Computers, Materials & Continua
DOI:10.32604/cmc.2022.019971
Article

Tech Science Press

Controller Placement in Software Defined Internet of Things Using Optimization Algorithm

Sikander Hans[1], Smarajit Ghosh[1], Aman Kataria[2], Vinod Karar[2,*] and Sarika Sharma[3]

[1]Department of Electrical and Instrumentation Engineering, Thapar Institute of Engineering & Technology, Patiala, India
[2]Optical Devices and Systems, CSIR-Central Scientific Instruments Organization, Sector 30-C, Chandigarh, 160030, India
[3]Department of Electrical Engineering, Government Polytechnic, Ambala City, Ambala, 134003, India
[*]Corresponding Author: Vinod Karar. Email: vkarar@rediffmail.com

Received: 03 May 2021; Accepted: 15 July 2021

Abstract: The current and future status of the internet is represented by the upcoming Internet of Things (IoT). The internet can connect the huge amount of data, which contains lot of processing operations and efforts to transfer the pieces of information. The emerging IoT technology in which the smart ecosystem is enabled by the physical object fixed with software electronics, sensors and network connectivity. Nowadays, there are two trending technologies that take the platform i.e., Software Defined Network (SDN) and IoT (SD-IoT). The main aim of the IoT network is to connect and organize different objects with the internet, which is managed with the control panel and data panel in the SD network. The main issue and the challenging factors in this network are the increase in the delay and latency problem between the controllers. It is more significant for wide area networks, because of the large packet propagation latency and the controller placement problem is more important in every network. In the proposed work, IoT is implementing with adaptive fuzzy controller placement using the enhanced sunflower optimization (ESFO) algorithm and Pareto Optimal Controller placement tool (POCO) for the placement problem of the controller. In order to prove the efficiency of the proposed system, it is compared with other existing methods like PASIN, hybrid SD and PSO in terms of load balance, reduced number of controllers and average latency and delay. With 2 controllers, the proposed method obtains 400 miles as average latency, which is 22.2% smaller than PSO, 76.9% lesser than hybrid SD and 91.89% lesser than PASIN.

Keywords: SD-IoT; adaptive fuzzy controller; enhanced sunflower optimization (ESFO); POCO placement tool; controller placement problem; and minimum latency

1 Introduction

Physical devices are connected to the internet with the help of Internet of Things (IoT) in which that is used to collect, share and analyze huge number of data. This concept is basic for the 2nd digital revolution and it was developed by Kevin Ashton before 17 years [1]. Generally, scalability and management problems rises due to the interconnection of large number of devices. But, it also has benefits in it. Benefits are considered as advance IoT applications like smart buildings, smart energy systems, intelligent healthcare systems and smart transport systems [2]. Moreover, another one advantage is that the devices in IoT can be connected with other devices at anytime and anywhere. Different kind of technologies like cloud computing, embedded systems, SDN and Wireless Sensor Networks (WSNs) [3]. To control the overall behavior of the network, SDN allows a central program in IoT framework and it is considered as one of the advanced technologies in IoT. Dynamic security problems, granular traffic filtering issues and security threats are quickly enabled by this controller [4].

Generally, researchers use this kind of controllers to secure computer networks by integrating SDN with IoT. So, the controllers are considered as the brain of the network and the performance of network is degraded when the controller does not work well. Proper placement of controller is required to operate the network [5]. Otherwise, some problems occurred in resource management, data routing paths and scalability properties. So, this paper concentrates on the deployment of controller in a suitable place based on the swarm intelligence concept. Because, the controller placement problem is the main issue faced in the SDN network and also the allocation of the switches to these controllers [6]. The foremost contributions of this paper are given by:

- In SD-IoT device, Adaptive Fuzzy logic controller has been designed to select the suitable location of controller and switch in a network to reduce the load balancing and propagation delay problems.
- Then, ESFO algorithm is suggested to split the network and to solve controller placement problem by selecting the controller based on a high density and minimum distance value.
- In this case, Pareto optimal controller (POCO) placement tool is used to design the controllers and nodes to perform the controller placement problem and the performance of proposed methodology is compared with existing methods by means of delay and average latency.

The rest of the paper structured is as follows, some of the related works are discussed in Section 2, Section 3 deals with the proposed methodology, and results are discussed in Section 4 and Section 5 is the conclusion.

2 Related Works

In large area network, mobile technology is used for the network. So, Song et al. [7] proposed SD-IoT networks. It is named such a way that SDN where used to combine with IoT network. Detecting spatial events and partitioning problems over multiple flow paths were the main aim of their work. A fragmentation-based control was proposed by Kobo et al. [8] for SD wireless network reselection mechanism and efficient controller placement. It offers an efficient route for distributing the SD wireless networking control across the network. A nature inspired algorithm for load balancing in SDN was presented by Akbar Neghabi et al. [9]. Load balancing would maximize network efficiency and reliability by improving the load balancing of distribution across multiple resources. Abdelaziz et al. [10] introduced a controller for the real setting of SDN clustered with distributed controller architecture. It comprises of multiple popular SDN controllers. The proposed distributed controller had been designed by considering the parameters like packet loss and latency with reasonable CPU utilization and it's outperformed without clustering.

A new hybrid SD architecture was proposed by Lee et al. [11] for integrating IoT technologies for smart environments. To enable self-configuration and SDN infrastructure, it would be compared with an advanced optimization for the real time network. Sub-modularity optimization model was proposed in [12] to formulate different view of controller placement problem in distributed network. By means of execution time and network latency, the proposed method outperformed all the existing methods. A new routing protocol was developed in [13] to minimize the energy consumption in heterogeneous device. SDN controller was acting as a centralized manager, which provided security to the whole network. In [14], a set of heuristic algorithms were presented to minimize control traffic in SD-IoT network. These two algorithms were fast and simple heuristic algorithm which even performs better in time complexity analysis. Moreover, Density Peak Clustering (DPC) algorithm was presented in [15] to minimize the average communication latency in multi-network area. That algorithm enhanced the stability and reliability as well as improves the network performance. In that article [16], an effective system model was created using SD-IoT framework against man-in-the-middle attacks in the whole network. Additionally, that model was executed using OpenFlow protocol, Raspberry Pi, and Kodi Media center tool.

