ISSN 2303-4521
Periodicals of Engineering and Natural Sciences *Original Research*
Vol. 10, No. 4, July 2022, pp.79-93

Using metaheuristics to improve the placement of multi-controllers in software-defined networking enabled clouds

Neamah Sattar Radam[1], Sufyan Al-Janabi[2], Khalid Shaker Jasim[3]
[1,2,3] College of Computer Science and IT, University of Anbar, Ramadi, Iraq

ABSTRACT

SDN is a model that separates the control and the data levels in an arrangement to enhance capability to program and configure the network in a more agile and efficient manner. Multiple controller modules have been used in the SDN engineering to empower programmable and adaptable configurations such as improving scalability and reliability. The distance and time calculations and other performance measures have to be considered in solving the Multi-Controller Position Problem (MCPP). This paper investigates the use of metaheuristic algorithms to build an MCPP mathematical model. Both the symmetric Harmony Search (HS) modelling and the Particle Swarm Optimization (PSO) algorithm are considered in this respect. Thus, our hybrid approach is proposed and known as Harmony Search with Particle Swarm Optimization (HSPSO) is applied and we compared the extracted results with the state-of-the-art techniques in the previous literature. Besides the development of the mathematical model, a simulation study has been done considering the relevant parameters including the link distance description and the access time between the SDN entities. The console simulation uses NetBeans with CloudsimSDN procedure files in the SDN-based cloud environment.

Keywords: SDN, HS, PSO, MCPP, HSPSO, CloudsimSDN.

Corresponding Author:

Neamah Sattar Radam
College of Computer Science and IT, University of Anbar
Ramadi, Iraq
Email: nea19c1010@uoanbar.edu.iq

1. Introduction

Software Defined Networking (SDN) is an energetic, developing, reasonable, cost-effective and scalable design. These features make it perfect for the tall transfer speed and energetic nature of today's applications. These separate the physical engineering of the organized control plane from the redirection plane. Here, the control plane controls multiple devices i.e., separating the organization control and sending capacities. This separation empowers the organization control to be straightforwardly programmable. It also extricates the fundamental foundation for the applications and organizes administrations. SDN hubs (sending switches) with Link Layer Discovery Protocol identify the complete organize topology [1].

SDN is a novel network paradigm separating routing and controls aircraft traditionally paired with one another, with the adoption of a logical centralized architecture aimed at increasing elements throughout the router [2, 3]:

- Control function: This function controls the traffic in relation to priority taking.
- Data function: This function performs data-based control functions by the police and provided by the SDN. It shows how some long-term problems in network security can be addressed by exploiting the capabilities of the SD [4].

To summarize, it is easy to characterize the first definition of SDN as an organization where control plane is partitioned in a physical way from the forwarding plane while a single control plane controls many forwards [5]. SDN has noteworthy preferences and characteristics within the network in terms of arrangements and applications, and this makes the Controller Placement Problem (CPP) a significant point of research works and solutions. Here, the focus is on performance measures and many other objectives.

Console mode approaches and some mechanisms are introduced for discovering and constantly updating the network topology. SDN has found multidimensional challenges such as scalable technical challenges, financial and business challenges, and fault tolerance and centralization [6].

The general aspects found in console placement solutions are described along with three distinct aspects: Static vs. dynamic, protected vs. unprotected, and real network conditions vs. assumptions. There are two broad approaches of control plane scalability: topology-related and mechanisms-related approaches [7-9]. The control plane is scalable and resilient and is discussed in terms of the controller placement [10, 11].

The paper then consists of: Section II presents a short literature survey. The SDN concept is discussed in Section III, including SDN architecture, SDN interfaces API and open source of SDN controllers. Section IV presents problem statement along with the research methodology which includes the proposed method procedure, pseudocode and a case study with the two algorithms (HS and PSO). Next, Section V discusses the results of the simulation experiments and the discussion. Lastly, Section VI suggests further works and puts some concluding remarks.

2. Related works

In this section, we review some interesting literature related to the use of heuristic approaches and calculations for solving the MCPP. A study [7] provided a qualitative comparative analysis of thirty-four different controllers and a quantitative analysis of a network of nine controllers to improve its performance. They also presented a comprehensive study of benchmarking techniques. Three different standard measurement tools were used for measuring and evaluating the performance of SDN controllers such as testing performance for latency, throughput, and round-trip time, the aim of which is to evaluate the architecture and modern developed SDN controllers. Another study [10] used eight controllers and measured the average delay in the network and used nine algorithms. These algorithms worked to choose the candidate keys, including the selection of the controller position in the SDN network and the k-Median selection algorithm and controller using k-Center to find the number of controllers and their positions. They are also fast failover approach constraints evaluating the robustness of the control level as well the forwarding and backward phase for the switch. Some scholars [12] studied a system with fifteen controllers to find a propagation latency and cost and used two PSOs and Firefly (FA) algorithms meta-heuristic population-based to obtain optimal placement of the controllers to solve (CPP). In [13], the study prepared eleven controllers to find the average delay and used heuristic algorithms, with a low time complexity which suits solving high-dimensional problems such as parameter optimization model (POM) to optimize the parameters of other heuristic algorithms for solving CPP. The main algorithms were synthetically-delay controller placement model (SDCPM), PSO-based parameter optimization (PSOPO), FA-based synthetically delay controller placement (FACP), BA-based synthetically-delay controller placement (BACP) and VARNA-based optimization -based synthetically-delay controller placement (VBOCP) according to (PSO), (FA), Bat algorithm (BA) and (VBO). The results show that in terms of synthetic delay, variance, and time consumption, teaching learning-based optimization (TLBO), (PSO), (FACP), (BACP), and (VBOCP) perform better. Also [14] four SDN controllers were utilized for evaluating and comparing the performance of various Open Network Operating System (ONOS), Open Daylight (ODL), POX and Ryu, by two performance tests to measure the average Round-Trip Time (RTT), throughput, and jitter. It has been found that the performance results of the POX controller are good, better and more stable than other controllers. Also, in [15-17], they have configured and shaped SDN and the Internet of Things (SD-IoT) for the connection and organization of different objects to internet. In addition, the used three fuzzy controller’s placement controllers are used to find the average delay and used Pareto Optimal Controller placement (POCO) and the enhanced sunflower optimization (ESFO) algorithm for solving (CPP). In [18], an enhanced dynamic algorithm based on the Salp Swarm Optimization (SSOA) algorithm to obtain Chaotic Salp Swarm (CSSA) has been developed with the introduction of chaotic maps. These maps help to improve the performance of the optimizer and to obtain the optimum number of controllers and allocate switches to them in large scale SDN networks which reduce the latency and cost of deployment. Also, Gray Wolf Optimization (GWO) and (PSO) were. The proposed algorithm outperforms meta-heuristic algorithm execution time and reliability. In addition, [19] (VBO) algorithm was developed and used to solve CPP for the minimization of the total propagation latency of SDN along with execution of TLBO and Jaya algorithms. According to the findings, TLBO performance is better than that of PSO, and VBO is better than TLBO and Jaya algorithms. Also, another study [20] aims to find the cost model of Multi-Objective Hybrid Harmony Search-Simulated Annealing (MOHS-SA) and Economic Production Quantity (EPQ). They proposed a hybrid of PSO and HS, called PSOHS for determining and estimating the kinetic parameter of essential amino acids, primarily aspartate metabolisms, in Arabidopsis thaliana. PSOHS outperformed SA and the downhill simplex methods and confirmed the suitability of PSOHS to find time session (TS) [21]. They used thirteen controllers to find reliability. The groups selected for the algorithms were a High-Degree (HD) clustering algorithm, a controller selection algorithm, Max Degree with Short Distance (MDSD), a High Degree with Independent Dominating Set (HDIDS), a Low Degree with Short Distance (LDSD) and an Inter-Domain Adjacent with Short Distance (IDASD), Control Packet Drop Rate (CPDR), Weakly Connected Dominating Sets (WCDS), Connected Dominating Set (CDS), and Full Enumeration (FE), HDIDS Performance about Reliability, throughput, and survivability best performance and best average handling response time (CPP) [22]. There were eight controllers used NOX, POX, ODL, ONOS, Ryu, Floodlight, Beacon, and Trema [23]. Also Distributed Flow Architecture for Networked Enterprises (DIFANE) and Graphical User Interface (GUI) were used to reduce packet delay of (RTT) and describe open-source controllers. In [24], four controllers Ryu, Floodlight, ONOS and ODL were used to find (RTT), propagation latency, throughput and used the South African National Research Network (SANReN) to reduce the average delay load balancing, latency, reliability and cost-effective regarding Capital Expenditure (CapEx) and Operational Expenditure (OpEx). In [25], ready two ONOS and POX controllers help to find a propagation latency and Time Session (TS). Also, GUI modules, Worst Fit, Best Fit and Energy-saving was used to develop the framework effectiveness of CloudSimSDN and energy consumption. Furthermore, [26] SDN/Network Function Virtualization (NFV) Single Controller (ODL) and Adaptive Containers for Microservices in Distributed Cloud (ACMDC), SA, Tunneling, 3rd Generation Partnership Project (3GPP) Selection and Service Level Agreement (SLA), ARBAT Cellular System to find propagation latency, cost, and average delay were used to achieve scalability, flexibility, and throughput.

