Int. J. Nonlinear Anal. Appl. 13 (2022) 1, 1747-1761
ISSN: 2008-6822 (electronic)
http://dx.doi.org/10.22075/ijnaa.2022.5789

Demming regressive multiobjective dragonfly optimized controller placement in SDN environment

J. Hemagowri[a,*], P. TamilSelvan[a]

[a] *Department of Computer Science, Karpagam Academy of Higher Education, Coimbatore, India*

(Communicated by Madjid Eshaghi Gordji)

Abstract

Wireless networks are of high significance in current telecommunication systems, and in order to improve these systems, Software-Defined Networks (SDN)is used to centrally monitor and control the whole network with the help of a controller. The design of an SDN-based network is needed to identify the optimal amount of controllers for improving the performance of the network. The controller placement problem is determined on propagation latency but it failed to consider the load balancing, fault tolerance, bandwidth consumption and data transmission rate. Plan to develop a novel technique called Demming Regressive Multiobjective Dragonfly Optimized Controller Placement (DRMDOCP) for an optimal number of controller placement for enhancing network performance during different topologies. By applying DRMDOCP, the optimum number of controllers are selected and placed into the network to improve the overall network's performance. Therefore, a delay minimization-based controller placement strategy is extremely preferred. The simulation of the DRMDOCP technique and existing methods is conducted using a network topology dataset and different performance metrics. The simulation results demonstrate that the DRMDOCP increases the packet delivery, throughput and reduces the average latency, packet drop when compared to the state-of-the-art method.

Keywords: Control placement Problem, Dragonfly optimization, SDN.

1. Introduction

Wireless Networks is a highly dynamic nature, SDN is a promising network technology that distributes the programmable networks by partitioning the control and data plane from the conventional architecture.The controlplane is then reasonably centralized in an external entity called a controller

*Corresponding author

Email addresses: hemagowri.j@gmail.com (J. Hemagowri), tmselvanin@gmail.com (P. TamilSelvan)

Received: March 2021 *Accepted:* May 2021

used to centrally monitor and control the entire network. Thus, several controllers are employed for handling the software-defined wireless network to achieve the enhanced performance. Deploying many controllers depend on dissimilar assignment parameters namely capacity of controllers, load on switches, and latencies, and so on. Therefore, optimum amount of controllers effectively improves network performance is known as controller placement problem.

In order to obtain minimal time consumption, Garter Snake Optimization Capacitated Controller Placement Problem (GSOCCPP) was developed in [19]. But it failed to consider the different factors such as load balancing, fault tolerance, bandwidth for minimizing the average latency. A varna-based optimization (VBO) algorithm was introduced in [17] to solve Controller Placement Problem. Though the designed VBO algorithm reduces the average latency, the throughput was not achieved. A Parameter Optimization Model (POM) was introduced in [11] to resolve the controller placement problem. The designed model reduced the time consumption but the parameter optimization problem algorithm was not solved to obtain a reasonable SDN controller placement scheme. A Placement Availability Resilient Controller (PARC) system was developed in [4] for reducing Latency. The designed scheme was failed to use the efficient optimization scheme for solving the controller's placement.

In order to solve the multi-objective controller placements and minimizing the switch-to-controller delay, an efficient metaheuristic-based Reliability-Aware and Latency-Oriented controller placement method (RALO) was designed in [6]. The designed method was not efficient to solve the fault-tolerant controller placement. A controller placement approach was designed in [14] for solving the multiple link failures, to minimize the latency. The designed approach was failed to solve the multi-objective problems.

In order to improve the device-to-device communication by placing the controllers, Ant colony system with external memory (ACS-EM) algorithm was introduced in [12]. However, algorithm failed to solve the controller placement has additional parameter namely reliability and distance among controllers. A multi-criteria decision-making (MCDM) approach was introduced in [1] for selecting an optimal SDN controller to enhance the QoS metric such as delay and throughput. But the other metrics such as delivery ratio and the drop rate were not estimated.