The controller placement problem is the main issue faced in the SDN network and also the allocation of the switches to these controllers. From the analysis, it is cleared that most of the controller placement problem in SD-IoT network degraded with large packet propagation latency and irregular topology of controller placement. In order to overcome these issues, this paper concentrates on the swarm intelligence-based algorithms for proper controller placement in SD-IoT network. In the last two decades, it attracts lot of researchers to use this concept in different kind of computer science applications. The swarm intelligence-based algorithms gain lot of popularity among other algorithms and they have more flexible and dynamic ability in different fields.

3 Methodology of Proposed Work

3.1 Software-Defined Network IoT

IoT can allow the combined physical and virtual objects, in which the virtual objects can only be seen in the monitor screen. This helps to provide a new opportunity for the users to contact the objects of worldwide. IoT can be implemented with a higher number of devices and users. Fig. 1 explains the basic architecture of an IoT network. It consists of four basic layers namely, the data processing layer, application layer, sensing layer, the network layer.

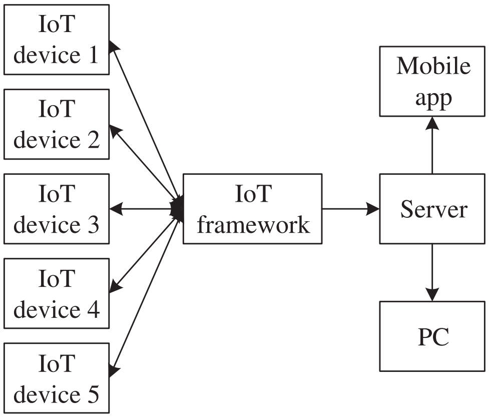


Figure 1: Simple architecture of an IoT network

In this, the different devices are connected to the IoT framework and stored all the data. The main objective of the SD-IoT network is to organize the network using the data plane and control plane and connect them over the internet. The selection of a suitable location of the controller and switch in a network for reducing the latency problem is the major challenge in the SD network [17]. Normally, controller placement leads to performance degradation as well as resilience issues. By considering these factors an adaptive fuzzy controller placement has been designed.

3.2 Adaptive Fuzzy Controller

The communication burdens among the single controller and a larger number of forwarders increase the network delay because of the restricted control capability of a single controller. Therefore, research into an IoT aims to place multiple controllers at different locations. When deploying multiple controllers, the latency among each controller plays an important role. Also, a real network will consider the critical factor on the controller load. Accordingly, when the load of a controller reaches a threshold, the propagation latency will be increased between the controller and the node. In this concept, an optimal solution to the design of the adaptive fuzzy controller placement problem by considering the load balancing, delay constraints, failure tolerance is defined. The objective of the proposed method is to find an optimal solution to delay constraints, controller selection, and adaptive fuzzy controller placement by balancing the load and reducing the propagation latency.

3.2.1 Fuzzy Logic System

The fuzzy logic system includes four stages namely, 1) the fuzzifier, 2) the knowledge base, 3) the defuzzifier, and 4) the fuzzy inference engine on fuzzy rules [18]. Then the fuzzy dynamic system can be represented by Eq. (1).

$$Y(x) = \frac{\sum_{i=1}^{\Delta} N^i \Pi_j^n \eta_{A_{ij}}(x_j)}{\sum_{i=1}^{\Delta} \Pi_j^n \eta_{A_{ij}}(x_j)} \tag{1}$$

where, $N^i = \max_{Y\in\aleph} u_N^i(Y)$, Where $[x = x_1, x_2, \ldots, x_n]^T$ represents the system input, Δ denoted the number of fuzzy rules and Y represents the output of the system

The membership function is described by Eq. (2).

$$\phi_i(x) = \frac{\Pi_{p=1}^n u_{M_i^p}(x_p)}{\sum_{i=1}^{\Delta} \Pi_{p=1}^n u_{M_i^p}(x_p)} \tag{2}$$

Defined $[N^1, N^2, N^3, \ldots, N^\Delta]^T = [\lambda^1, \lambda^2, \lambda^3, \ldots, \lambda^\Delta]^T$ and $\phi(x) = [\phi_1(x), \phi_2(x), \ldots, \phi_\Delta(x)]^T$. Therefore, the output of the system can be described by Eq. (3).

$$y(x) = \lambda^T \phi(x) \tag{3}$$

3.2.2 Design of Adaptive Fuzzy Control

The three steps have to be derived in this section such as adaptive laws, controller derivation and control signals. Before deriving it has to be considered the changes in the coordinates.

Let β^{i} be the control signal, where i = {1, 2, 3,..., n} and e^{i} is the change in coordinates. It can be defined by Eq. (4).

$$\left\{\begin{array}{l} e^1 = x_1 \\ . \\ . \\ e^i = x^i - \beta^{i-1} \\ e^n = x_n - \beta^{n-1} + \frac{1}{\Delta} x_{n+1} \end{array}\right\} \tag{4}$$

where, x_{n+1} is replaced by $\left(\frac{1}{\Delta}\right) x_{n+1}$ in the last step of the above equation. The first immediate control signal β^1 and the adaptive law $\dot{\hat{\Phi}}$ are given by Eqs. (5) and (6) respectively.

$$\beta^1 = -h^n r^n - \frac{r^n \hat{\Phi} \phi^T(x_i) \phi_i(x_i)}{2\chi_i^2} - \frac{r^i}{2} \tag{5}$$

$$\dot{\hat{\Phi}} = \frac{\lambda_i r^{i^2} \phi_i^T(x_i) \phi_i(x_i)}{2\chi_i^2} - \Pi_i \Phi_i \tag{6}$$

Here, $h^i, \chi_i, \lambda_i, \Pi_i > 0$ are the constants, $\hat{\Phi}$ is determined by Φ_i and $\hat{\Phi}_i = \Phi_i - \bar{\Phi}_i$. The adaptive law and the desired controller can be determined by Eqs. (7) and (8) respectively.

$$\nu = -h^n r^n - \frac{r^n \hat{\Phi} \phi^T(x_n) \phi_n(x_n)}{2\chi_i^2} - \frac{r^n}{2} \tag{7}$$

$$\dot{\hat{\Phi}} = \frac{\lambda_n r^{i^2} \phi_n^T(x_n) \phi_n(x_n)}{2\chi_n^2} - \Pi_n \Phi_i \tag{8}$$

Here, $h^n, \chi_n, \lambda n, \Pi_n > 0$ are the constants. $\hat{\Phi}$ is determined by Φ_i and desired controller $\hat{\Phi}_i = \Phi_i - \tilde{\Phi}_i$. The following are the detailed proof.

