Vol. 29 计算机集成制造系统 ISSN
No. 5 Computer Integrated Manufacturing Systems 1006-5911

# MULTIVARIATE COMBINATORIAL ELITISM GOLDEN EAGLE OPTIMIZATION FOR CONTROLLER PLACEMENT IN SDN ENVIRONMENT

**J. Hemagowri**[*1] **& P. Tamil Selvan**[2]
[*1&2]Department of Computer Science, Karpagam Academy of Higher Education, Coimbatore 641 021, Tamil Nadu, India

**Abstract:**
SDN is the model which integrates control plane over data plane. It monitors as well as controls network with the help of a controller. Numerous controllers were requirement of the SDN-based wireless network. Multiple controllers are termed by CPP. CPP focused on latency however unnoticed server under dynamic switches. In this paper, novel Multivariate Combinatorial Elitism Golden Eagle Controller Placement Optimization (MCEGECPO) technique is developed with better resource capacity of controllers. By applying a Combinatorial Elitism Golden Eagle Optimization, a controller placement is performed based on Multivariate functions. Firstly, the populations of golden eagles (i.e. controllers) are initialized in the graph structures. For each eagle, the fitness is estimated along with the Multivariate functions. The Elitism selection is applied to Golden Eagle Optimization to randomly select the controllers with the best fitness. Followed by, the global optimum solution is determined based on the position updates. As a result, the overall network performance is improved and obtains the delay. Experimental evaluation of MCEGECPO as well as existing techniques are conducted with various parameters such as packet delivery ratio, packet drop rate, throughput, average latency, and execution time. Experimental assessment shows MCEGECPO enhances packet delivery as well as throughput and minimizes latency, packet drop, and execution time compared with conventional methods.

**Keywords:** Controller Placement, Multivariate functions, Combinatorial Elitism Golden Eagle Optimization.
**DOI:** 10.24297/j.cims.2023.5.9

## 1. Introduction

In SDN, numerous controllers were positioned and every controller is reasonably centralized for organizing switches. It influences the network. In SDN, a challenge arises for reducing controllers deployed in distributed network. PHCPA was designed in [1] to minimize the propagation delay by implementing the two optimizations such as MRFO and SS). But it failed for utilizing parameters using latency for Controller Placement. GSOCCPP was introduced in [2] to achieve lesser delay. But GSOCCPP technique was not focused on considering the various metrics to solve the controller placement.

Multiple-criteria decision-making (MCDM) models were introduced in [3] for solving the CPP. However, it was unsuccessful for tackling dynamic controller placement to meet reliable low-latency communications. A nonlinear optimization model, based on GA was proposed in [4] for discovering specified number of controllers for decreasing the control overhead.

A new MOGA using Variant Particle Swarm Optimization were developed in [5] for controller placement. However, it failed to implement the newer technologies with multi-objective algorithms. A Fuzzy C-Means for Controller Placement (FCMCP) approach was introduced in [6] to reduce the latency. However, it failed to estimate various delays of transmission to improve the performance. A joint optimization model was developed in [7] for finding the exact location of controller placement. But it was not able to determine the optimal solution.

CPP-MLF was solved in [8]. However, it failed to improve the heuristic algorithm for reducing overheads. Heuristic multi-objective optimization approach was developed in [9] using a DCCP. However, the execution time of the optimization algorithm was not minimized. A VBO were developed in [10] to consistent controller placement that reduces the total average latency. However, it failed to consider multiple constraints of controller placements.

DRMDOCP was solved [21] for placing an optimal number of controllers to enhance the network performance with different topologies.

## Contribution of the research work

The major issues reviewed by the literature were overcome with developing MCEGECPO technique. Major contribution of proposed MCEGECPO is explained below,

- MCEGECPO technique was developed to solve the CPP with multivariate functions.
- A Combinatorial Elitism Golden Eagle Optimization is applied to an MCEGECPO technique to find the optimum controller based on multicriteria functions.
- A Combinatorial function was utilized for evaluate the fitness for finding controller with minimum as well as a maximum ability. It refers to the minimum propagation latency and maximum server capacity.
- Elitism selection strategy was used for minimizing the complexity of the algorithm (i.e. execution time) for finding the optimal solution by removing the controllers with the worst fitness. As a result, an optimal controller is identified to enhance data delivery as well as reduce packet loss network hence it increases the throughput and minimizes the latency.
- Finally, a simulation test is conducted with the network topology to validate the performance of the MCEGECPO technique and existing methods. The observed results of the MCEGECPO technique are discussed with different parameters.

**Organization of the paper**

The structure of article is summarized by.  Section 2 explains literature works are discussed. Section 3 briefly describes the proposed methodology MCEGECPO technique. Section 4 describes simulation settings using network topology. Section 5 gives simulation results with different metrics. Section 6 provides the conclusion of article.

## 2. Literature Review

A novel optimization technique was designed in [11] to resolve the latency-aware controller placement issue. But it was not efficient to achieve higher throughput. A fault-tolerance metaheuristic-based approach was introduced in [12] for controller placement. However, the performance latency was not effectively minimized.

POM was implemented in [13] for heuristic algorithm applied to CPP.  Heuristic method was developed in [14] to solve controller placement problem. But the network cost was high. An integer-programming formulation was designed in [15] to deal with controller placement problems for enhancing network survivability as minimizing network. However, it failed for heuristic algorithms to perform optimal solution. A joint optimization approach was introduced in [16] for controller placement to minimize the delay and availability constraints.  However, it failed to develop an efficient heuristic for minimizing the complexity of controller placement.

Analytical Network Process (ANP) implemented to MCDM was developed in [17] to controller placement. But the load balancing capabilities of the network were not considered. A static and dynamic controller placement was developed in [18] for reducing costs. But the optimal placement was not achieved. Non-dominated sorting moth flame controller placement optimization was presented in [19] for performing link load balancing. However, the efficiency of the optimization framework was not improved. A kernel search introducing integer programming was developed in [20] for solving CPP. However, it failed for tackling dynamic CPP.

