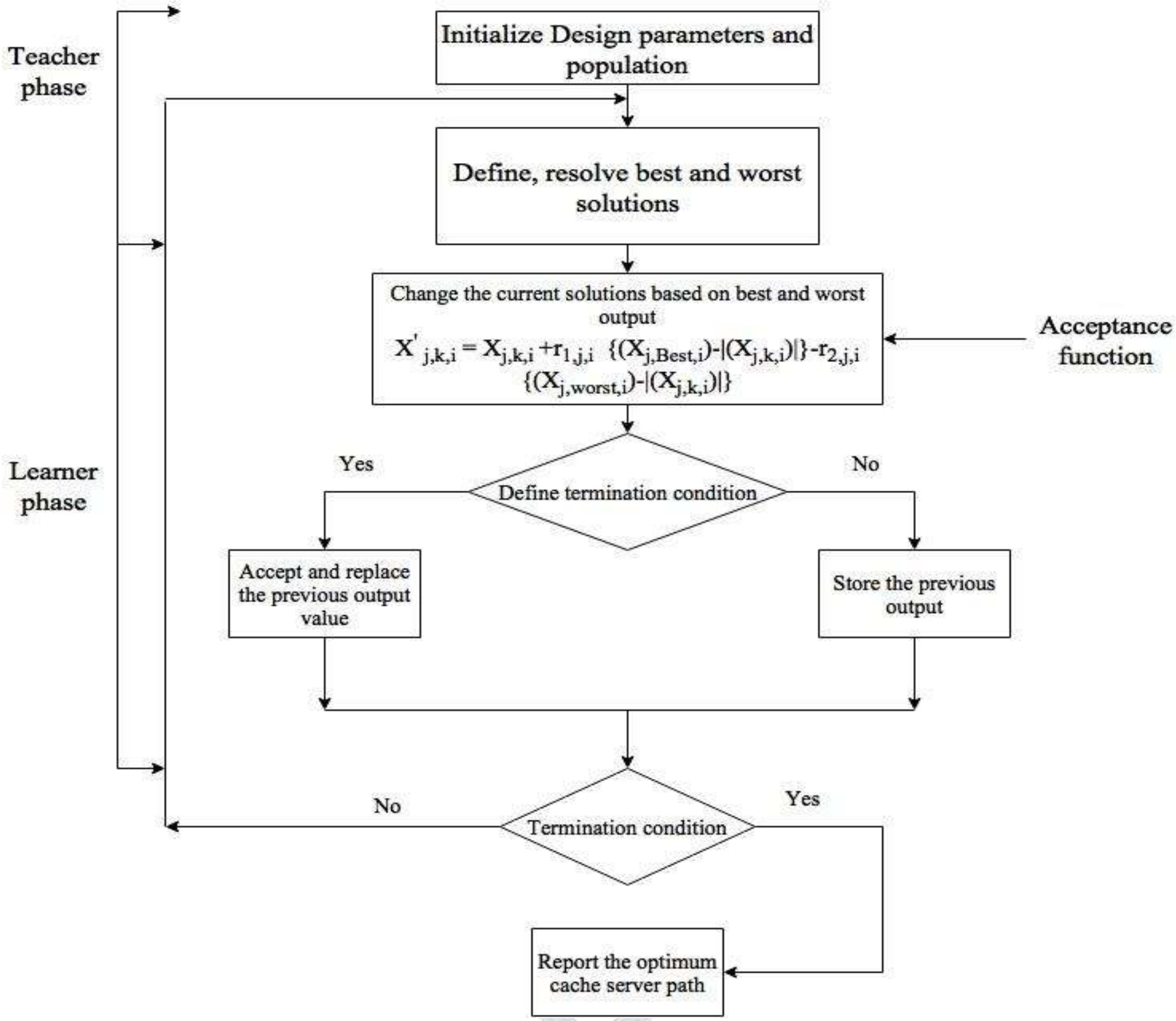

Figure 3: Flowchart of MTLBO Algorithm

Step 1: The number of design variables for the problem are four, these are inputs and one is the output variable. These design variables help the algorithm to obtain the most optimal value close to QoS metric. The design variables are TH, DL, PW and respective QoS metric value that reflects the optimal value at given time.
Definition of the objective function: The objective function is a mathematical representation of the design variable at which these variable values are closest to best QoS. For this the relationship between these variables is to find by using the curve fitting method that helps us to obtain the intersection point with respect to the QoS metric. It sets that standard or the benchmark that qualifies of all the parameters that influence other factors such as errors, dropped packets and out rages

Step 2: Since , the current algorithm is based on the concept of finding 'Best', solution verses the 'previous' solution or the worst solution as shown in the Figure 3.