Theory 1: Under the immediate control signal β^{i}, adaptive law $\dot{\hat{\Phi}}$ and desired controller u, the variable of the system will drive to the origin.

Proof:

Step 1:

The $\dot{r}_1$ can be calculated by Eq. (9).

$$\dot{r}^1 = \dot{x}_1 = x_2 + f_1(x_1) + \Lambda_1(x, l) \tag{9}$$

The Lyapunov function is defined by Eq. (10).

$$\Theta^1 = \frac{1}{2} r_1^2 + \frac{1}{2\lambda_1} \tilde{\Phi}^2 \tag{10}$$

Then, $\dot{\Theta}^1$ can be derived from Θ^1 and given by Eqs. (11) and (12) respectively.

$$\dot{\Theta}^1 = r^1\dot{r}^1 - \frac{1}{\lambda_1}\hat{\Phi}\dot{\hat{\Phi}} \tag{11}$$

$$\dot{\Theta}^1 = r^1\left(x_2 + f_1(x_1) + \Lambda_1(x,l) + \frac{1}{\lambda_1}\tilde{\Phi}_1\dot{\hat{\Phi}}\right) \tag{12}$$

According to young's equality [19], the following expression is obtained by Eq. (13).

$$r^1\mathit{\Lambda}_1(x,l) \le \frac{r_1^2D_1^2}{2b_1^2} + \frac{b_1^2}{2} \tag{13}$$

Substituting Eq. (13) into Eq. (12), the expression given in Eq. (14) is obtained.

$$\dot{\Theta}^1 \le r^1\left(x_2 + f_1(x_1) + \left(\frac{r^1D_1^2}{2b_1^2}\right) + \frac{b_1^2}{2} - \frac{1}{\lambda_1}\tilde{\Phi}_1\dot{\hat{\Phi}}_1\right) \tag{14}$$

If $\tilde{f}_1(x_1) = f_1(x) + \left(\frac{r^1D_1^2}{2b_1^2}\right)$, then we have

$$\tilde{f}_1(x_1) = \Theta_1^T\phi_1(x_1) + \tau_1(x_1) \tag{15}$$

Here, τ_1 is known as the scalar. The expression in Eq. (15) can be rewritten as Eq. (16),

$$\dot{\Theta}^1 \le r^1(x_2 + \Theta_1^T\phi_1(x_1) + \tau_1) + \frac{b_1^2}{2} - \frac{1}{\lambda_1}\tilde{\Phi}_1\dot{\Phi}_1 \tag{16}$$

By substituting the values from young's inequality and $\dot{\Theta}^1$ represented by Eq. (17),

$$\dot{\Theta}^1 \le r^1\left(\frac{r^1\Phi_1\phi_1^T(x_1)\phi_1(x_1)}{2\chi_1^2} + \frac{r^1}{2} + x_2\right) + \frac{\chi_1^2 + \tau_1^2 + b_1^2}{2} - \frac{1}{\lambda_1}\tilde{\Phi}_1\dot{\Phi} \tag{17}$$

Here, $\Phi = ||\Theta||^2$. In this work the following virtual control signal expression given in Eq. (18) is chosen for stabilizing the first order subsystem.

$$\beta^1 = -h^1r^1 - \frac{r^1\hat{\Phi}\phi_1^T(x_1)\phi_1(x_1)}{2\chi_1^2} - \frac{D^1}{2} \tag{18}$$

Using Eqs. (17) and (18), the expression given in Eq. (19) is obtained.

$$\dot{\Theta}^1 \le -h^1r^1 + \frac{\tilde{\Phi}}{\lambda_1}\left(\frac{\lambda_1{r^1}^2\phi_1^T(x_1)\phi_1(x_1)}{2\chi_1^2} - \dot{\hat{\Phi}}\right) \tag{19}$$

Step 2:

Similarly, $\overset{\bullet}{\Theta}^2$ is computed as Eq. (20).

$$\overset{\bullet}{\Theta}^2 = -\sum_{k=1}^{2} h^k r^k + \sum_{k=1}^{2} \frac{\overset{\sim}{\Phi}_k}{\lambda_k} \left(\frac{\lambda_k {r^k}^2 \phi_k^T \left(\hat{x}_k \right) \phi_k \left(\hat{x}_k \right)}{2\chi_k^2} - \overset{\bullet}{\Phi}_k \right) + r^2 r^3 + \sum_{k=1}^{2} \frac{\chi_k^2 + \tau_k^2 + b_k^2}{2} \tag{20}$$

Also, the value of $\overset{\bullet}{\Theta}^i$ can be computed as Eq. (21).

$$\overset{\bullet}{\Theta}^n = -\sum_{k=1}^{i} h^k {r^k}^2 + \sum_{k=1}^{i} \frac{\overset{\sim}{\Phi}_k}{\lambda_k} \left(\frac{\lambda_1 {r^k}^2 \phi_k^T \left(\overset{\bullet}{x}_k \right) \phi_k \left(\hat{x}_k \right)}{2\chi_k^2} - \overset{\bullet}{\hat{\Phi}}_k \right) + r^i r^{i+1} + \sum_{k=1}^{i} \frac{\chi_k^2 + \tau_k^2 + b_k^2}{2} \tag{21}$$

Finally, $\overset{\bullet}{\Theta}^n$ can be computed as Eq. (22).

$$\overset{\bullet}{\Theta}^n \leq -\sum_{k=1}^{n} \left(h^1 {r^k}^2 + \frac{\Pi_k}{2\lambda_k} {\overset{\sim}{\Phi}_k}^2 \right) + \sum_{k=1}^{n} \left(\frac{\chi_k^2 + \tau_k^2 + b_k^2}{2} + \frac{\Pi_k \Phi_k^2}{2\lambda_k} \right) \tag{22}$$

Here, $\overset{\bullet}{\Theta}^n$ can be represented as Eq. (23).

$$\overset{\bullet}{\Theta}^n \leq -b^0 \Theta^n + c^0 \tag{23}$$

where, $b^0 = \min\{2h^k, \tau_k\}, c^0 = \sum_{k=1}^{n} \left(\frac{\chi_k^2 + \tau_k^2 + b_k^2}{2} + \frac{\Pi^k \Phi_k^2}{2\lambda_k} \right)$. This gives a better performance as the parameters are regulating in b^0 and c^0. By designing appropriate parameter, it will provide the stability of a closed loop system. For this an optimal solution to delay constraints, controller selection by balancing the load and reducing the propagation latency is found. Here, by considering b^0 as the objective function with minimum delay and latency.