## 3. Proposal Methodology

From software-defined networking (SDN) structural design, solving CPP was the process of determining exact controllers as well as mapping the association among controllers to reduce the control overhead in the network. Numerous controllers placement in the SDN simultaneously increases the delays between devices and the load imbalance among devices also gets increased. Based on this motivation, the controller placement problem is a nonlinear optimization model and solves the issue via a Multivariate Combinatorial Elitism Golden Eagle Controller Placement Optimization. MCEGECPO technique helps to identify the minimum controllers as well as optimal position.

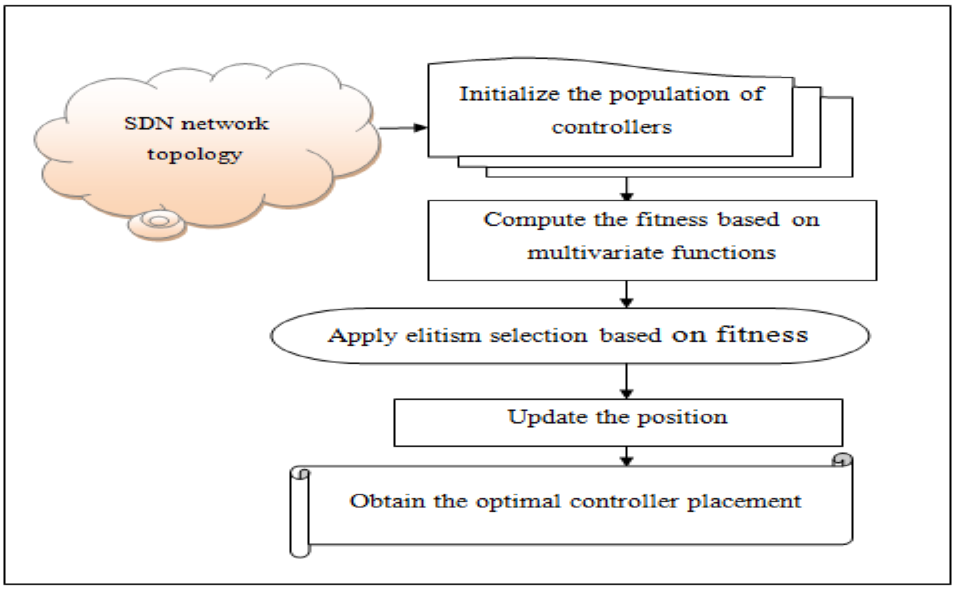
SDN network topology
Initialize the population of controllers
Compute the fitness based on multivariate functions
Apply elitism selection based on fitness
Update the position
Obtain the optimal controller placement

**Fig. 1** Architecture of the Multivariate Combinatorial Elitism Golden Eagle Controller Placement Optimization (MCEGECPO)

Figure 1 shows MCEGECPO for solving controller placement problems by applying a Multivariate Combinatorial Elitism Golden Eagle optimization. To improve the speed of data transmission in SDN network topology, an optimum position of the controller is identified based on multivariate functions. The proposed MCEGECPO technique is modeled as an undirected graphical model.

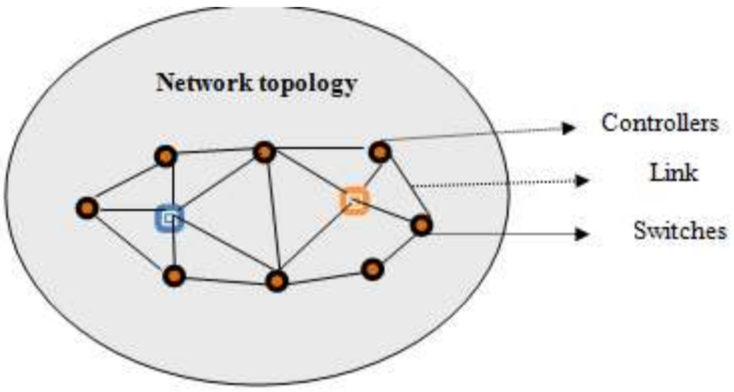

**Fig. 2** Network Model

Figure 2 describes proposed MCEGECPO technique is discussed with three major entities namely controller, switches ad links. CPP is summarized within undirected graphical scheme $G(S,E)$ and $G$ represents the graph, $S$ indicate a set of wireless network devices i.e. switches and $E$ specifies a set of the link between switches.

By applying a Combinatorial Elitism Golden Eagle Optimization, the number of controllers $C = \varphi_1, \varphi_2, \dots \varphi_n$ is positioned and determines the efficient one.

Golden Eagle Optimization worked on basis of golden eagles. Proposed optimization divides the golden eagles based on their hunting behavior such as attack and cruise. Attack vector of every eagle is called as search agent to find its food at the current location and ends its movement at the location of prey. The prey is the best location for each golden eagle. Here, the Golden Eagles are related to the controllers, and the prey is denoted as the optimal location of the controllers in the network.

The main advantage of Golden Eagle Optimization is to find the optimal location of prey in search space in the lesser possible time. The proposed Golden Eagle Optimization includes a combinatorial function that deals with both a minimization problem as well as a maximization problem, depending on whether the given objective function (i.e. fitness function) is to be minimized or maximized. Elitism strategy is applied for selecting the limited number of individuals (i.e. Eagles) along with the best fitness values to pass to the next process resulting it minimizing the complexity of the algorithm and it also speeds up the convergence of the algorithm.

First, the proposed Combinatorial Elitism Golden Eagle Optimization begins for initializing population of '$n$' number of golden eagles '$C = \varphi_1, \varphi_2, \dots \varphi_n$ (i.e. SDN).

$$C = \varphi_1, \varphi_2, \dots \varphi_n \quad (1)$$

Behind initialization, fitness was evaluated based on Multivariate functions. It means that involving the multiple dependent variables resulting in one outcome (i.e. optimum results).

- **Propagation latency**

It is estimated as number of time consumed to transmit data from the controller for the switches in an SDN topology.

$$\alpha_{lat} = time\ [TD] \quad (2)$$

From (2), $\alpha_{lat}$ indicates propagation latency, $TD$ designates a transmission of data from the controller to the switches in the SDN network.