Step 3: The Best solution is analogous to the find best team of Teacher-Student, who produce maximum score or marks. The idea is to find the best team having cumulated score 'x'. The cumulated score is sum of all the scores/ marks of students, and can be represented / or expressed as

$$\frac{\exp(C_{score} - P_{score})}{Total\ score}$$

Where C_{score} is the current score and P_{score} is the previous score.

In our context, the marks of the group can be considered on the basis of the aforesaid parameters and mathematical relationship mentioned in the section.
Primary, the smaller the rate of change in the 'marks' or (solution output), higher the marks, the more likely it is for the algorithm to accept the final output as the best solution.

Teaching Learning Based Optimization Algorithm Steps:

1. This step is to set the initial design parameters and initial marks and generate random initial output solution.
2. Then, the iteration begins, and it work till the condition is fulfilled. This is the time the maximum chances of improvement have been exhausted, and maximum possible marks have been achieved. It consist of two phases:

   Teacher phase: It is the first phase, in analogous to the learning from the teacher. And due to the efforts of the teacher the class mean score /marks increase.
   Learning phase: Learning by the students for improvement.

3. In each iteration, we select a neighbor by making a small rate of change to current output value. The next neighbor is the next 'cache server' in the fog network.
4. Then, we decide to move the coordinates of the next neighbor solution (fog cache).
5. Finally, we continue to decrease the 'maximum marks, so that maximum possible solution can enter the reach space and finally compare the iteration.

INITIALIZATION :

```
Deploy , Nodes in the Vector Scope model.
Initialize , Default Mean Score .
Initialize , Mean Score Rate Change .
Initialize  , Solution  Population
Teacher Phase pseudo code :
        For each subnet in the Fog Network
        For each 'cache server' in Network
                Compute  "Cost /quality of link / mean performance "
                Assign , Cost score .
        End
End

 Learner  Phase pseudo code :
Initialize the Fog Resource Monitor .
        Register , Fog Node Statistics,
For each "Cache " in the FogNetwork ,
        Compute the Difference between the Previous and Current Performance Statistics
        Execute Acceptance Function

        If (AcceptanceFlag ==True )
                        If Check Termination Condition = True
                                AcceptValue , Cache Server IP
                                Else
                                Continue ,
                        End
                        Else
                        Continue ,
        End if
End For
```

V. RESULTS

The evaluation of the algorithm is done based on the accuracy of the algorithm to select the right cache server from an array of servers available in the network.

Table 2: Parameters used

Cache servers	Number of caches
Learning rate	0.005882
Score	1000
Optimization Parameters	
Bw (Bandwidth rate)	Bandwidth describes the maximum data transfer rate of a network. It measures how much data can be sent over a specific connection in a given amount of time.
Dw (Download rate)	The download speed is defined as how fast user can pull data from the server Download speed is measured in megabits per second (Mbps)

Table 3 : Bandwidth and Download Rate Ranges using in Simulation

Case no.	Bw (Bandwidth rate)		Dw (Download rate)	
	Minimum	Maximum	Minimum	Maximum
1	3334 Kbps	5000Kbps	10 Kbps	20 Kbps
2	6667 Kbps	13334Kbps	10 Kbps	20 Kbps
3	16667 Kbps	33334Kbps	10 Kbps	20 Kbps

The table 3 gives the minimum and maximum values the bandwidth and download rates used in the Fog Zone of the cloud network. The bandwidth and download rates are normally low as compared to the computer network zone.

Table 4: Initial Population and cost analysis

Case no.	Number of Cache machines	Total Cost in Rs	Bw, Dw Pair Values: Computed Cost per Cache as per path	Total Time
1	5	7	##33053, 10##26473, 17##37142, 13##43460, 18##45375, 21##	1 Second
2	10	12	##48667, 27##46940, 25##37578, 17##17145, 19##20961, 21##23665, 24##25253, 22##39350, 16##38616, 18##48253, 18##	2 Second
3	15	15	##31289, 19##48074, 26##49813, 20##47448, 23##32718, 12##24762, 11##19983, 14##20409, 11##24995, 25##24619, 14##31286, 13##34409, 14##35545, 22##25969, 28##29490, 29##	3 Second

The above table shows the show the initial population is generated for TBLO and the experiment design parameters (number of cache machines in the Fog Network) and it shows total execution time.