4 ESFO Algorithm

In this proposed work, ESFO is used for splitting the network and selecting the controller based on a high density and minimum distance. Pseudocode of the proposed methodology is displayed in Algorithm 1.

Algorithm 1: Pseudocode of ESFO algorithm

```
Begin
Initialize mortality rate (m), pollination rate (p), maximum no of iterations, LB, UB, population size (n) and no of dimension (dim)
Randomly initialize the population of n flowers
Calculate objective function for all individual sunflowers based on Eq. (24).
```

(Continued)

```
Find the best solution in the initial population.
Orient all plants towards the sun
While (k < MaxDays)
        Estimate the orientation vector for each plant
        Remove m (%) plants further away from the sun
        Calculate the step for each plant
        Best b plants will pollinate around the sun
        Evaluate the new individuals
        If the new individual is global best, update the sun
end while
Return best solution found
```

The optimized number of controllers is identified using the ESFO. The objective function can be expressed by Eq. (24).

$$f = \max(den)\min(Dis) \tag{24}$$

Here, den represents the density and Dis denotes the distance to the controllers. ESFO is a population-based nature inspired algorithm. It mimics the movement of sunflower towards the sun. From the movement, there will be possibility to fertilize between the neighboring flowers closest to it. The absorbed radiation for each flower is based on the distance between it and the sun. The sun will reduce the radiation absorbed from this as the distance increasing between them [20]. This can be expressed by Eq. (25).

$$S^i = \frac{P^s}{4\Pi d_i^2} \tag{25}$$

Here, S^i represents the amount of radiation, P^s is the power of sun, and distance between the flower and the sun is represented by d. The direction is adjusted by each flower as shown by Eq. (26).

$$\vec{D} = \frac{\overset{\circ}{x} - x^i}{||\overset{\circ}{x} - x^i||} \tag{26}$$

where, $i = 1, 2, 3, \ldots, n$. $\overset{\circ}{x}$ and x^i represents the best and current position of flower, o denotes the size of the population. The steps of individual flower towards the sun is given by Eq. (27).

$$d^i = \chi \times P(||x^i + x^{i-1}|| \times ||x^i + x^{i-1}||) \tag{27}$$

Here, d^i represents the inertial displacement of the plant and is also the pollination probability i.e., according to each distance between the flowers, the sunflower will pollinate with the neighboring flower and thus a new individual is generated in that position.

It has to be restricted the individuals from the maximum stepping [21]. The maximum step is given by Eq. (28).

$$d^{\max} = \frac{||x^{\max} - x^{\min}||}{2 \times N^p} \tag{28}$$

Here, $x^{\max}$ and $x^{\min}$ are upper and lower band values that give the objective function and N^p represents the total controllers. The new plantation will be given by Eq. (29).

$$\vec{x}^{i+1} = \vec{x}^{i} + d^i \times \vec{D}^{i} \tag{29}$$

The population of the individual generation has been the starting of the algorithm and this population may be random or even. Tab. 1 shows the parameters used in the optimization algorithm. From these procedures the controller is selected with high density and minimum distance, also for splitting the network. The following section discusses the main problem in the controller placement. Flowchart of the proposed methodology is displayed in Fig. 2.

Table 1: Simulation parameters and their values

Parameters	Values
Number of nodes	35
Number of controllers	3 (3, 19, 23 nodes are considered as controllers)
Packet size	64 kB
Traffic rate	10–100 pps
Number of sunflower (n)	20%
Pollination rate best values (p)	0.05%
Mortality rate (m)	0.1%
Problem dimension (d)	2%
Lower Band (LB)	[−5 −5]
Upper Band (UB)	[5 5]
Maximum number of iterations	100%

4.1 Controller Placement

Controller placement in SD-IoT is the main issue i.e., to meet the requirement of the network it has to arrange the limited controllers. In a network, latency is an important factor to be considered and the main aim is to implement a minimum-latency placement. Certain resilience constraints have to be considered for the controller placement. For example, here it shows the searching for the best controller placement in internet2 OS3E with k = 5 controllers [22]. Different issues like latency between inter controllers, load unbalancing, failure of the controller, disruption of the network, and the controller placement problem are explained in it.

POCO placement tool can be used to calculate the optimal results and the resilience against the failure of the controller. The inclusion of resilience against controller failures are detailed below. If a controller faces a failure condition, the nodes associated with this controller are reassigned to some other controller having latency issues. Let $L_{\max}$ be considered as the maximum latency between nodes to the controller, $q(x, y)$ which is the shortest path between all the nodes x and y in the set of Δ. For $\chi \in 2^{\Delta}$, maximum latency can be given by Eq. (30).