- **Load balancing capability ($\boldsymbol{\alpha_{load}}$)**

The Load balancing capability is the major factor that influences data transmission fover controller. It is calculated by proportion of average load for maximum demand.

$$L_f = \left[\frac{Avg_L}{Max_D}\right] \quad (3)$$

From (3), $L_f$ designates load factor, $Avg_L$ symbolizes average load, $Max_D$ indicates a maximum demand. Load factor minimal than '1' indicated by controller is higher load balancing capability.

- **Bandwidth**

It was a significant factor during data transmission. It is referred by maximum rate of data send within particular number of time over controller.

$$\alpha_{bw} = \left[\frac{Max\ D\ (bits)}{S}\right] \quad (4)$$

From (4), $\alpha_{bw}$ indicates a Bandwidth, $Max\ D\ (bits)$ represents maximum rate of data transfer in the unit bits, $S$ indicates time in second. Hence, overall bandwidth was computed by Mbps or Gbps.

- **Fault tolerance**

It is a capability of controller for maintaining the operation accurately when a failure occurs. It is measured on the ratio of the number of failures occurred for total time as follows,

$$R_f = \frac{Number\ of\ failures}{Total\ operating\ time} \quad (5)$$

From (5), $R_f$ indicates a failure rate. When the minimum failure rate has occurred, controller is higher.

- **Data transmission rate**

It is the significant metric that referred by a number of data transferred from channel in time. Data transmission rate are expressed by bits/s.

- **Server capacity:**

Server capacity defines the amount of memory capacity '$\alpha_{SC}$' of the controller. Therefore, it is formulated as given below,

$$\alpha_{SC} = Mem(C) \quad (6)$$

Where $Mem$ denotes the memory capacity of the controller. It is measured in terms of Megabytes (MB).

Based on the above-said resources estimation, the optimal location of controller placement is identified with the help of the fitness measure. The fitness is estimated by applying the combinatorial function as given below,

$$F = [arg\ min\ \alpha_{lat}]\ \&\&\ [\ arg\ max\ \{\alpha_{load}, \alpha_{bw}, \alpha_{ft}, \alpha_{DTR}\ , \alpha_{SC}\ \}] \quad (7)$$

Where, $F$ denotes fitness function, $\arg min$ indicates argument of minimum function, $\arg\max$ indicates argument of maximum function. Combinatorial function satisfies both minima as well as maximum functions.

After that, the elitism selection technique is applied for finding the best individuals (i.e. golden eagles) among the populations on fitness evaluation by setting the threshold.

$$Y = \begin{cases} F > T\ ; & select\ individuals \\ Otherwise; & Reject\ the\ individuals \end{cases} \quad (8)$$

Where $Y$ denotes an elitism selection outcome, $T$ denotes a threshold, $F$ indicates a fitness.

Based on the fitness value, different behaviors of the Golden Eagle such as Prey selection, Exploitation, exploration, and position updates are estimated.

### a. Prey selection

For each eagle in search space, prey is randomly selected. Prey is a fundamental behavior of the eagle that helps to search the food source. The eagle concurrently is attraction with attacking prey as well as cruise to food (i.e. best fitness). For each iteration, the golden eagle ought to select prey for executing cruise and attack. From proposed optimization technique, each eagle maintains the memory in which the location of prey is stored into different blocks. In this technique, location of prey is assigned.

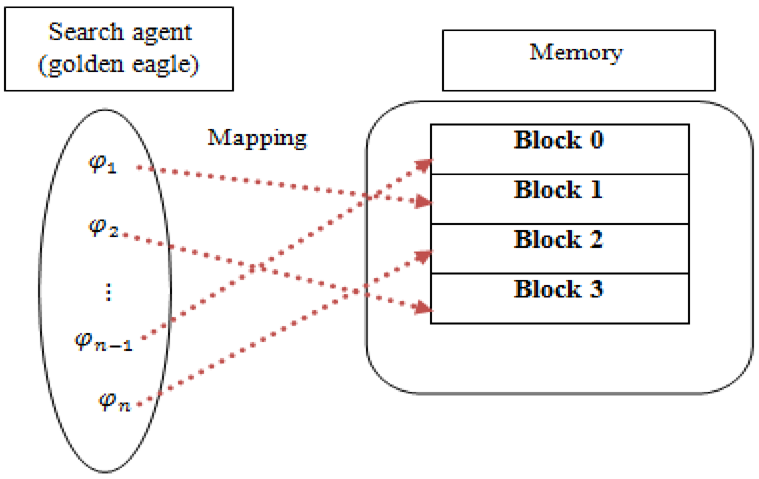

**Fig. 3** Direct cache mapping

Figure 3 illustrates the process of mapping the input (Search agent) into the memory for selecting the prey. Therefore, the mapping process is given below,

$$F: \varphi_i \rightarrow b_i \quad (9)$$

Where, $F$ denotes a mapping function, $\varphi_i$ denotes an input (Search agent), $b_i$ denotes memory blocks that contain the location of prey.

### b. Exploitation (attack vector)

It is formed using a vector starting over current position of eagle as well as ending location of prey. The attack vector for the golden eagle is formulated as given below,

$$v_a(\varphi_i) = p_f(\varphi_i) - p_t(\varphi_i) \quad (10)$$

Where, $v_a(\varphi_i)$ denotes an attack vector, $p_f(\varphi_i)$ denotes best location (prey), $p_t(\varphi_i)$ indicates current position of eagle.

### c. Exploration (Cruise vector)

The cruise is another part of golden eagle optimization that is computed on attack vector. It is vertical to attack vector as well as the tangent vector to circle. The destination point of the golden eagle on a cruise is given below,

$$\vartheta_d = \frac{D - \sum_{j \neq k} A_j}{A_k} \quad (11)$$

Where,

$$D = \sum_{i=1}^{n} \delta_i \beta_i \quad (12)$$

Where, $\vartheta_d$ indicates a destination point of the golden eagle, $A_j$ denotes $j$-th of attack vector, $A_k$ indicates $k$ -th element of attack vector, $D$ indicates n-dimensional space, $\delta_i$ denotes a normal vector, the variables vector '$\beta_i$',

**d. Position update:**

The positions of golden eagles updating depend on attack as well as cruise vectors. Calculate step vector for golden eagle $(\varphi_i)$ '$t$' as given below,

$$\nabla p_i = R_1 c_a \frac{v_a\,(\varphi_i)}{|v_a\,(\varphi_i)|} + R_2 c_c \frac{v_c\,(\varphi_i)}{|v_c\,(\varphi_i)|} \quad (13)$$

Where, $\nabla p_i$ denotes a step vector for golden eagle, $R_1, R_2$ are the random values lie in the interval [0, 1]. $c_a$, $c_c$ denotes an attack coefficient and cruise coefficient, $v_a\,(\varphi_i)$ attack vector value, $v_c\,(\varphi_i)$ denotes a cruise vector.