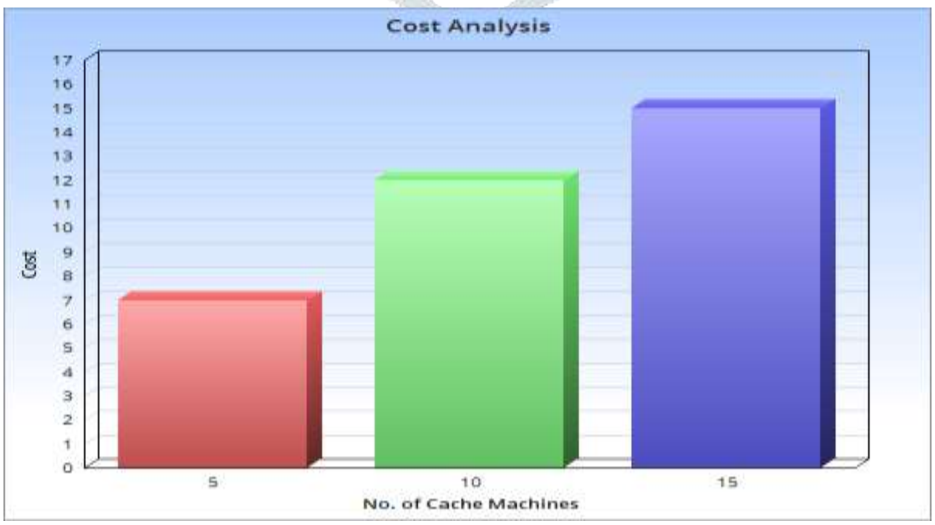

Figure 4: Cost Analysis

Interpretation: The bar graph shows the cost per evaluation round conducted by the optimization algorithm . In each optimization round the number of machines at the edge are increased so that all the aspects of the evaluation can be checked. The Fog network size is increased with 5, 10 , 15 Nano cache servers , as expected the cost increases exponentially . The cost was computed based on the bandwidth and download link and distance from the start of the source from where the data is send from the cache or temporary store of the content.

Evaluation Parameters:

In this work, the significance of setting up a test harness that encompasses the assortment of tests which would show the performance of the algorithm that has been implemented and used. The design of training datasets is also considered for checking the performance of the optimization problem with which we are dealing In evaluating the optimization algorithms that find approximated values based on the upper and lower limits of the bounded data can be either done by comparing the process with benchmarked optimization algorithm. But sometimes it does not make sense, if the benchmark algorithm is not working for a similar problem on which we are working. The first thing in such case, will be the definition of 'optimality' of an optimization algorithm.

VI. CONCLUSION

In this research work, we were able to develop a highly optimized solution for conducing link analysis in the Fog Computing network. The optimization algorithm was always able to optimal value of the derived metric that would reflect the health of the network at that every end. The optimization algorithm was able to solution that would satisfy minimum criteria (The Quality Parameter: QoS) conditions and took minimum number of iterations to reach the optimal conditions, from which we can infer that the algorithm was performing with minimum overhead (Computation and other resources). This was evident from the fact that from next successive iterations did not produce better results. This way the Fog Network became efficient and the number of resources that we did not utilized and/ or over utilized were balanced out with passage of time.

REFRENCES

[1] Siming Wang , Xumin Huang, Yi Liu, and Rong Yu , "CachinMobile: An energy-efficient users caching scheme for fog computing," in *IEEE*, 2016, pp. 1--6.

[2] Mahadev Satyanarayanan, "Caching trust rather than content," in *Proceedings of the 9th workshop on ACM SIGOPS European workshop: beyond the PC: new challenges for the operating system*, 2000, pp. 245-246.

[3] Pantelis A Frangoudis, Yala Louiza, Adlen Ksentini, and Tarik Taleb, "An architecture for on-demand service deployment over a telco CDN," in *IEEE*, 2016, pp. 1--6.

[4] Sem Borst, Varun Gupta, and Anwar Walid, "Distributed caching algorithms for content distribution networks," in *IEEE*, 2010, pp. 1--9.

[5] Faqir Zarrar Yousaf, Marco Liebsch, Andreas Maeder, and Stefan Schmid, "Mobile CDN enhancements for QoE-improved content delivery in mobile operator networks," *IEEE Network*, vol. 277, pp. 14--21, 2013.

[6] Ramesh Raagopalan and Pramod K Varshney, "Data aggregation techniques in sensor networks: A survey," 2006.

[7] Rebecca Kai Cassar, Joseph Vella, and Joshua Ellul, "A conflict resolution abstraction layer for eventually consistent databases," in *IEEE*, 2016, pp. 1--5.