3. SDN concept

The basic concepts of SDN are as follows:

- Network programmability via standard Application Programming Interfaces (APIs).
- Network stripping for easy programming.
- Network automation using policy-based control.
- Global asset resource administration capacities over network layers.

3.1. SDN Architecture

SDN architecture has three layers that can be four planes: data, control, application and management and administration [2] as shown in Figure 1.

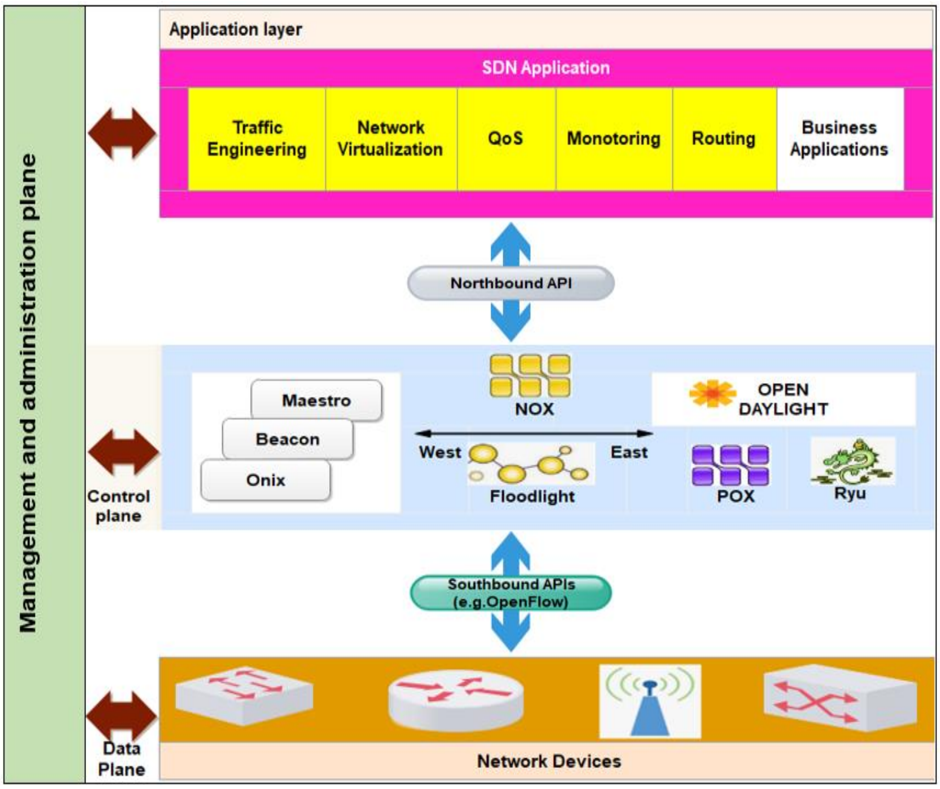

Figure 1. SDN Architecture [22, 2]

The traditional single SDN architecture has 6 major components are:

- Data Layer infrastructure layer (user plane).
- Management plane.
- East-West interfaces.
- Control Layer is the middle layer.
- Southbound interfaces.
- Northbound interfaces.

3.2. SDN interfaces API

API (Application programming interface) is a messenger that takes a request from an app to O.S and tells what you want to do and returns the (O.S) response back to the application. It is a software intermediary that allows two applications to talk to each other as Figure 2 shows.