.

$$p_{t+1}(\varphi_i) = p_t(\varphi_i) + \nabla p_i \quad (14)$$

Where, $p_{t+1}(\varphi_i)$ denotes an updated position of the eagle, $p_t(\varphi_i)$ indicates a current position of the eagle, $\nabla p_i$ denotes a step vector. Again, the fitness is evaluated for the newly updated position of an eagle. If fitness of novel position of golden eagle is higher than old position, then it replaces new position in the memory of this eagle.

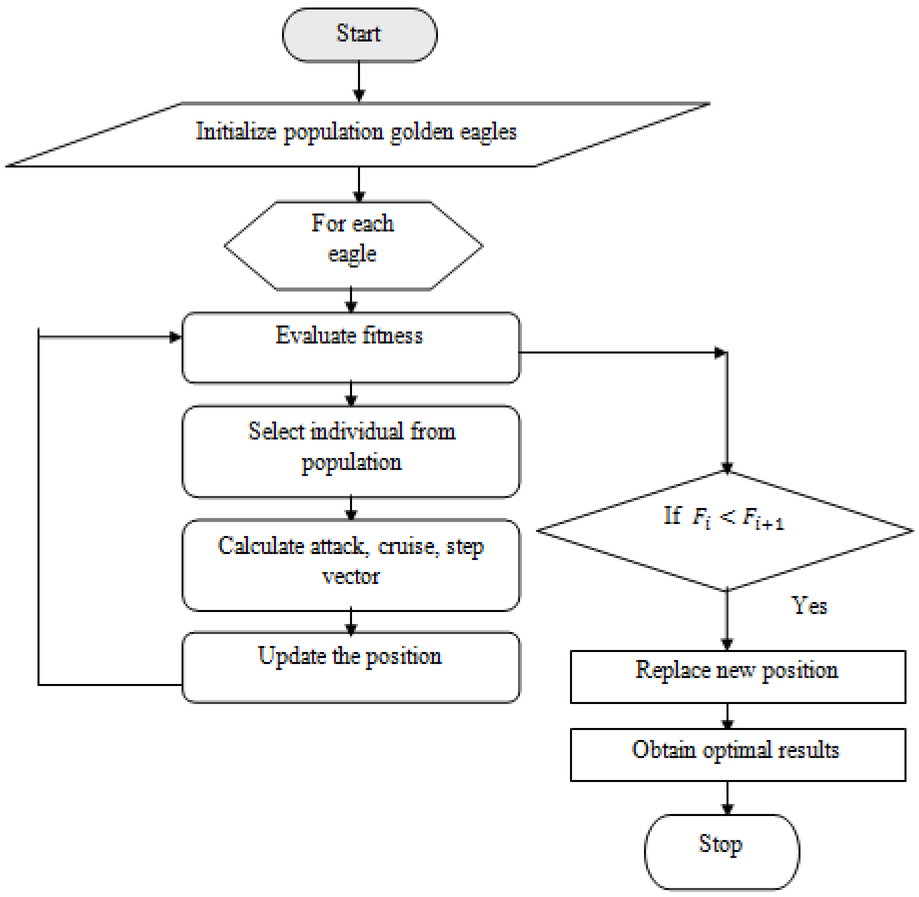

**Fig.**4 Flowchart of the Combinatorial Elitism Golden Eagle Optimization

Figure 4 illustrates the flowchart of the Combinatorial Elitism Golden Eagle Optimization to find the best optimal controller for improving the data transmission. The algorithmic process of Combinatorial Elitism Golden Eagle Optimization is given below,

```
// Algorithm 1 Combinatorial Elitism Golden Eagle Optimization
Input: Dataset, Number of controllers $C = \varphi_1, \varphi_2, \dots \varphi_n$ ;
Output: Find optimal controllers for placement
Begin
    1.  Initialize the current population of $C = \varphi_1, \varphi_2, \dots \varphi_n$
    2.  Calculate the fitness ' $F$' based on multivariate functions
    3.    Apply elitism selection
    4.  if ($F > T$) then
    5.        Select current best 'n' controllers
    6.  else
    7.        Remove the controllers
    8.  end if
    9.    For each iteration 't'
    10.       For each selected golden eagle '$\varphi_i$'
    11.         Randomly select a prey from the population's memory
    12.          Calculate attack vector '$v_a(\varphi_i)$'
    13.   if attack vector's length ($v_a(\varphi_i)$) is not equal to zero
    14. Calculate cruise vector $v_c(\varphi_i)$
    15.     Calculate step vector $\nabla\varphi_i$
    16.         Update the position ($\varphi_{t+1}(\varphi_i)$)
    17.           Evaluate fitness function for the new position
    18.  if ($F_i < F_{i+1}$) then
    19.   Replace the new position in eagle i's memory
    20.   Replace the old controllers into the current best
    21.  end if
    22. end if
    23. end for
    24. end for
    25. Return (best optimal solution)
End
```

The above algorithmic process of Combinatorial Elitism Golden Eagle Optimization-based controller placement is described in SDN network. The current population of controllers they were initialized randomly. Fitness function is calculated based on multivariate functions. Then the current best 'n' controllers are selected by applying the elitism function. If the fitness of old position ($F_i$) is better than fitness of new position ($F_{i+1}$), then replace new position in eagle memory. Obtain the best optimal solution. The entire process is repeated until the maximum iteration gets reached. Finally, the optimal controller is obtained for improving the data delivery.