$$L_{\max}(\chi) = \max \min q(x, y) \tag{30}$$

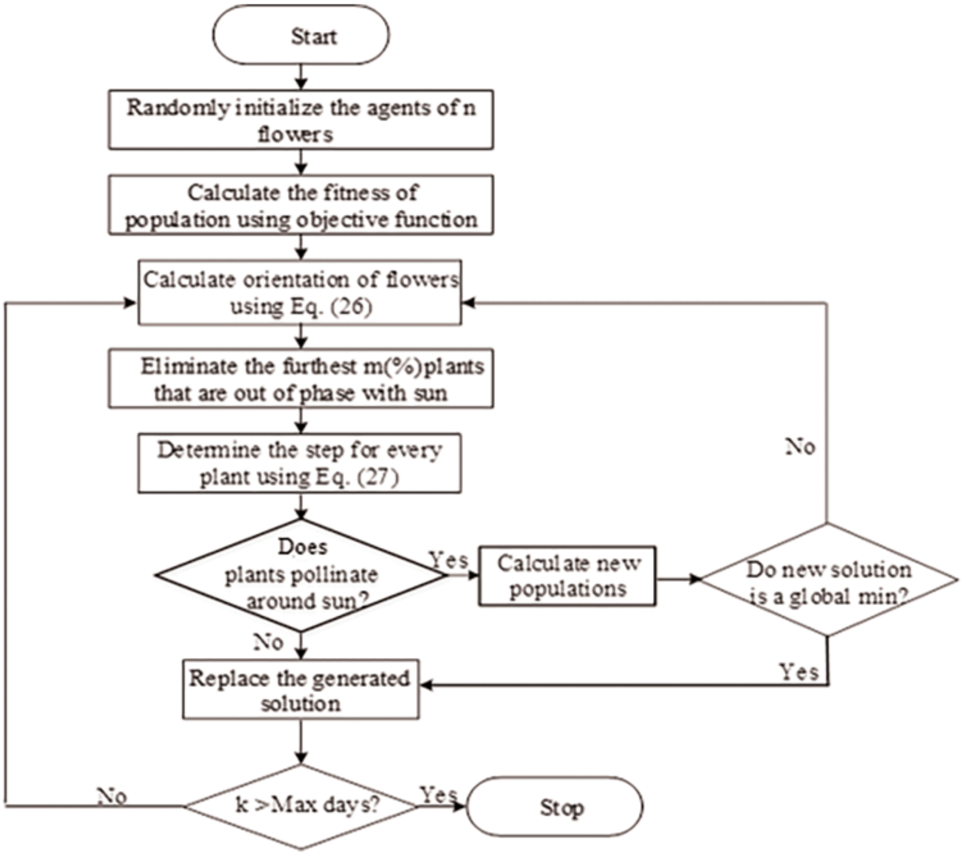

Start
Randomly initialize the agents of n flowers
Calculate the fitness of population using objective function
Calculate orientation of flowers using Eq. (26)
Eliminate the furthest m(%)plants that are out of phase with sun
Determine the step for every plant using Eq. (27)
No
Does plants pollinate around sun?
Yes
Calculate new populations
Do new solution is a global min?
No
Replace the generated solution
Yes
No
k >Max days?
Yes
Stop

Figure 2: Flowchart of proposed ESFO algorithm

Here, for maximum $x \in \Delta$ and minimum $y \in \chi$, $L^{\Phi}_{\max}$ represents the maximum latency for the failure free case. By considering the set scenarios ζ it takes all combinations up to $n-1$ controller failure.

Let the placements χ^1 be the working controller subsets of a placement which can be considered the controller failure ζ. Then, the maximum latency is given by Eq. (31).

$$L^{\zeta}_{\max}(\chi) = \overset{x\in\Delta}{\max} \; \overset{\Phi\subset\chi^1\subseteq\chi}{\max} \; \overset{y\in\chi^1}{\min} \; q(x,y) \tag{31}$$

Here, $L^{\Phi}_{\max}$ can be minimized, when controllers are placed in the network. On the other hand, if even in work failure cases we place several controllers to reach the best $L^{\Phi}_{\max}$.

Corresponding latency can be expressed by Eq. (32).

$$L^{\zeta}_{\max} = \overset{\chi\in\{\chi^1,\chi^2,\ldots.,\chi^{|x^n|}\}}{\min} \; L^{\zeta}_{\max}(\chi) \tag{32}$$

The center of the network always gives the space for the controller as we expected. Fig. 2 shows the worst condition; i.e., if one controller failed its operation [23]. In this placement, they give the best five combinations of placement with the single controllers of k = 5. When all the controller fails expect one, thus, the maximum latency can be defined by Eq. (33).

$$L_{\max}^{\zeta}(\chi) = \overset{x\in\Delta}{\max}\ \overset{y\in\chi}{\max}\ q(x,y) \tag{33}$$

Even in the case of $n-1$, the maximum latencies are low also for regular outage of the controller, as compared to the best placement $\chi_{\max}^{\zeta}$. This placement with low $L_{\max}^{\zeta}$ leads to an increase of $L_{\max}$.

Fig. 3 indicates the latency in worst case of failure with the black points with black lines for the interconnection. In failure free case, for resilient Pareto-optimal placements that give better performance. Thus, for a selected placement it does not offer any recommendation. However, it enables the operator for choosing the placement, which returns the Pareto optimal placement. For most controllers, the shown placement results in high maximum latencies to other controllers, which might be not acceptable for certain use cases. Therefore, inter-controller latency is part of POCO's set of analyzed objectives. It is able to handle huge network instances without sacrificing much accuracy.

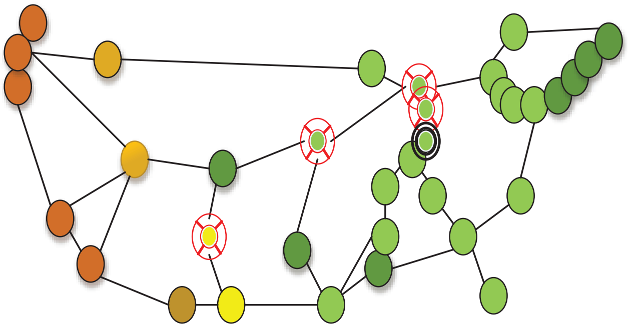


Figure 3: Latency in worst case of failure

In the case of controller-less nodes, it contains a controller for nodes always having two neighbors, which fails its operation. $L_f^{controller-less}$ can be defined by Eq. (34).

$$L_f^{controller-less}(\chi) = \overset{r\in N}{\max} \sum_{x\in\Delta} \overset{y\in\chi}{\min}\, h^r(x,y) \tag{34}$$

Here, Maximum latency on the controller less node looking for placement with failure δ is represented by $L_{\max}^{\delta}$. Where $h^r(x,y)$ be disconnection matrices.

5 Result and Discussion

In this paper, IoT is implemented with adaptive fuzzy controller placement using the enhanced sunflower optimization and POCO placement tool for the placement problem of controller. The splitting of the network is done by the sunflower optimization algorithm and the controller placement problem is solved using the Pareto optimal controller (POCO) placement tool by considering the latency and load constraints. When a controller fails, some nodes are reassigned to other controllers and experience increased latencies. To quantify the latencies in a network, we take the maximum over all node to-controller-latencies. Here, clustering is inbuilt with node failure rectification. Furthermore, load balancing is also considered along with the propagation delay metric. The MATLAB/Simulink is used to implement the IoT and comparing it with the existing technique. Tab. 1 lists the simulation parameters and their values.

Here, we considered the POCO placement tool with 35 sets of nodes and controller k = 3 (3 19 33), which are shown in Fig. 4. The nodes are represented with numbers noted in a circle and the controllers are represented by double circles. Fig. 5 represents the controller failure node with the controller misbalancing and node 6 denotes the node failure. Therefore, it faces the misbalancing condition and it affects the node from the latency problem and causes the delay operation of the network. Fig. 6 represents the controller less node operation with the controller imbalance. The color of each node, based on a traffic light color scheme, denotes the latency to the node's controller, which is selected based on the lowest distance. Here the controllers are highlighted with the double circle with number 33,19 and the controller less node is represented by 3. The proper working of the network with this controller imbalance from the maximum imbalance is explained as follows.