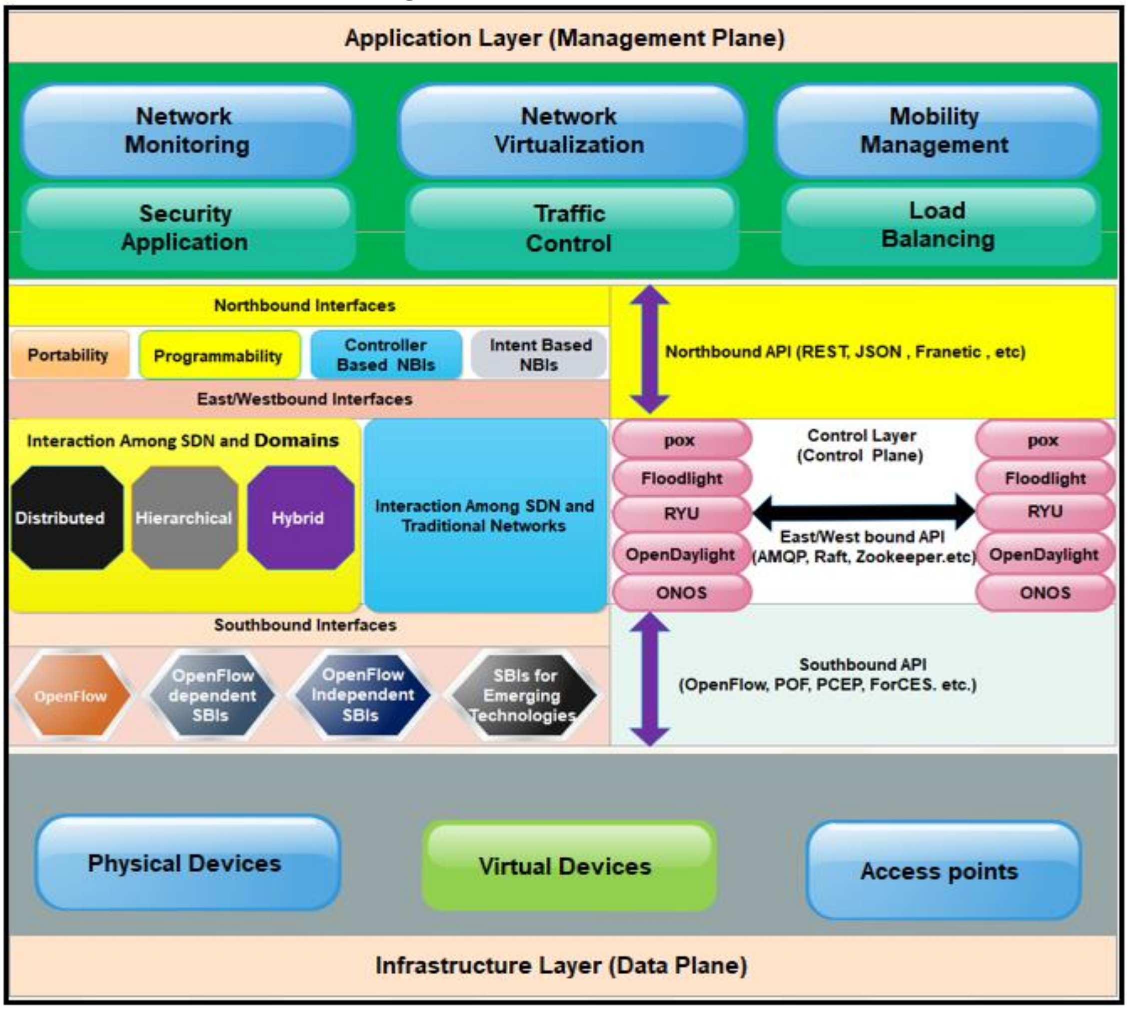

Figure 2. SDN Interfaces [27]

3.3. Open source of SDN controllers

There are a variety of Controllers and platforms to consider when choosing an SDN strategy. This provides only the introduction of profiles for open-source frameworks and consoles. We used some NOX, BOX, BEACON and FLOODLIGHT and the key features of the consoles are also presented. What should be highlighted is the comparative analysis and selection of the controller to choose the most appropriate controller according to the requirements for it [23].

4. Research methodology

The adaptability of the SDN controller could be a basic concern in such a complex environment. It is challenging for SDN arrangement because it is a central rationale to design and prepare the control plane from the redirection plane. The issue of CPP when using multiple SDN controllers is crucial in this regard. Scalability has issues of SDN-enabled cloud to accurately define its scopes and outcomes consequences. SDN's own optimization issue, particular for a multi-controller position or placement and various relevant requirements, ranges from latency imperatives to load failure tolerance and balancing to better understand their relations to controller placement problem (CPP). Scalability and reliability help to solve Multi-Controller Placement Problem (MCPP) distance and timeless between HS and PSO (HSPSO).

4.1. Proposed method procedure

The proposed optimization algorithms were used according to criteria namely symmetric Harmony Search modelling (HS) along with particle swarm optimization (PSO) of metaheuristic algorithms to build the mathematical model. HSA Scheduler builds a model to solve controller placement problem (CPP) using the following:

- Build (Cloudlet List; Virtual Machine (VM) List; Datacenter [28, 29, 47]; Execution, Common Matrix;Container; Broker later Linked List).
- Creating and sending VMs and Cloudlets to broker and starting the simulation

Table 1. Parameters VM and cloudlet

· **VM Parameters**		
Name of Parameter	value	Describe
long size = Ram = mips = long bw = pesNumber = String vmm =	10000 MB 512 MB 250 1000 1 "Xen"	Image of size Vm of memory Number Cpus VMM name
· **Cloudlet Parameters**		
Name of Parameter	Value	Describe
long file Size = long output Size = pesNumber = user	300 300 1 1	 Number of grid users

Table 2. Cloudlet and arrays of list to construct a graph

Construct a graph	
Number of Edge	value
Edge 1	(0, 1, 2)
Edge 2	(0, 2, 4)
Edge 3	(1, 2, 4)
Edge 4	(2, 0, 3)
Edge 5	(2, 1, 4)
Edge 6	(3, 2, 1)
Edge 7	(3, 2, 1)
Edge 8	(2, 4, 3)

Figure 3 below explains deployments Controllers, Switches and Hosts (C, S, H).

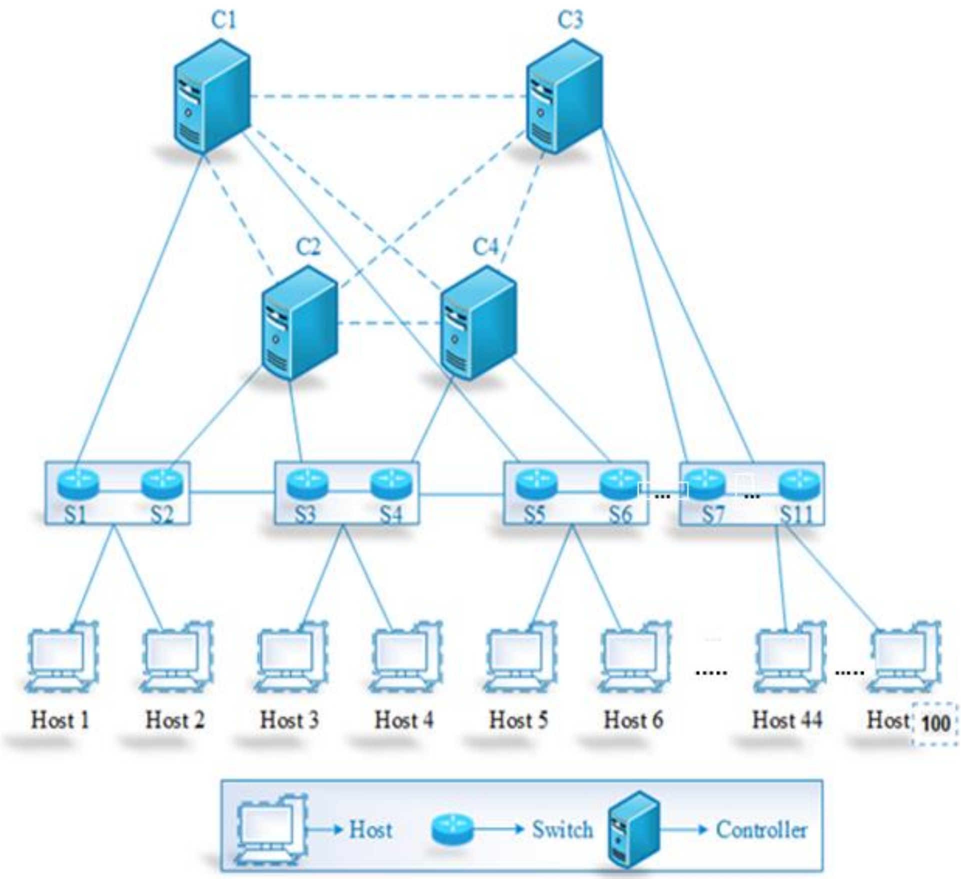

Figure 3. Hybrid multi-controller architecture model

In this section, algorithms are used to find the optimal number of controllers for the purpose of deploying them and providing a broadband network. This Table 3 is the parameter names of (HSA) from values for (HMCR, PAR, HS, BW) as steps.