## 4. Simulation Scenario

In this section, a simulation of the MCEGECPO and existing PHCPA [1], and GSOCCPP [2] is implemented by NS-2 with GBN network topology dataset. This dataset comprises simulation results of Bandwidth transmitted, total number of data packets transferred, the entire number of packets dropped, Average per-packet delay, and Variance of jitter from packets transferred from every source-destination pair. Simulation parameter settings are presented in Table I.

**Table 1:** Simulation Parameter

| Simulation Parameters | Values |
|---|---|
| Network Simulator | NS2.34 |
| Number of nodes (switches) in GBN network topology | 17 |
| Number of data packets | 30,60,90,120,150,180,210,240,270,300 |
| Number of controllers | 5 |
| Simulation time | 300sec |
| Protocol | DSR |
| Number of runs | 10 |

## 5. Performance Analysis

Experimental analysis of MCEGECPO and existing PHCPA [1], as well as GSOCCPP [2] are explained using different parameters namely packet delivery ratio, packet drop rate, throughput, and average latency, and execution time.

**Packet delivery ratio:** It is defined by the number of packets successfully delivered over source to destination pair. It is mathematically formulated as below,

$$R_{PD} = \left[\frac{ND}{N}\right] * 100 \quad (14)$$

Where, $R_{PD}$ designates a packet delivery ratio, '$N$' indicates the number of data packets, $ND$ indicates the number of packets delivered. It is calculated by percentage (%).

**Table 2:** Comparison of packet delivery ratio

| Number of data packets | Packet delivery ratio (%) | | |
|---|---|---|---|
| | MCEGECPO | PHCPA | GSOCCPP |
| 30 | 90 | 86.66 | 83.33 |
| 60 | 91.66 | 85 | 81.66 |
| 90 | 92.22 | 88.88 | 85.55 |
| 120 | 92.5 | 87.5 | 85 |
| 150 | 93.33 | 88 | 84.66 |
| 180 | 92.22 | 87.22 | 84.44 |
| 210 | 91.42 | 88.09 | 85.71 |
| 240 | 92.08 | 87.5 | 85 |
| 270 | 93.70 | 90 | 87.40 |
| 300 | 92.66 | 89.33 | 87 |

Table 2 explains the simulation results of packet delivery ratio with different number of data packets using three methods MCEGECPO and existing PHCPA [1], GSOCCPP [2]. While conducting the simulation 30 data packets are transferred. MCEGECPO technique has 27 data packets are delivered at destination as well as therefore, observed delivery ratio is 90%. Whereas, the delivery ratio of existing methods namely [1] [2] are 86.66% and 83.33% respectively. Therefore, the average overall results indicate packet delivery ratio was significantly improved as 5% and 9% compared with conventional methods.

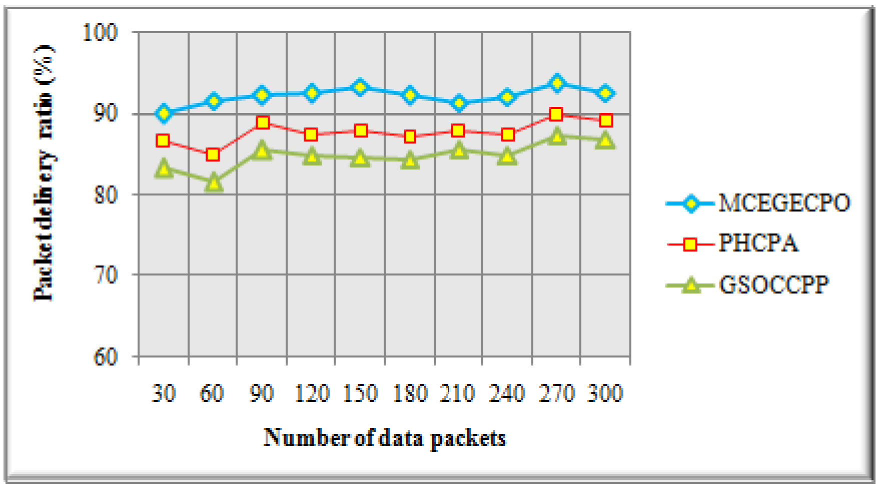

**Fig. 5** Performance analysis of packet delivery ratio

Figure 5 represents comparative performance analysis of packet delivery ratio with numbers of data packets varies of 30-300. From $the\ 'x'$ axis, number of data packets to be transmitted from the source node. The simulation results of packet delivery ratio using three various methods are observed at $'y'$ axis. The graphical result demonstrates the packet delivery ratio was significantly enhanced using the MCEGECPO technique. It is achieved with finding optimal controller placement in an SDN network with multivariate functions. The controller which has a higher data transmission rate is used for increasing the data delivery.

**Packet drop rate:** It is defined by the number of packets dropped over source to destination pair. The Packet drop rate is measured as given below,

$$Packet\ drop\ rate = \left(\frac{number\ of\ packets\ dropped}{N}\right) * 100 \quad (15)$$

**From (12),** $'N'$ indicates the number of data packets. It is calculated by percentage (%).

**Table 3.** Comparison of packet drop rate

| Number of data packets | Packet drop rate (%) | | |
|---|---|---|---|
| | MCEGECPO | PHCPA | GSOCCPP |
| 30 | 10 | 13.33 | 16.66 |
| 60 | 8.33 | 15 | 18.33 |
| 90 | 7.77 | 11.11 | 14.44 |
| 120 | 7.5 | 12.5 | 15 |
| 150 | 6.66 | 12 | 15.33 |
| 180 | 7.77 | 12.77 | 15.55 |
| 210 | 8.57 | 11 | 14.28 |
| 240 | 7.91 | 12.5 | 15 |
| 270 | 6.29 | 10 | 12.59 |
| 300 | 7.33 | 10.66 | 13 |

Table 3 reports the simulation results of packet drop rate during the data transmission with number of data packets. From Table 3, the overall results of packet drop rate are significantly minimized when compared to the existing techniques. The simulation is carried out with 30 data packets in the first iteration. By applying the MCEGECPO technique, a 10% packet drop rate was observed whereas the 13.33% and 16.66% of packet drop rates were observed using PHCPA [1] and GSOCCPP [2] respectively. The overall observed results designate that the comparison oucomes denotes packet drop rate is relatively reduced as 35% and 48% compared with existing methods.