A simulated annealing-based heuristic technique was introduced in [3] with the aim to minimize the latencies from all switches to the particular backup controllers. The designed technique failed to consider the inter-controller latencies and Load balancing aspects. A multi-period offline optimization technique was developed in [7] to minimize the total cost and reduce the complexity. The designed technique used more controllers to increase the flexibility of the dynamic control plane. But the average latency was not accurately reduced.

In order to resolve the issues of controller placement in SDN, a novel machine learning-based optimization technique called DRMDOCP is introduced. The novel contribution of the DRMDOCP technique is summarized as given below,

- A novel DRMDOCP is introduced for solving the multi-objective controller placement problem by integrating the Demming regression with Dragonfly optimization.

- In DRMDOCP, dragonfly metaheuristic optimization is applied for finding the optimum controller with the help of a multi-objective function that includes a propagation latency, load balancing capability, bandwidth, fault tolerance and data transmission rate.

- The Demming regression is applied to a Dragonfly optimization to analyze the multiple objective functions and finds the best-fit controller among the populations. The final best solution helps to minimize the average latency of data transmission, drop rate and improve the throughput.

- Finally, the simulation is conducted with the network topology for evaluating proposed DRMDOCP technique with existing methods based on different metrics.

The article is organized as follows. Section 2 describes related works on Controller Placement Problem. Section 3 defines network structure, and formulating proposed DRMDOCP algorithm. Section 4 explains simulation settings with network topology. Section 5 illustrates results of simulations for three different methods. Section 6 explains discussion of algorithm evaluations.

2. Related Works

A Varna-based optimization (VBO) method was presented in [16] for reliable Controller Placement has reduced overall latency of SDN.In order to consider other constraints of SDN optimization approach was unsuccessful. A Reliable Controller Placement Problem Model (RCPPM) was developed in [9] to enhance the reliability of SDN. The designed model reduces the average execution time but the multi-objective optimization was not considered.
Controller Placement Optimization was presented in [13] for software-defined wide-area networks to reduce the latency. However, the designed optimization technique failed to consider the different link optimization method was introduced bandwidths. A Pareto Integrated Tabu Search (PITS) algorithm was introduced in [15] to detect the optimal position of controllers to improve the performance of the network. However, the machine learning-based approach was not applied for predicting the optimum controllers for the placement methodology.
Evolutionary Multi-Objective Placement of SDN Controllers was developed in [18] for solving the fault tolerance. Though the average execution time of the controller's placement was minimized, the average latency was not minimized. A simulated annealing-based algorithm was designed in [20] to solve the objective function for improving the throughput and spatial link failure probability. However, the designed algorithm failed to minimize the packet drop.
An efficient, heuristic multi-objective optimization method was introduced in [8] for controller placement. The designed method failed to consider the load balancing and optimal resource management cost. A simulated annealing genetic algorithm was designed in [5] for reducing latency with adequate wireless controllers. However, the designed algorithm failed to improve the throughput.
Controller Placement Genetic Algorithm (CPGA) was designed in [10] for reducing latency-based assignment in link load balancing. However, the higher throughput was not achieved. Generic Controller Adaptive Load Balancing (GCALB) algorithm was introduced in [2] to increase the throughput and response time metrics. But the designed algorithm was not solving the multiple objective problems for achieving higher throughput.

3. Methodology

SDN assists a centralized networking system where a controller handles the global view of the network. With help of softwarization, the capability for examining and controlling the network is called as controller. In order to manage huge network traffic, single controllers are deployed. Therefore, wide area networks have current SDN for making various controllers. It is important research trouble with minimum number of controllers. Based on the motivation, a novel technique called DRMDOCP is introduced. The system model of the proposed DRMDOCP is discussed in the below subsections.