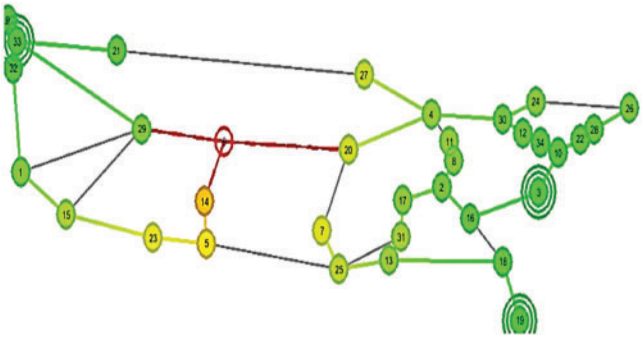


Figure 4: Controller placement with nodes

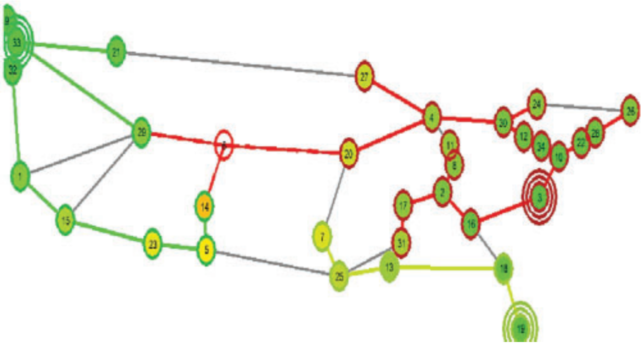


Figure 5: Node failure with controller imbalance

The controller imbalance in two cases are shown in Fig. 7, which represents the two cases also with a different number of controllers used to represent the accurate results. Fig. 7a gives the controller imbalance with the failure free operation in the y-axis with respect to the maximum controller imbalance in the x-axis. The controller imbalance is taken from node 5 to node 35 and it also shows the mean value of the maximum controller imbalance. Fig. 7a show a proper failure free condition with the controller. Fig. 7b shows the failure condition of a controller where a controller fails its operation the other will gives a proper placement. Therefore, the POCO placement tool gives a proper controller placement by considering the load balancing with a reduced latency. Tab. 2 lists the comparison analysis of delay and average latency or proposed method with existing methods.

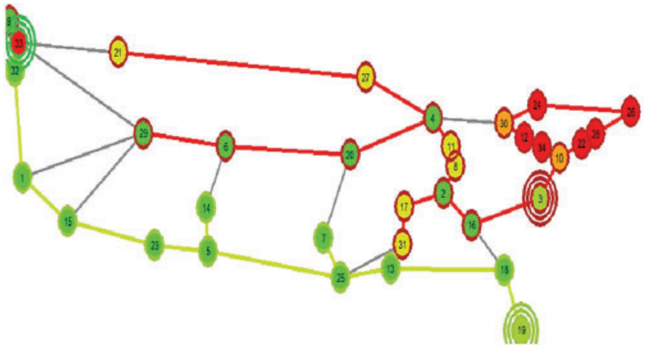
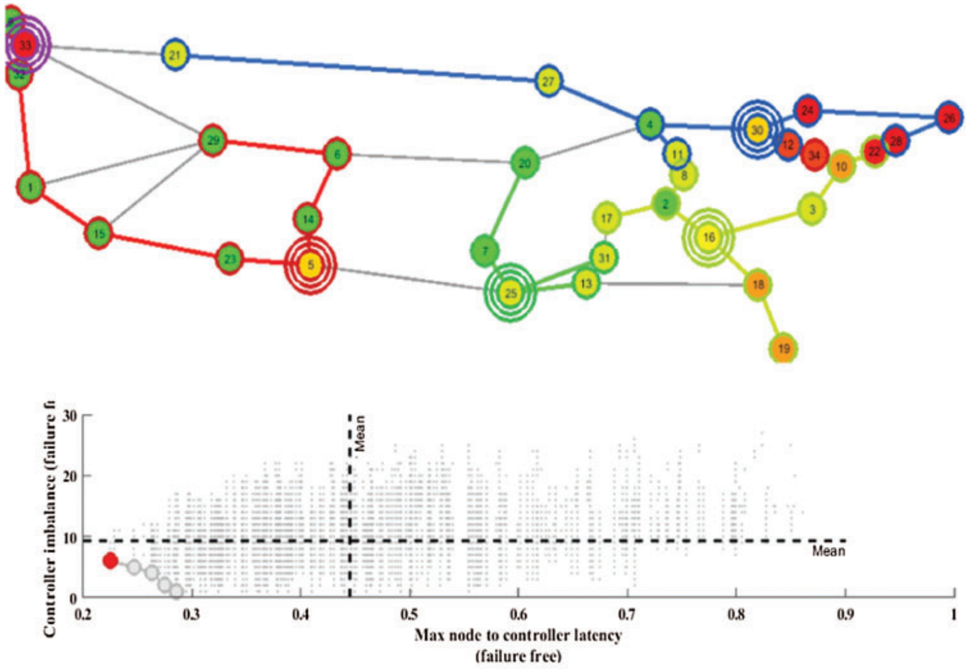
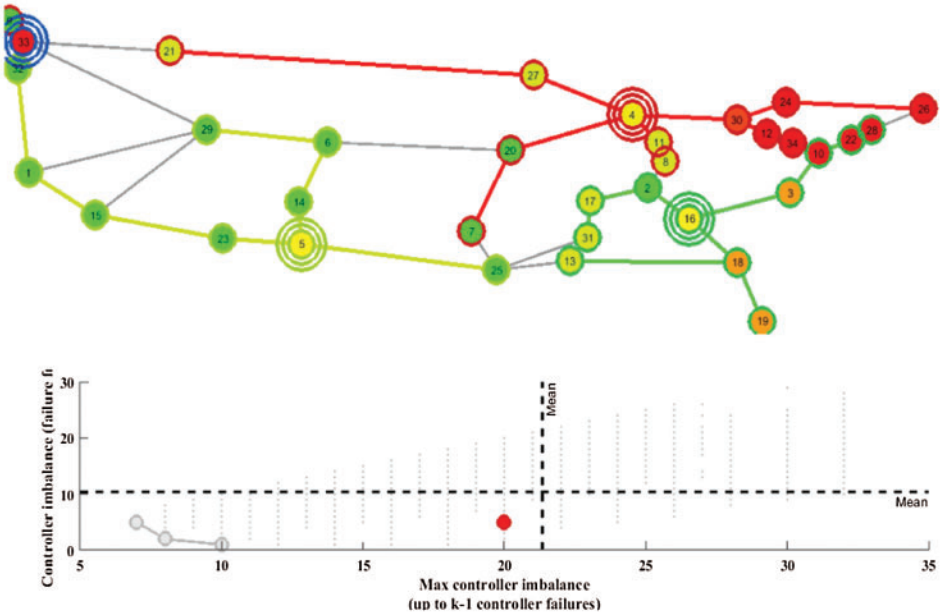