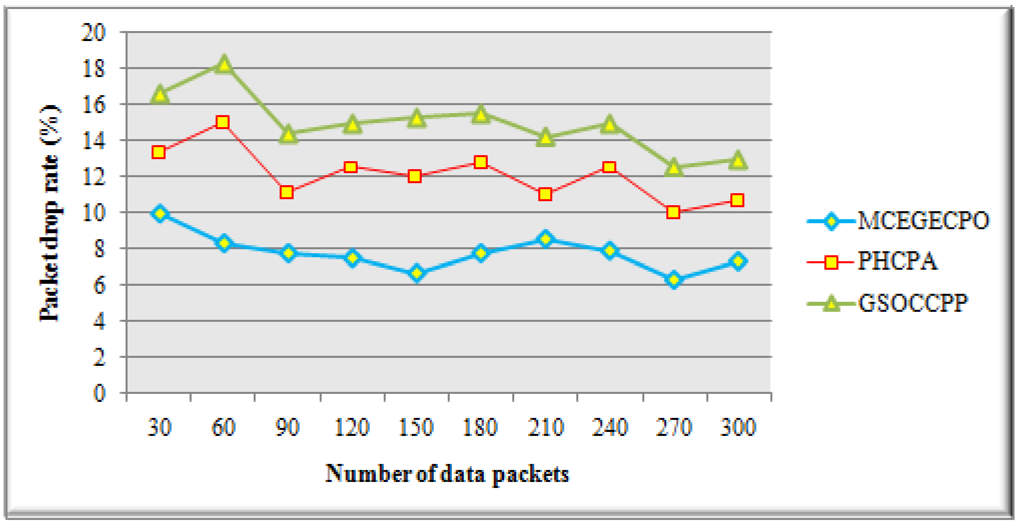

**Fig. 6** Performance analysis of packet drop rate

Figure 6 demonstrates graphical illustration of packet drop rate of three different methods using the MCEGECPO technique, PHCPA [1], and GSOCCPP [2] respectively. As shown in the above graphical illustration, the proposed MCEGECPO achieves a lesser packet drop rate for effective data transmission when compared to the conventional method [1], [2]. This improvement is achieved by selecting the controller with maximum bandwidth, fault tolerance, and load balancing capacity. During the transmission, the selected optimum controller has ability to maintain the operation accurately when the failure occurs. The Load balancing capability improves a data transmission by handling the load across the controllers to the switches. This helps to reduce the packet drop in the data transmission process.

**Throughput**: It is referred by a size of data packets delivered by the destination within specified amount of time. Throughput is calculated by,

$$Throughput = \left(\frac{Amount\ of\ data\ packets\ delivered\ (bits)}{time\ (sec)}\right) \qquad (13)$$

It is evaluated by bits per second (bps).

**Table 4** Comparison of Throughput

| Data packets (size) | Throughput (bps) | | |
|---|---|---|---|
| | MCEGECPO | PHCPA | GSOCCPP |
| 20 | 210 | 180 | 163 |
| 40 | 362 | 280 | 240 |
| 60 | 410 | 365 | 345 |
| 80 | 523 | 465 | 420 |
| 100 | 630 | 580 | 530 |
| 120 | 785 | 690 | 640 |
| 140 | 890 | 780 | 720 |
| 160 | 1020 | 880 | 830 |
| 180 | 1230 | 1100 | 980 |
| 200 | 1420 | 1310 | 1250 |

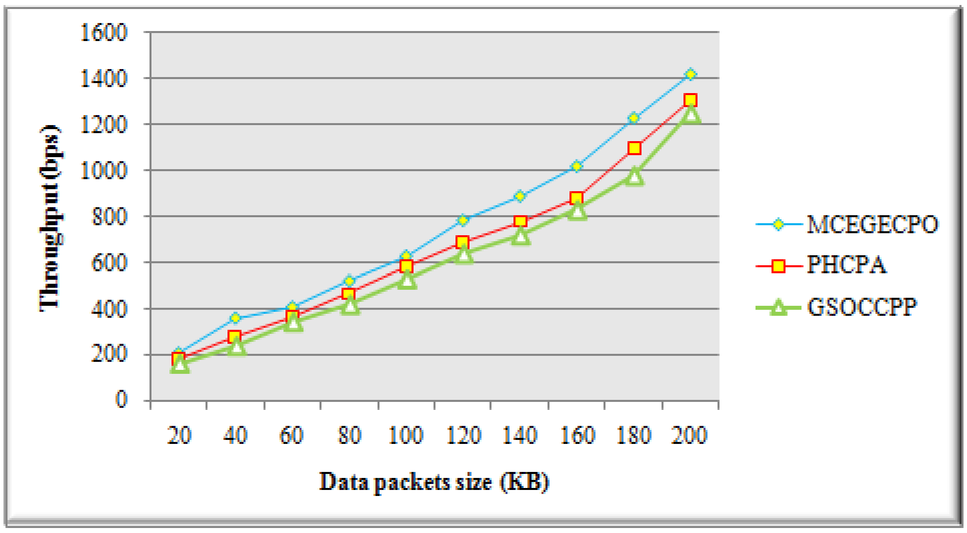

**Fig. 7** Performance analysis of throughput

Table 3 as well as figure7 indicate throughput with size of the packet being transmitted from source to destination. The above graphic representation indicates the horizontal axis denotes data packets and perpendicular demonstrates throughput in bits per second. The observed results indicate that the throughput is superior using MCEGECPO. Let us consider $20\ KB$ of data packets being transfer. MCEGECPO has $210$ bits of the data packets are transmitted within one second. But the throughput of PHCPA [1] and GSOCCPP [2] are achieved by 180 bps and 163bps respectively. This improvement is achieved by using the higher server capacity of the controller to distribute higher sizes of data packets from source to destination hence a greater throughput is achieved.