Table 3. The parameters of (HSA) harmony search algorithm

Parameter names	**Parameter symbols**	**Value**
Harmony Memory Consideration Rate	HMCR	0.8
Random	Rand	0.1
Pitch Adjusting Rate	PAR	0.2
Harmony Memory	HM	5
HS iteration	H(I)	100
Distance bandwidth	BW	1000
Minimum bandwidth	MiBW	0.1
Maximum bandwidth	MBW	0.4

HSA could be a population-based metaheuristic that mimics the melodic act for an ensemble [30]. Here the mimics of the melodic act improve harmony with these three alternatives:

- Within the recorded harmony, memory can select any tone;
- They identify and settle any past score in memory, or
- They help to find a new music format in the playing field.

The method of getting this perfect solution is similar to actively exploring a perfect state of harmony [31, 32]. Figure 4 show steps of pseudocode (HSA).

Begin;

Define objective function:

$$H_i^{New} = \begin{cases} H_i(l) \in \{H_i(1), H_i(2), \dots, H_i(l)\} & R_1 > hmcr \\ H_i(l) \in \{H_i^1, H_i^2, \dots H_i^{HMS}\} & R_1 \leq hmcr \\ H_i(l) + R_3 * BW & R_2 \leq par \end{cases} \quad (1)$$

Define (HMCR), (PAR) and other parameters
Generating a HM with a random harmony
(R < max number of iterations)
(i<= number of variables)
When (rand< HMCR).
Selecting a value from harmony memory for the variable *i*
If (rand<PAR),
Adjusting the value by the addition of a definite amount
Ending if
Else
Choose a random value
End if
End while
Accepting the new harmony (solution) if better
Ending while
Finding the current best solution
End

Figure 4. Pseudocode harmony search algorithm (HSA) [33]

The use of particle swarm optimization (PSO) helps solving location optimization processes [34].

- Obj1(Average RTT vs iterations).
- Obj2(propagation latency vs iterations).
- Obj3 (Average Delay vs iterations).
- Obj4 (Cost vs iterations).
- Obj5 (Time Session vs iterations).
- Obj6 (Reliability vs iterations).

PSO is an intelligent random population-based flock analysis that simulates the social behavior of natural organisms such as crowding birds to find a place with sufficient food and is similar to teaching fish using local movements without a central control. It is designed to solve continuous dynamic improvement problems [35, 36]. Figure 5 shows the steps of Pseudocode (PSO).

***Step 1*. Initialization**
To every particle ij = 1, . . . , NP, do.
(a) The particle's position is initialized with a uniformly distribution as $Pij(0) \sim U$(UB, LB), where UB represent upper and LB depict a lower search space bound.
(b) Initialize Parameters ($pbest$, $gbest$, V and R): $pbest(ij, 0) = P_{ij}(0)$ its initial position and $gbest(0)$ = argmin $f(P_{ij}(0))$ its minimal value of the swarm for all particles ij.
(c) Initialize velocity: $v_{ij} \sim U$ (-|UB - LB|, |UB - LB|).

***Step 2*. Repeat until a termination criterion is met**

To every particle ij = 1, . . . , NP, do
(a) Pick random numbers: $R_1, R_2 \sim U(0, 1)$.
(b) Particle's velocity is updated from equation: $v_{ij}(R+1) = W(R)v_{ij}(R) +$

$$C_1 R_1 \left(P\_Best_{ij} - P_{ij}(R)\right) + C_2 R_2 (G_{Best} - P_{ij}(R)) \quad (2)$$

(c) Move to the new position: $P_{ij}(R+1) = P_{ij}(R) + v_{ij}(R+1)$ (3)
(d) If $f(P_{ij}(R)) < f(pbest(ij, R))$, do
(i) Updating the best known position of particle ij: $pbest((ij, R) = Pij(R)$.
(ii) If $f(Pij(R)) < f(gbest(R))$, updating the swarm's best known position: $gbest(R) = Pij(R)$.
(e) $R \leftarrow (R + 1)$;
(f) Calculation of weight value: $W = W_u - (W_u - W_l)\left(\frac{i}{I_{max}}\right)$ (4)

***Step 3*. Output $gbest$ (R) holding the best obtained solutions.**

Figure 5. Pseudocode (PSO) [37]

4.2. Case study with two algorithms (HS, PSO)

The selection of the optimal location for controller placement is utilized by a hybrid algorithm (HS-PSO). The PSO is utilized for HSA parameter initialization which also enhances the controller placement performance (CPP) and minimizes the latency from the controllers of the distance to the propagation delay between the switches and controllers. The proposed work achieves an improved scalability and reliability by the objective metrics according to the above processes.
This table depicts the parameter results of our new algorithm (HSPSO):

Table 4. Parameter results of (HSPSO)

Parameter names	Parameter symbols	Value
Harmony Search Particle Swarm Optimization Algorithm	HSPSO (HS, PSO)	
PSO Inertia weight (w) Acceleration factors $c1$ and $c2$	$c1 = c2$	1:49445
Pitch Adjusting Ratio	*PAR*	0.3
Maximum velocity	*V max*	20

The calculations are according to the equations below:

$$\frac{x\alpha \quad x\beta \quad x\sigma \quad \cdots xm}{TS = S1\ S2 \vdots Sn\ [TS1{,}\alpha\ TS1{,}\beta\ TS1{,}\sigma \cdots TS1{,}m\ TS2{,}\alpha\ TS2{,}\beta\ TS2{,}\sigma \cdots TS2{,}m \vdots \vdots \vdots \ddots \vdots TSn{,}\alpha\ TSn{,}\beta\ TSn{,}\sigma \cdots TSn{,}m\ ]} \quad (5)$$

- **TS** = Time Session, S1, S2, and S3, $\cdots$, Sn = numbers of switches, SPα, SPβ, and SP$\sigma\cdots$SPm = numbers of server's position.