[8] Florence Clevenot and Philippe Nain, "A simple fluid model for the analysis of the squirrel peer-to-peer caching system," in *IEEE*, 2004.

[9] Shinji Minami, Masaaki Wakamoto, and Mitsuaki Kakemizu, "Data management system, primary server, and secondary server for data registration and retrieval in distributed environment," 2001.

[10] Paul FELIOT, Julien BECT, and Emmanuel VAZQUEZ, "A Bayesian approach to constrained single-and multi-objective optimization," *Journal of Global Optimization*, vol. 67, pp. 97--133, 2017.

[11] Zesong Fei et al., "A survey of multi-objective optimization in wireless sensor networks: Metrics, algorithms, and open problems," *IEEE Communications Surveys \& Tutorials*, vol. 19, pp. 550--586, 2017.

[12] M. Nazari-Heris, B. Mohammadi-lvatlo, and G.B. Gharehpetian, "A comprehensive review of heuristic optimization algorithms for optimal combined heat and power dispatch from economic and environmental perspectives," *Renewable and Sustainable Energy Reviews*, 2017.

[13] Raghu Bollapragada, Richard Byrd, and Jorge Nocedal, "Adaptive Sampling Strategies for Stochastic Optimization," *arXiv preprint arXiv:1710.11258*, 2017.

[14] A. N. Gorozhankin, M. A Grigor'ev, A. M. Zhuravlev, and D. A. Sychev, "Parametric optimization of a synchronous electric drive with improved mass and size parameters," *Russian Electrical Engineering*, vol. 86, pp. 697--699, 2015.

[15] Zahra Behesti and Siti Mariyam Shamsuddin, "Non-parametric particle swarm optimization for global optimization," *Applied Soft Computing*, vol. 28, pp. 345--359, 2015.

[16] Maxime C. Cohen, Ngai-Hang Zachary Leung, Kiran Panchamgam, Georgia Perakis, and Anthony Smith , "The impact of linear optimization on promotion planning," *Operations Research*, vol. 65, pp. 446--468, 2017.

[17] Nidhi Jain Kansal and Inderveer Chana, "Cloud load balancing techniques: A step towards green computing," *IJCSI International Journal of Computer Science Issues*, vol. 9, pp. 238-246, 2012.

[18] Xiao Chen and Liangmin Wang, "Exploring Fog Computing Based Adaptive Vehicular Data Scheduling Policies through a Compositional Formal Method-PEPA," *IEEE Communications Letters*, 2017.

[19] Victor Shnayder, Mark Hempstead, Bor-rong Chen, Geoff Werner Allen, and Matt Welsh, "Simulating the power consumption of large-scale sensor network applications," in *ACM*, 2004, pp. 188-200.

[20] Kaur Amanpreet, Bikrampal Kaur , Amanpreet Kaur, Bikrampal Kaur, and Dheerendra Singh, "Particle Swarm Optimization based Dynamic Load Balancing in Cloud Environment," in *International Confrence on "Computing for Sustainable Global Development"*, New Delhi, 2017.

[21] Dolly and Er. Suchitra , "Review on Optimization Techniques Used In Cloud Scheduling," *International Research Journal of Engineering and Technology (IRJET)*, vol. 4, 2017.

[22] Gilsoo Lee, Walid Saad, and Mehdi Bennis, "Online optimization techniques for effective fog computing under uncertainty," *MMTC Communications-Frontiers*, vol. 12, pp. 19--23, 2017.

[23] Dilip Kumar and Dr. Tarni Mandal, "Multi-Objective Virtual Machine Placement using Improved Teaching Learning Based Optimization in Cloud Data Centers," *International Journal of Applied Engineering Research*, vol. 12, pp. 10809--10815, 2017.

[24] R. Venkata Rao, "Jaya: A simple and new optimization algorithm for solving constrained and unconstrained optimization problems," *International Journal of Industrial Engineering Computations*, vol. 7, pp. 19--34, 2016.

[25] Dr. Anantkumar J.Umbarkar, N.M. Rothe, and Ashutosh Sathe, "OpenMP teaching-Learning based optimization algorithm over Multi-Core system," *Internatiional journal of intelligent systems technologies and Applications*, 2015.

[26] Rodrigo N and Ranjan, Rajiv and Beloglazov, Anton and De Rose, C, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," *Software: Practice and experience*, vol. 41, pp. 23--50, 2011.

[27] Harshit and Vahid Dastjerdi, Amir and Ghosh, Soumya K and Buyya, Rajkumar Gupta, "iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments," *Software: Practice and Experience*, vol. 47, pp. 1275--129, 2017.