Figure 6: Controller less node operation



(a)



(b)

Figure 7: Controller imbalance (a) Controller imbalance with failure free conditions, (b) Contoller imbalance with a controller failure condition

Table 2: Comparison analysis of delay and average latency

Number of controllers		2	4	6	8	10	12	14	16	18	20
Delay (ms)	Proposed	2.8×10^4	2.99×10^4	3.2×10^4	3.5×10^4	3.99×10^4	4.1×10^4	4.5×10^4	5×10^4	5.2×10^4	5.8×10^4
	PASIN	3×10^4	3.4×10^4	3.8×10^4	4.1×10^4	4.5×10^4	5×10^4	5.2×10^4	5.5×10^4	5.9×10^4	6.2×10^4
	PSO	3.2×10^4	3.8×10^4	4.2×10^4	4.8×10^4	5.5×10^4	6.1×10^4	6.5×10^4	6.9×10^4	7.1×10^4	7.5×10^4
	Hybrid SD	3.4×10^4	4×10^4	4.5×10^4	5×10^4	5.5×10^4	6×10^4	6.3×10^4	6.5×10^4	6.8×10^4	6.8×10^4
Average latency (miles)	Proposed	400	180	100	100	100	80	60	40	40	40
	PASIN	1100	900	590	400	340	240	220	210	220	200
	PSO	500	210	180	150	150	150	100	80	80	60
	Hybrid SD	900	440	390	220	200	170	170	170	150	150

Fig. 8 represents the comparison analysis of the network based on the delay response and the average latency response by comparing it with the different recent technics. The proposed system gives a reduced latency and delay issue as compared to the existing system. Generally, as the number of controllers increases, the delay and latency value will seem to be increased for every network. Each sub topology consisting of at least two nodes that can be cut off from the entire network by at most two link- or node-failures has to be covered by at least one controller. Placing a controller in a sub topology automatically also covers all larger sub topologies that includes the smaller one. This allows to develop a procedure to calculate k for a topology without evaluating πN controller-less for all possible placements. Without this trick, the computational effort to find an optimal k could not be handled. The minimum number of controllers can be found as follows. Pareto-optimal placements that perform better in the resilient case perform worse in the failure free case and vice versa. Thus, POCO usually gives no recommendation for a particular placement, but returns the set of Pareto-optimal placements, which enables the network operators to choose the placement that fits their needs best. For our proposed system, if the number of controllers considered to be 6 and the value of delay, latency will be 3×10^4 ms and 400 miles, respectively and the other existing techniques like PASIN, which mentioned in the related work, Hybrid SD [21] and PSO [24]. It will have high value of delay and latency. The latency and delay gained by the proposed system are relatively stable and volatile as compared to the others. Thus, the proposed algorithm is seemed to be the best solution for setting the minimum latency and delay issues in the SD-IoT system.

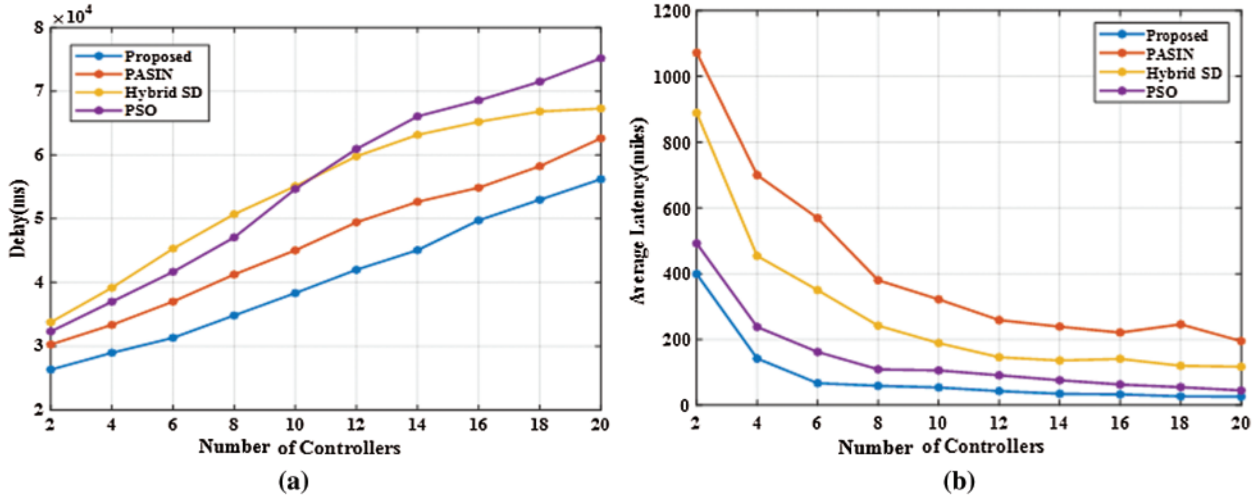


Figure 8: Comparison analysis. (a) Delay response with the number of controllers, (b) Average latency with the number of controllers

6 Conclusion

In this paper, the controller placement problem for the distributed network has been focussed, specifically for the IoT network, and the concept of optimization algorithm has been applied to solve the class of controller placement problem, which is common in real systems. Initially, an adaptive fuzzy controller concept has been utilized to find an optimal solution for controller selection in which it has balanced the load and minimized the propagation latency in whole SD-IoT network. Finally, an effective ESFO algorithm is proposed to split the network and to select the best controller based on high density and minimum distance value using POCO tool. The MATLAB/Simulink is used to implement the system and compared the proposed system with the existing techniques like PASIN, Hybrid SD and PSO. With 2 controllers, the proposed method obtains 400 miles as average latency, which is 22.2% smaller than PSO, 76.9% lesser than hybrid SD and 91.89% lesser than PASIN. In the future, we intend to test the proposed strategy in realistic scenarios and design a dynamic provision mechanism for the SDN Controller. Moreover, different performance metrics like energy consumption, and response time must be considered in the future work to improve the quality of service (QoS) in IoT networks.