**Average latency:** It is measured by the amount of time consumed for transmitting data packets.

$$L_{Avg} = N * t\,[TD] \qquad (14)$$

Where, '$L_{Avg}$' indicates average latency, $N$ indicates number of data, $t\,[TD]$ indicates the time taken by algorithm for efficient data transmission. It is calculated by milliseconds (ms).

**Table 5** Comparison of Average Latency

| Number of switches | Average latency (ms) | | |
|---|---|---|---|
| | MCEGECPO | PHCPA | GSOCCPP |
| 2 | 0.15 | 0.19 | 0.3 |
| 4 | 0.24 | 0.3 | 0.4 |
| 6 | 0.4 | 0.5 | 0.6 |
| 8 | 0.55 | 0.6 | 0.7 |
| 10 | 0.7 | 0.8 | 0.89 |
| 12 | 0.89 | 1.1 | 1.2 |
| 14 | 1.3 | 1.5 | 1.6 |
| 16 | 1.8 | 1.9 | 2 |

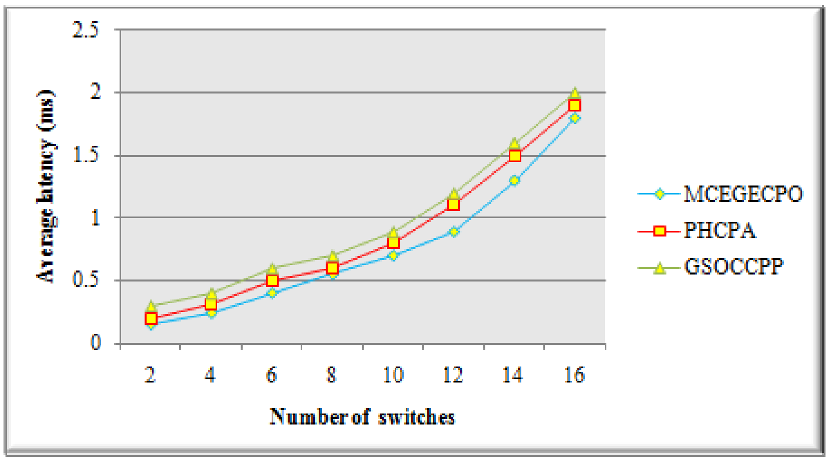

**Fig. 8** Performance analysis of average latency

The simulation results of average latency with respect to a number of switches are reported in table 4 and figure 8. The latency of three different methods is illustrated in the above figure with different lines of the curve. From the observed results, average latency during the communication is reduced with the MCEGECPO technique than the two existing methods namely PHCPA [1], and GSOCCPP [2] respectively. The significant reason for this improvement is to select the optimal controller with better load balancing capacity and higher bandwidth using Combinatorial Elitism Golden Eagle Optimization. Moreover, the Optimization finds the best optimal controller with lesser propagation latency to transmit the data in an SDN topology. This increases the data delivery with minimum latency.

**Execution time:** It is measured by number of time for finding the optimal controller for efficient data transmission.

$$ET = n * t\,[FOC] \qquad (14)$$

Where '$ET$' indicates an Execution time, $n$ indicates the number of switches (i.e. nodes) in the network, and $t\,[FOC]$denotes the time taken by the algorithm for finding the optimal controller. It is computed by milliseconds (ms).

**Table 6** Comparison of Execution Time

| Number of switches | Execution time (ms) | | |
|---|---|---|---|
| | MCEGECPO | PHCPA | GSOCCPP |
| 2 | 6 | 8 | 10 |
| 4 | 8 | 12 | 16 |
| 6 | 10.8 | 14.4 | 15.6 |
| 8 | 12 | 16 | 18.4 |
| 10 | 14 | 18 | 20 |
| 12 | 15.6 | 20.4 | 22.8 |
| 14 | 17.5 | 22.4 | 25.2 |
| 16 | 19.2 | 24 | 27.2 |

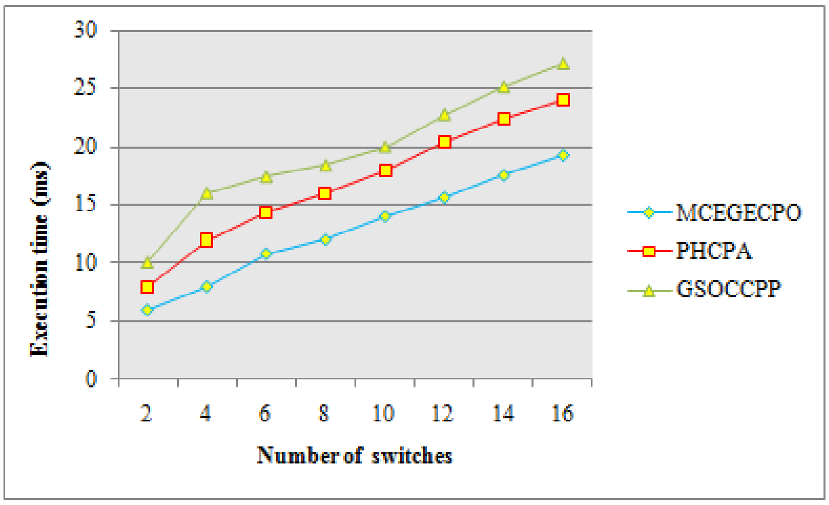

**Fig. 9** Performance analysis of execution time

Table 6 and figure 9 given above illustrate the performance results of the execution time of three methods namely the MCEGECPO technique, PHCPA [1], and GSOCCPP [2] with number of switches. Different outcomes are attained with number of input data to every technique. It denotes the performance of execution time using MCEGECPO is found to be minimized by 24% and 36% compared with existing methods. This is because of Combinatorial Elitism Golden Eagle Optimization. The optimization uses the Combinatorial as well as Elitism strategy to minimize the time complexity of the algorithm. In addition, the graphical arrangement of switches in the network also minimizes the finding of optimal controller placement.