- $Average\ Delay\ over\ Link\ = \frac{Mean\ Packet\ Size}{Available\ Link\ Bandwidth}$ (6)

The average delays between controller to controller (C2C) and controller to switch (C2S) are shown in (7) and (8)[13]:

$$D_{CC}^{mean} = \left(\sum_{i=1}^{k} \sum_{j=1}^{k} link_{dst}(SPi) \right) / (k-1) \times k \times H \quad (7)$$

Here, D_{CC}^{mean} refers the average transmission delay between (C2C), $link_{dst}$ (SPi) is the link distance between the two SP, *K* stands for the number of controllers, and *H* indicates the data transfer speed of the link.

$$D_{CS}^{mean} = \sum_{i=1}^{N} link_{dst}\big(si, B(si)\big) / (N \times H) \quad (8)$$

where, D_{Cs}^{mean}represents the mean transmission delay between (C2S); $link_{dst}$// shows the link distance between *switches*, assigned its controllers. *N* indicates the number of switches, and *H* entails the data transfer speed of the link.
The calculation of the average distances between each point in the dataset to its centroid is performed by equation (9)[24]: where $\varphi1$ and $\varphi2$ are the latitudes of point (1 , 2) , $\lambda1 and\ \lambda2$ are the longitudes of point (1 , 2) and r is the radius. S2C communication of out-of-band.

$$Distance = 2r \times arcsin\sqrt{sin^2(\tfrac{\varphi2-\varphi1}{2}) + cos(\varphi1)cos(\varphi2)sin^2(\tfrac{\lambda2-\lambda1}{2})} \quad (9)$$

Latency

Packets can be exchanged from the control unit to the switches and vise versa through latency messages in SDN, where the minimum latency can be calculated by firefly (FF) algorithms. The objective function has to be calculated by single and MCP uses of the latency formula to configure the link properties by Equation (10)[5]:

$$\text{Latency} = \frac{Number\ of\ iteration}{Number\ of\ controllers} \tag{10}$$

Latency will be measured by the ping program. The ping program measures RTT dividing the measured value into two giving about the one-way latency. The latency has four types C2C average latency, S2C worst latency, S2C average latency and average latency. Equation (11) shows the objective function of propagation latency [24]:

$$\text{Propagation Latency (sec)}= \frac{distance\ (m)}{speed(\frac{m}{sec})} \tag{11}$$

Reliability

The connection between switch and multiple controllers may prevent single points of failure and clearly increase reliability as in equation (12) [38]:

$$R = max \sum_{v \in V} \left(1 - \prod_{s \in S} \left(1 - p(v,s)\right) \right) \tag{12}$$

The mathematical reliability for *minimizing control-path* expression is (13) where *p* (*s; v*) denotes probability available to the control path. MCP (*v; s*) is the disjoint path number between (*v, c) [*38]:

$$R = max \sum_{v \in V} \sum_{s \in S} p(v,s) \tag{13}$$

Cost formula

Two types costs of energy consumption and cost deployment are discussed. The cost of the SDN is represented by mathematical equation (14):

$$\mathbf{C} = \min\ (\mathrm{Cs} + \mathrm{C}l + \mathrm{Ct}) \tag{14}$$

where C is cost, Ct refers to cost of controller interconnection C*l* denotes cost to connect S2C; Cs is the cost of controllers for minimizing the cost to plan the network.

Cost-latency trade-off analysis

Cost factor is calculated by equation (15)[24]: where C_k is the cost of deployment of a controller, k is the controller number and L_{avg} is the mean latency when *k* are deployed.

$$\textbf{Cost factor} = \frac{k \times C_k}{L_{avg}} \left(\frac{\$}{ms}\right) \tag{15}$$

Equation (16) calculated energy consumption cost [38].

$$C = \sum_{v \in V} w(v) f(v,s) \tag{16}$$

Here, f (v; s) is the communication overhead between (v, s) while w(v) is packet number [39].

5. Simulation results and discussion

In [29], Cloud Sim SDN is a simulation of hosts and networks, and requests latency usage in SDN-enabled cloud data centers. Cloud Sim SDN supports computing power which both hosts and switches consume. It is also an add-on package for Cloud Sim. Mammoth cloud suppliers such as Google have as of now embraced the SDN concept in their data center to extend manageability and adaptability [40]. For example, network-aware virtual machine mode policies can be evaluated using Cloud Sim SDN. As an example, energy savings can be introduced in a cloud data center that supports SDN via VM integration. Direct paging relies heavily on computer networks for the transference of the data and the memory states [1]. If virtual machines are combined with a minimum number of hosts, then unused hosts and switches is turned off for saving more power. There are two different policies for VM placement: best fit (MFF, most complete first) and worst fit (LFF, least) [39].

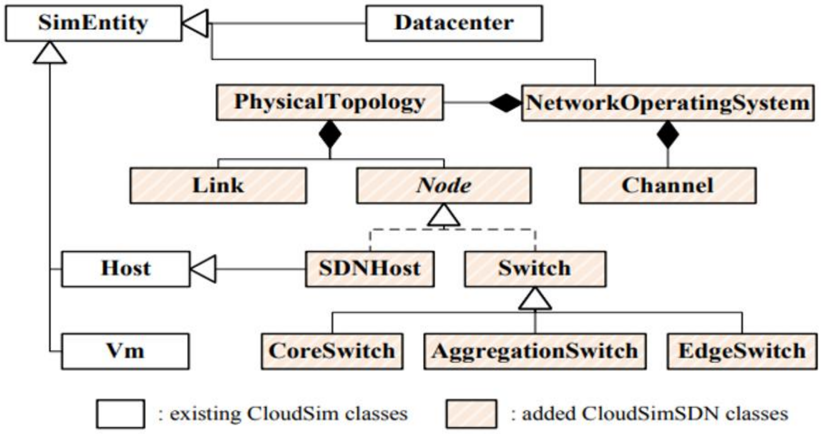

Figure 6. Cloud Sim SDN class diagram [25]

SDN controller sets up need lines (e.g., Linux) for the basic application streams on each switch at the side of the link. The organized activity is produced from VMs of the basic application [41]. SDCon (SDClouds Controller) is an integrated control stage platform such as OpenStack, VM Ware, and Xen Cloud Platform to enable orchestrated resource management [42]. There are six objectives for multi-controllers represented by figures and the best position of controllers is Nox to solve MCPP to verify or get a minimum time and distance between C2C and C2S.

Table 5. The Results parameters input and output of (PSO) algorithm

Parameters		
variable for (x) items position	i	
variable for particles	j	
Parameter names	**Parameter symbols of problem**	**Value**
Number of problem variable	*n*	5
Low Bound of decision variable	*Min*	-5.12
Upper Bound of decision variable	*Max*	5.12
Parameters of PSO		
Number of iterations	*max Iteration*	1000
Population size- swarm size	*n Pop*	50
Population Size - Particles	No. of Particles	25
Inertia Coefficient	*w*	1
Acceleration Coefficient of Personal	*c*1	1.3
Acceleration Coefficient of Social	*c*2	2.7
Damping ratio of inertia weight	*W* Damp	0.99
Output		
Random Position	2.4367301621978212, -3.9887949718658775, 4.961728085398628, 1.4826969609561251, -0.2222465685609265	
Random Position Cost	48.71466861964629	
Global Best Position (gbPos)	0.8536256345903634, -1.0534801237074873, -2.0159917965883745, -0.42002425910453844, -2.2936418269237664	
global Best Cost (gbCost)	0.18849474369081137	

There are six objectives for multi-controllers in the figures and the best position for controllers is Nox to solve MCPP verification or get a minimum time and distance between C2C and C2S: Iterations vs Propagation latency, Reliability, Cost, Average Round-Trip Time (RTT) Matrix of Time Session (TS), Average Delay are present.