Funding Statement: The authors received no specific funding for this study.

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

References

[1] F. Tang, Z. M. Fadlullah, B. Mao, N. Kato, “An intelligent traffic load prediction-based adaptive channel assignment algorithm in SDN-ioT: A deep learning approach,” *IEEE Internet of Things Journal,* vol. 5, no. 6, pp. 5141–5154, 2021.

[2] R. Krishnamurthi, A. Kumar, D. Gopinathan, A. Nayyar, B. Qureshi, “An overview of IoT sensor data processing, fusion, and analysis techniques,” *Sensors,* vol. 20, no. 21, pp. 6076 (1–23), 2020.

[3] S. Sennan, S. Ramasubbareddy, S. Balasubramaniyam, A. Nayyar, M. Abouhawwash, N. A. Hikal, “T2fl-pSO: Type-2 fuzzy logic-based particle swarm optimization algorithm used to maximize the lifetime of internet of things,” *IEEE Access,* vol. 29, no. 9, pp. 63966–63979, 2021.

[4] F. Tang, B. Mao, Z. M. Fadlullah, N. Kato, “On a novel deep-learning-based intelligent partially overlapping channel assignment in SDN-ioT,” *IEEE Communications Magazine,* vol. 56, no. 9, pp. 80–86, 2018.

[5] M. J. Islam, M. Mahin, S. Roy, B. C. Debnath, A. Khatun, “Distblacknet: A distributed secure black sdn-iot architecture with nfv implementation for smart cities,” in *2019 Int. Conf. on Electrical, Computer and Communication Engineering (ECCE)*, Cox’sBazar, Bangladesh, IEEE, pp. 1–6.

[6] A. Nayyar, D. N. Le, N. G. Nguyen. *Advances in Swarm Intelligence for Optimizing Problems in Computer Science,* New York: CRC Press, 2018.

[7] C. Song, J. Wu, X. Chen, L. Shi and M. Liu, “Towards the partitioning problem in software-defined IoT networks for urban sensing,” in *Int. Conf. on Pervasive Computing and Communications (PerCom)*, Athens. Greece, pp. 1–9, 2018.

[8] H. I. Kobo, M. A. Abu-Mahfouz and G. P. Hancke, “Efficient controller placement and reelection mechanism in distributed control system for software defined wireless sensor networks,” *Transactions on Emerging Telecommunications Technologies,* vol. 30, no. 6, pp. e3588, 2019.

[9] A. Akbar Neghabi, N. Jafari Navimipour, M. Hosseinzadeh and A. Rezace, “Nature-inspired metaheuristic algorithms for solving the load balancing problem in the software-defined network,” *International Journal of Communication Systems,* vol. 32, no. 4. pp. e3875, 2019.

[10] A. Abdelaziz, A. T. Fong, A. Gani, U. Garba, S. khan *et al.*, “Distributed controller clustering in software defined networks,” *PloS One,* vol. 12, no. 4, pp. e0174715, 2017.

[11] A. Lee, X. Wang, H. Nguyen and I. Ra, “A hybrid software defined networking architecture for next-generation ioTs,” *KSII Transactions on Internet and Information Systems,* vol. 12, no. 2, pp. 932–945, 2018.

[12] A. K. Tran, M. Piran, C. Pham, “SDN controller placement in IoT networks: An optimized submodularity-based approach,” *Sensors,* vol. 19, no. 4, pp. 5474, 2019.

[13] C. Kharkongor, T. Chithralekha, R. Varghese, “A SDN controller with energy efficient routing in the internet of things (IoT),” *Procedia Computer Science,* vol. 89, pp. 218–227, 2016.

[14] K. Choumas, D. Giatsios, P. Flegkas, T. Korakis, “SDN controller placement and switch assignment for Low power IoT,” *Electronics,* vol. 9, no. 2, pp. 325, 2020.

[15] X. Cui, X. Gao, Y. Ma, W. Wang, “A controller deployment scheme in 5G-ioT network based on SDN,” *EURASIP Journal on Wireless Communications and Networking,* vol. 248, pp. 1–19, 2020.

[16] A. Al Hayajneh, M. Z. Bhuiyan, I. McAndrew, “Improving internet of things (IoT) security with software-defined networking (SDN),” *Computers,* vol. 9, no. 1, pp. 8(1–14), 2020.

[17] M. Cuka, D. Elmazi, M. Ikeda, K. Matsuo and L. Baroli, “Iot node selection and placement: A New approach based on fuzzy logic and genetic algorithm,” in *Conf. on Complex, Intelligent, and Software Intensive Systems*, Cham, Springer, pp. 22–35, 2019.

[18] A. Ali, "A robust direct adaptive fuzzy control for a class of uncertain nonlinear MIMO systems," *Soft Computing,* vol. 23, no. 19, pp. 9747–9759, 2019.

[19] C. Wu, J. Liu, X. Jing, H. Li and L. Wu, "Adaptive fuzzy control for nonlinear networked control systems," *IEEE Transactions on Systems, Man and Cybernetics: Systems,* vol. 47, no. 8, pp. 2420–2430, 2017.

[20] G. F. Gomes, S. S. da Cunha and A. C. Ancelotti, "A sunflower optimization (SFO) algorithm applied to damage identification on laminated composite plates," *Engineering with Computers,* vol. 35, no. 2, pp. 619–626, 2019.

[21] M. H. Qais, H. M. Hasanien and S. Alghuwainem, "Identification of electrical parameters for three-diode photovoltaic model using analytical and sunflower optimization algorithm," *Applied Energy,* vol. 250, pp. 109–117, 2019.

[22] S. Lange, S. Gebert, T. Zinner, P. Tran-Gia, D. Hock *et al.*, "Heuristic approaches to the controller placement problem in large scale SDN networks," *IEEE Transactions on Network and Service Management,* vol. 12, no. 1, pp. 4–17, 2015.

[23] D. Hock, M. Hartmann, G. Gebert, M. Jarschel, T. Zinner *et al.*, "Pareto-optimal resilient controller placement in SDN-based core networks," in *Proc. of the 2013 25th Int. Tele Traffic Congress (ITC)*, Shanghai, China, pp. 1–9, 2013.

[24] C. Gao, H. Wang, F. Zhu, L. Zhai, S. Yi *et al.*, "A particle swarm optimization algorithm for controller placement problem in software defined network," in *Int. Conf. on Algorithms and Architectures for Parallel Processing*, Cham, Springer, pp. 44–54, 2015.