## 6. Conclusion

In this paper, a new mathematical optimization algorithm called MCEGECPO is developed by deploying numerous controllers in SDN networks. MCEGECPO is developed for addressing multi-objective CPP. The MCEGECPO algorithm is low-time complexity and used to address CPP. By applying a Combinatorial Elitism Golden Eagle Optimization, a controller placement is carried out based on Multivariate functions. Fitness is estimated with the Multivariate functions to select the best optimal based on position updates. Hence, the overall data transmission performance gets improved with minimum latency. Simulations were conducted in the GBN with various parameters. The overall assessment outcomes show effectiveness of our proposed MCEGECPO technique for performing better packet delivery ratio with minimum latency, execution time, and packet loss.

## References

1. Nasrin,F., Mohammad, M., Amin, B. S., Kambiz M., (2021). A novel controller placement algorithm based on network portioning concept and a hybrid discrete optimization algorithm for multi-controller software-defined networks. Cluster Computing, 24, 2511–2544. https://doi.org/10.1007/s10586-021-03264-w

2. Sahand, T., Mohsen, J., (2020). A new GSO based method for SDN controller placement. Computer Communications, 163, 91–108. https://doi.org/10.1016/j.comcom.2020.09.004
3. Ali, A.S., Seyed, A.H.S., (2021). Multi-criteria decision-making for controller placement in software-defined wide-area networks. The Journal of Supercomputing, 77, 13447–13473. https://doi.org/10.1007/s11227-021-03815-3
4. Afsane, Z., Chung-Horng, L., Thomas, K., (2022). Controller Placement in Software-Defined Multihop Wireless Networks: Optimal Solution and GA-based Approximation. Mobile Networks and Applications, 1-16. https://doi.org/10.1007/s11036-021-01894-3
5. Lingxia, L., Victor, C. M. L., Zhi, L., Han-Chieh, C., (2021). Genetic Algorithms with Variant Particle Swarm Optimization Based Mutation for Generic Controller Placement in Software-Defined Networks. Symmetry, 13(7), 1-24. https://doi.org/10.3390/sym13071133.
6. Vidya, S.T., Koppala, G., (2022). FCMCP: Fuzzy C-Means for Controller Placement in Software Defined Networking. Procedia Computer Science, 201, 109-116. https://doi.org/10.1016/j.procs.2022.03.017
7. Dorabella, S., Teresa, G.,(2021). Joint optimization of primary and backup controller placement and availability link upgrade in SDN networks, Optical Switching and Networking, 42, 1-12. https://doi.org/10.1016/j.osn.2021.100634.
8. Tao, H., Quan, R., Peng, Y., Ziyong, L., Julong, L., Yuxiang H., Qian, L., (2021). An efficient approach to robust controller placement for link failures in Software-Defined Networks, Future Generation Computer Systems, 124, 187-205. https://doi.org/10.1016/j.future.2021.05.022
9. Abeer, A. Z. I., Fazirulhisyam, H., Nor, K. N., Aduwati, S., Keivan, N., Saber,M. E. F., (2020). Heuristic Resource Allocation Algorithm for Controller Placement in Multi-Control 5G Based on SDN/NFV Architecture. IEEE Access, 9, 2602 – 2617. DOI: 10.1109/ACCESS.2020.3047210.
10. Ashutosh, K., Singh, S. M., Naveen, K., Shashank, S., (2020). Heuristic approaches for the reliable SDN controller placement problem. Transactions on Emerging Telecommunications Technologies, Wiley, 31(2), 1-18. https://doi.org/10.1002/ett.3761
11. Mili, D., Bidyut, K. B., Mrinal, K. D., Swapan, D., (2021). A new optimization technique to solve the latency aware controller placement problem in software defined network's. Transactions on Emerging Telecommunications Technologies, 32(10), 1-17. https://doi.org/10.1002/ett.4316.
12. Nivine, S., Muhammed, S., (2021). A fault tolerance metaheuristic-based scheme for controller placement problem in wireless software-defined networks. International Journal of Communication Systems, 34(4), 1-22. https://doi.org/10.1002/dac.4624.
13. Yi, L., Shaopeng, G., Conghui, Z., Wenwen, S., (2020). Parameter Optimization Model of Heuristic Algorithms for Controller Placement Problem in Large-Scale SDN. IEEE Access, 8, 151668 – 151680. DOI: 10.1109/ACCESS.2020.3017673
14. Afsane, Z., Chung-Horng, L., (2020). Using Heuristics to the Controller Placement Problem in Software-Defined Multihop Wireless Networking. Communications and Network, 12, 199-219. DOI: 10.4236/cn.2020.124010.

15. Ali, A.S., Seyed, A.H.S., Ahmad, M., Rahmat, B., (2021).Cost-Effective Survivable Controller Placement in Software-Defined Networks. IEEE Access, 9, 29130 – 129140. DOI: 10.1109/ACCESS.2021.3113496
16. Dorabella, S., Teresa, G., David, T., (2021).SDN Controller Placement With Availability Upgrade Under Delay and Geodiversity Constraints. IEEE Transactions on Network and Service Management, 18(1), 301 – 314. DOI: 10.1109/TNSM.2020.3049013.
17. Jehad, A., Byeong-hee, R., (2022). An Effective Approach for Controller Placement in Software-Defined Internet-of-Things (SD-IoT). Sensors, 22, 1-16. https://doi.org/10.3390/s22082992
18. Ali, A. S., Seyed, A.H.S., Ahmad, M., (2021). Dynamic controller placement in software-defined networks for reducing costs and improving survivability. Transactions on Emerging Telecommunications Technologies, 32(1), 1 -17. https://doi.org/10.1002/ett.4152.
19. Ahmad, J., Manijeh, K., Reza, A., (2020). A new framework for reliable control placement in software-defined networks based on multi-criteria clustering approach. Soft Computing, 24, 2897–2916. https://doi.org/10.1007/s00500-019-04070-8.
20. Ali, .S., Seyed,A.H.S., Rahmat, B., (2021). Kernel Search-Framework for Dynamic Controller Placement in Software-Defined Network. Computers, Materials & Continua, 68, (3), 3391–3410. doi:10.32604/cmc.2021.017313
21. Hemagowri, J., TamilSelvan, P., (2022). Demming Regressive Multiobjective Dragonfly Optimized Controller Placement in SDN Environment. International Journal of Nonlinear Analysis and Applications, 13(1), 1747-1761. doi: 10.22075/IJNAA.2022.5789.