Table 6. Constants and cloudlets parameters

Constants	

NO_OF_TASKS=
NO_OF_DATA_CENTERS =
POPULATION_SIZE =
	30; // Number of Cloudlets;
	5; // Number of Datacenters;
	25; // Number of Particles.

//cloudlet parameters	
file Size = output Size = pes Number =	300 300 1

1. *Number of iterations vs. propagation latency and number of iterations vs. controller reliability*

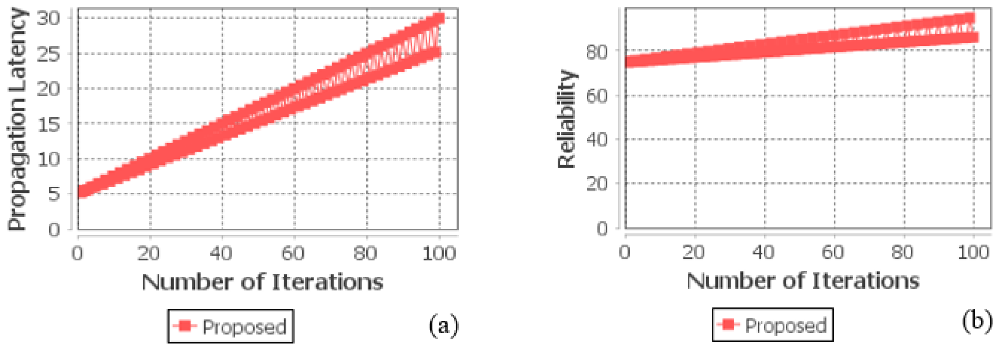

Figure 7. Graph number of iterations vs. (a) Propagation latency, (b) Reliability

2. *Number of iterations vs. (controller cost and controller average round-trip time (RTT))*

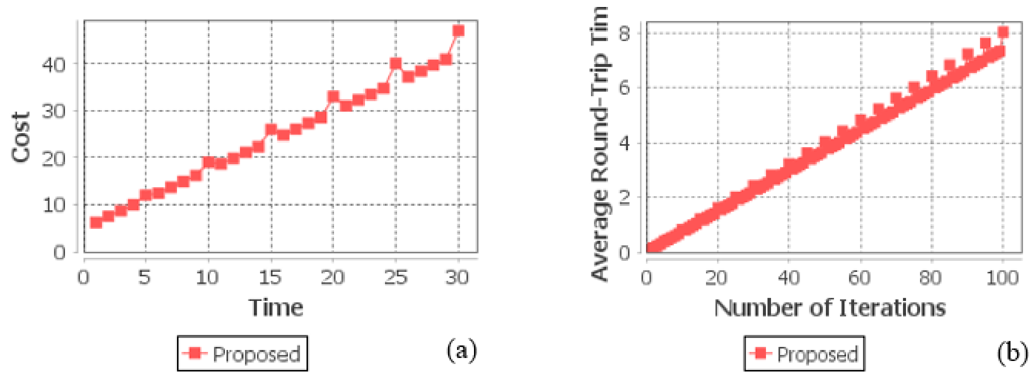

Figure 8. Graph number of iterations vs. (a) Cost, (b) Average Round-Trip Time (RTT)

3. *Number of iterations vs. controller matrix of time session (ts) and controller average delay*

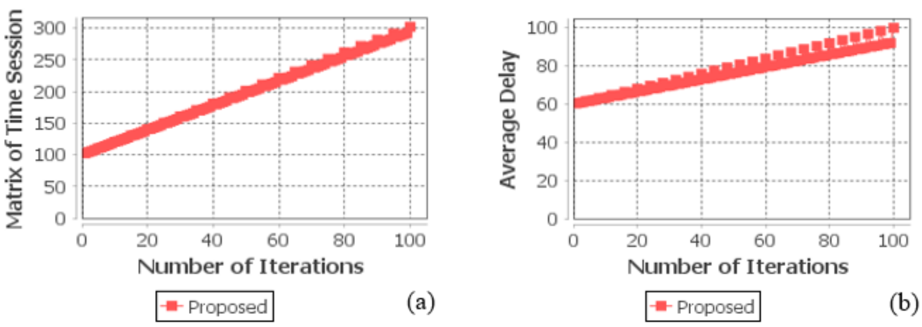

Figure 9. Graph number of iterations vs. (a) Matrix of time session (TS), (b) Average delay

Table 7. The results presented previous studies of performance measures compared to the proposed HSA-PSO method MC-SDN

Performance metrics	**Study Ref/No**	**Evaluate results previous studies**	**Evaluate results (HSA-PSO) proposed of (MC-SDN) method**
Propagation Latency	[2]	1 to 0.1 ms	11 to 20 ms
	[27]	100 ms	
	[12]	1.7 ms	
	[18]	2 ms	
	[19]	0.16 ms	
	[24]	12.90, 60.16 to 164.4 ms	
	[25]	0.5 ms	
	[26]	0.91 s	
Reliability	[22]	15.68%	88% ± 1%
	[43]	98.92%	
	[44]	99.999%	
Cost	[2]	2100	26.8 ± 0.1
	[12]	70%	
	[18]	18	
	[20]	252.5160 to 1302.9838	
	[26]	50 to 10, 000	
Round-Trip Time Average RTT	[27]	550 ms	7.3 ± 0.1 ms
	[14]	10.074 to 4.773 ms	
	[24]	52.316 to 112.67 ms	
Time Session (TS)	[21]	255.37sec	203.5 ± 0.1 sec
	[25]	1.929 to 2.009 sec	
Average Delay	[1]	6.62 ms	78.8 ± 0.1 ms
	[6]	0.000017 to 0.0000275 ms	
	[10]	23 to 60 μs	
	[13]	0.2 to 0.8 ms	
	[15]	2.8×10^4ms	
	[23]	10 to 0.4 ms	
	[24]	0.049 to 0.024 ms	
	[26]	0.05 to 5.85 s	

The algorithm and the proposed model were used with their associated objectives. Those objectives include reducing latency between the switches and their relevant controllers and that improve throughput and maximize reliability in avoiding failures and giving priority and attention to network flexibility to make the switches and controllers scalable and communicative. The algorithm also reduces the cost of deployment and power consumption by the control plane measuring the performance by connecting devices and gauges in a manner consistent with the best network performance in an SDN environment. Multiple consoles are positioned by using a metaheuristic algorithm. Also, the proposed approach indicates a superior performance to place multi-controller modes compared to other latest works. We compared the results of the experiments with the previous studies to evaluate our method and showed that the ratios of achieving the goals outperform similar trades-off between the performance of the proposed work based on several performance metrics (RTT, TS, propagation latency, delay, reliability, throughput and cost). According to the results in a table (6), it was found that the latency rate for previous studies is between (0.1 to 164 ms). As for the extracted value of the proposed model, it is between (11 to 20 ms), and thus the model is better at reducing latency. As for the reliability, it ranges between 15% to 99% and the model is 88%. So it is better than some of them and worse than others. As for the cost, it is between 15 to 9838.1302 while the cost of the proposed model is 26.8 and it is thus better in some and worse in others. RTT is between 4.773 to 112.67. The model is equal to 7.3 ms and that model is the best and the TS is 203.5. It is worse in the delay rate with a result of 78 .8 ms. Yet, the proposed (HSA-PSO) algorithm in the MC-SDN results is better in RTT, TS, reliability and cos. It is sometimes better in throughput, propagation latency. Also, the results are the worst in the average delay of the proposed (HSA-PSO) algorithm in the MC- SDN for the same metrics.

6. Conclusion and future work

SDN is constantly evolving, with new mechanisms emerging to solve scalability issues by taking full advantage of the programmability of controllers without sacrificing performance and management capabilities. We present a meta-heuristic algorithm to solve multi-controller scalability in SDN through Harmonious Search Modeling (HSA) coupled with Particle Swarm Optimization (PSO). The proposed approach provides an improved solution for scaling SDN control across multiple nodes, while at the same time providing a sufficient flexibility to deal with periodic or ring-containing network structures. The design is supported by the use of the state space description and the SDN controller model. CPP is the basic for SDN scaling and management and the goal of console mode optimization is to improve performance and scalability. This reflection gives a brief presentation to SDN, a development and an architecture of multiple (distributed) controllers in terms of deployment. Also, the necessary applications of SDN have important advantages and characteristics. The proposed solutions for CPP in this direction aim to ensure scalability and reliability by obtaining the results of the current study. There are future visions for treatments. In addition, there are more future studies on this subject which are necessary to identify the various issues and directions for research that can be discussed, especially the priorities and subsequent directions thereof. It is still early days to SDN. Cloud-hosted control planes are being conveyed to generation systems, where it is the beginning to see SDN tracking in getting to systems and programmable pipelines utilized to convey unused information level usefulness. As the innovation develops and APIs stabilize, we anticipate seeing an expanded adoption of use cases affecting the role SDN plays. It is troublesome to form vigorous networks and to secure verifiably against disappointments, assaults, and setup blunders. In spite of the progress in security at the application level and the progresses in making computer systems more programmable, most networks are still built utilizing closed, restrictive software and complex firmware. The emergence of 5G networks and applications means connecting smartphones and people. This emergence also helps to meet the need for bells and doors to lights, fridges, drone and self-driving cars. Each packet in the network must follow and validate the path set by the operator and only encounter a set of forwarding rules within each trace intended device. The Mobile network SDNs are represented in the first SDN flow table model and central logical control. The default network route management and route management are Wi-Fi, 3G and 4G [26, 2]. Then 5G and 6G are the wireless communication systems [45]. The birth of SDN-based 6G -SAGIN space–air-ground Integrated Network intelligent is hierarchical too [46].

Declaration of competing interest

The authors declare that they have no any known financial or non-financial competing interests in any material discussed in this paper.

Funding information

No funding was received from any financial organization to conduct this research.

References

[1] T. He, "Migration Management in Software-Defined Networking-enabled Edge and Cloud Computing Environments," degree of Doctor of Philosophy, School of Computing and Information Systems, THE UNIVERSITY OF MELBOURNE, ORCID: 0000-0002-5472-7681, 2021.

[2] X. You, et al., "Towards 6G wireless communication networks: vision, enabling technologies, and new paradigm shifts," *Science China Information Sciences,* vol. 64, no. 1, 2020.

[3] N. A. Jasim, H. TH, and S. A. Rikabi, "Design and Implementation of Smart City Applications Based on the Internet of Things," *International Journal of Interactive Mobile Technologies,* vol. 15, no. 13, pp. 4-15, 2021.

[4] A. A. Abbasi, A. Abbasi, S. Shamshirband, A. T. Chronopoulos, V. Persico, and A. J. I. A. Pescapè, "Software-defined cloud computing: A systematic review on latest trends and developments," vol. 7, pp. 93294-93314, 2019.

[5] S. P. Mythrayee Ramasamy, "Single and Multi-Type Controllers with Soft Computing Methods and Routing in Software Defined Network," *International Journal of Engineering and Advanced Technology (IJEAT),* vol. Volume-9 no. Issue-1S5, p. 7, December, 2019 2019.

[6] S. Harsh, "Evaluation of Multiple Controller based Software Defined Networks Architecture over Single Controller Software Defined Architecture," Master, Engineering Technology, Houston, 2016.

[7] H. Chen, E. Reid, J. Sinai, A. Silke, and B. Ganor, *Terrorism informatics: Knowledge management and data mining for homeland security*. Springer Science & Business Media, 2008.

[8] N. Alseelawi, and H. T. Hazim, "A Novel Method of Multimodal Medical Image Fusion Based on Hybrid Approach of NSCT and DTCWT," *International Journal of Online Biomedical Engineering,* vol. 18, no. 3, 2022.

[9] Y. S. Mezaal and S. F. Abdulkareem, "Affine cipher cryptanalysis using genetic algorithms," *JP Journal of Algebra, Number Theory Applications*, vol. 39, no. 5, pp. 785-802, 2017.

[10] Y. A. J. e. Agudelo, "Scalability and Robustness of the control plane in Software-Defined Networking (SDN)," phd thesis, Network Engineering Department, Universitat Politecnica de Catalunya, Barcelona, 2016.

[11] E. Q. Ahmed, I. A. Aljazaery, and A. F. Al-zubidi, "Design and implementation control system for a self-balancing robot based on internet of things by using Arduino microcontroller," *Periodicals of Engineering Natural Sciences*, vol. 9, no. 3, pp. 409-417, 2021.

[12] B. Sadun, M. S. Obaidat, D. Puthal, B. Sahoo, S. K. Mishra, A. Sarkar, and K. Sagar Sahoo, "Metaheuristic Solutions for Solving Controller Placement Problem in SDN-based WAN Architecture," presented at the Proceedings of the 14th International Joint Conference on e-Business and Telecommunications, 2017.

[13] Y. Li, S. Guan, C. Zhang, and W. Sun, "Parameter Optimization Model of Heuristic Algorithms for Controller Placement Problem in Large-Scale SDN," *IEEE Access,* vol. 8, pp. 151668-151680, 2020.

[14] M. Z. Abdullah, N. A. Al-awad, and F. W. Hussein, "Evaluating and Comparing the Performance of Using Multiple Controllers in Software Defined Networks," *International Journal of Modern Education and Computer Science*, vol. 11, no. 8, pp. 27-34, 2019.

[15] S. Hans, S. Ghosh, A. Kataria, V. Karar, S. J. C.-C. M. Sharma, and CONTINUA, "Controller Placement in Software Defined Internet of Things Using Optimization Algorithm," vol. 70, no. 3, pp. 5073-5089, 2022.

[16] H. T. ALRikabi and H. T. Hazim, "Enhanced Data Security of Communication System Using Combined Encryption and Steganography," *International Journal of Interactive Mobile Technologies*, vol. 15, no. 16, 2021.

[17] Y. S. Mezaal and A. S. Al-Zayed, "Design of microstrip bandpass filters based on stair-step patch resonator," *International Journal of Electronics,* vol. 106, no. 3, pp. 477-490, 2019.

[18] A. A. Ateya, A. Muthanna, A. Vybornova, A. D. Algarni, A. Abuarqoub, Y. Koucheryavy, and A. Koucheryavy, "Chaotic salp swarm algorithm for SDN multi-controller networks," *Engineering Science and Technology, an International Journal,* vol. 22, no. 4, pp. 1001-1012, 2019.

[19] A. K. Singh, S. Maurya, and S. J. F. o. C. S. Srivastava, "Varna-based optimization: a novel method for capacitated controller placement problem in SDN," vol. 14, no. 3, pp. 1-26, 2020.

[20] F. Misni, L. Lee, and N. Jaini, "Multi-objective hybrid harmony search-simulated annealing for location-inventory-routing problem in supply chain network design of reverse logistics with CO2 emission," in *Journal of Physics: Conference Series,* 2021, vol. 1988, no. 1, p. 012054: IOP Publishing.

[21] M. S. Rosle, M. S. Mohamad, Y. W. Choon, Z. Ibrahim, A. González-Briones, P. Chamoso, and J. M. J. P. Corchado, "A Hybrid of Particle Swarm Optimization and Harmony Search to Estimate Kinetic Parameters in Arabidopsis thaliana," vol. 8, no. 8, p. 921, 2020.

[22] A. Alowa, "Scalable ReliableControllerPlacementinSoftwareDefinedNetworking," Doctor of Philosophy Computer Science and Software Engineering, Concordia University, 2020.

[23] D. Sakellaropoulou and Business, "A qualitative study of SDN controllers," 2017.

[24] L. Mamushiane, "Towards the Development of an Optimal SDN Controller Placement Framework to Expedite SDN Deployment in EmergingMarkets," MS thesis, Electrical Engineering, University of Cape Town (UCT), 2019.

[25] J. Son, A. V. Dastjerdi, R. N. Calheiros, X. Ji, Y. Yoon, and R. Buyya, "Cloudsimsdn: Modeling and simulation of software-defined cloud data centers," in *2015 15th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, 2015, pp. 475-484: IEEE.

[26] A. Kak, "Towards 6G Through SDN and NFV-Based Solutions for Terrestrial and Non-Terrestrial Networks," Georgia Institute of Technology, 2021.

[27] L. Zhu, M. M. Karim, K. Sharif, F. Li, X. Du, and M. J. a. p. a. Guizani, "SDN controllers: Benchmarking & performance evaluation," 2019.

[28] building-intelligence-into-cloud. Available: https://www.intel.com/content/dam/www/central-libraries/us/en/documents/telemetry-building-intelligence-into-cloud-white-paper.pdf,2020.

[29] R.-H. Hwang, H.-P. Tseng, and Y.-C. Tang, "Design of SDN-Enabled Cloud Data Center," presented at the 2015 IEEE International Conference on Smart City/SocialCom/SustainCom (SmartCity), 2015. Available: https://ieeexplore.ieee.org/document/7463848/

[30] S.-J. Kim, S.-E. Yoo, B.-J. Lee, K.-T. Kim, and H.-Y. Youn, "A Load Balancing Scheme for Distributed SDN Based on Harmony Search with K-means Clustering," in *Proceedings of the Korean Society of Computer Information Conference*, 2019, pp. 29-30: Korean Society of Computer Information.

[31] F. Misni, L. S. Lee, and H.-V. J. A. S. Seow, "Hybrid harmony search-simulated annealing algorithm for location-inventory-routing problem in supply chain network design with defect and non-defect items," vol. 10, no. 18, p. 6625, 2020.

[32] T. A. Shaikh and R. J. I. i. M. U. Ali, "An intelligent healthcare system for optimized breast cancer diagnosis using harmony search and simulated annealing (HS-SA) algorithm," vol. 21, p. 100408, 2020.

[33] X.-S. Yang, "Harmony search as a metaheuristic algorithm," in *Music-inspired harmony search algorithm*: Springer, 2009, pp. 1-14.

[34] M. Tubishat, S. Ja'afar, M. Alswaitti, S. Mirjalili, N. Idris, M. A. Ismail, and M. S. Omar, "Dynamic Salp swarm algorithm for feature selection," *Expert Systems with Applications*, vol. 164, 2021.

[35] E.-G. Talbi, *Metaheuristics: from design to implementation*. John Wiley & Sons, 2009.

[36] A. Vybornova, "A Survay on the Swarm Intelligence Approaches to Controller Placement Problem in the Software Defined Networks Design and Optimization," *Telecom IT,* vol. 8, no. 4, pp. 83-92, 2020.

[37] Y. Zhang, S. Wang, and G. Ji, "A comprehensive survey on particle swarm optimization algorithm and its applications," (in English), *Mathematical Problems in Engineering,* Article 2015 Annual// 2015.

[38] J. Lu, Z. Zhang, T. Hu, P. Yi, and J. Lan, "A Survey of Controller Placement Problem in Software-Defined Networking," *IEEE Access,* vol. 7, pp. 24290-24307, 2019.

[39] G. Li, X. Wang, and Z. Zhang, "SDN-Based Load Balancing Scheme for Multi-Controller Deployment," *IEEE Access,* vol. 7, pp. 39612-39622, 2019.

[40] J. Son, A. V. Dastjerdi, R. N. Calheiros, and R. Buyya, "SLA-Aware and Energy-Efficient Dynamic Overbooking in SDN-Based Cloud Data Centers," *IEEE Transactions on Sustainable Computing,* vol. 2, no. 2, pp. 76-89, 2017.

[41] J. Son and R. Buyya, "Priority-Aware VM Allocation and Network Bandwidth Provisioning in Software-Defined Networking (SDN)-Enabled Clouds," *IEEE Transactions on Sustainable Computing,* vol. 4, no. 1, pp. 17-28, 2019.

[42] J. Son, R. J. Buyya, and D. Systems, "SDCon: Integrated control platform for software-defined clouds," vol. 30, no. 1, pp. 230-244, 2018.

[43] G. Schütz, "A k-cover model for reliability-aware controller placement in software-defined networks," in *international conference on computational science*, 2019, pp. 604-613: Springer.

[44] A. K. Singh, S. Maurya, N. Kumar, and S. J. Srivastava, "Heuristic approaches for the reliable SDN controller placement problem," vol. 31, no. 2, p. e3761, 2020.

[45] A. A. Z. Ibrahim and F. Hashim, "An architecture of 5G based on SDN NV wireless network," *Indonesian Journal of Electrical Engineering and Computer Science,* vol. 14, no. 2, 2019.

[46] Z. Liao, C. Chen, Y. Ju, C. He, J. Jiang, and Q. Pei, "Multi-Controller Deployment in SDN-Enabled 6G Space–Air–Ground Integrated Network," vol. 14, no. 5, p. 1076, 2022.

[47] Y.S. Mezaal, H.H. Madhi, T. Abd, S.K. Khaleel, "Cloud computing investigation for cloud computer networks using cloudanalyst", Journal of Theoretical and Applied Information Technology, vol.96, no.20, 2018.