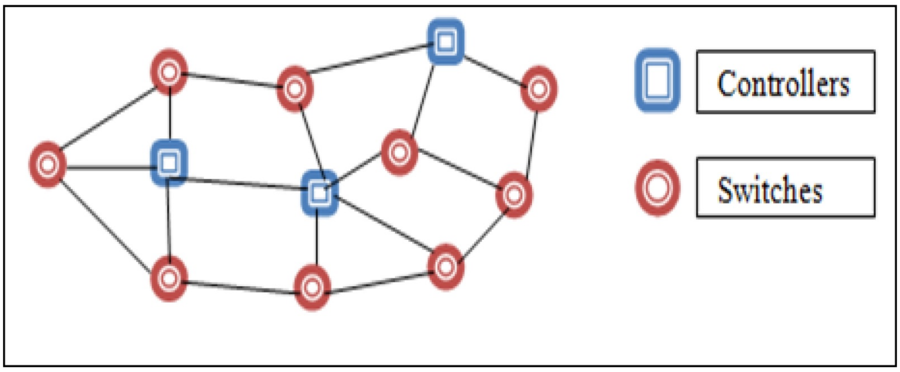

Figure 1: graphical model of the controller placement problem

3.1. System model

The system model of proposed DRMDOCP technique is discussed. Here, the controller placement problem is organized into the undirected graphical model $G(s, e)$ where sdenotes set of switches and indicates set of links among switches. The number of controllers' $C = \varphi_1, \varphi_2, ..., \varphi_n$ is positioned in an optimal way in the SDN-based infrastructure.

Figure 1 illustrates graphical model of controller placement problem where the red-colored circle denotes switches and the blue-colored denotes controllers. The SDN consists of a central controller with overall network visibility that communicates to\any switch directly for improving the speed of data transmission. According to network topology number of controllers are deployed in network is analyzed based on multiple objective functions. In DRMDOCP technique, the shortest distance among switch and controller placement is identified with binary variable $[0, 1]$ depend on multiobjective functions optimum solution has various factors namely propagation latency (α_{lat}), load balancing capability (α_{load}), bandwidth (α_{bw}), fault tolerance (α_{ft}), and data transmission rate (α_{DTR}). Binary variables have been applied for finding shortest distance among switch and controller is mathematically expressed as a given below,

$$D(s_i, C_j) = \begin{cases} 1, & \text{if } (\min \alpha_{lat}) \text{ and } (\max \alpha_{load}, \alpha_{bw}, \alpha_{ft}, \alpha_{DTR}); \\ 0, & \text{otherwise.} \end{cases} \tag{3.1}$$

Where, $D(s_i, C_j)$ represents the distance function between the switches "s_i" and controllers "C_j" If the binary function returns "1", then the switches are linked to the optimal controllers. Otherwise, the binary function returns "0". In this way, the controller placement is solved by using the following mathematical optimization.

3.2. Mathematical optimization model for controller placement

In this section, the Mathematical optimization model is designed by using called Demming Regressive Multiobjective Dragonfly optimization. The proposed dragonfly optimization is the metaheuristic technique that helps to find an accurately better solution to the optimization problems. The Multiobjective represents the proposed dragonfly optimization algorithm that solves the multiple objective problems in the controller placement such as propagation latency, load balancing capability, bandwidth, fault tolerance, and data transmission rate.

The behavior of the dragonfly is a movement which is seeking for its food source. Here the dragonfly is related to the number of controllers "$C = \varphi_1, \varphi_2, ..., \varphi_n$" and the food source is related to multiobjective functions i.e., propagation latency, load balancing capability, bandwidth, fault tolerance, and data transmission rate. A proposed Demming Regressive Multiobjective Dragonfly optimization works on the basis of a population-based approach called a swarm. The proposed optimization starts to initialize the population of the "n" number of dragonflies (i.e. controllers "$C = \varphi_1, \varphi_2, ..., \varphi_n$") in the search space (i.e. SDN).

$$C = \varphi_1, \varphi_2, ..., \varphi_n \tag{3.2}$$

After the initialization process, the fitness is computed for each dragonfly in the current swarm population. The fitness is calculated based on multiobjective functions. Initially, propagation latency among switches and controllers are estimated.

In networking, propagation latency is the amount of time taken of the data to transmit from the controller to the switches. It is mathematically estimated as given below,

$$\alpha_{lat} = time\ [TD] \tag{3.3}$$

Where, α_{lat} denotes a latency, TD denotes a data transfer from the controller to the switches.

Load balancing capability (α_{load}) between the switch and controllers is measured based on the calculation of load factor. The load factor is measured as the fraction of the average load to the maximum demand during a particular time.

$$L_f = \left[\frac{Avg_L}{\max_D}\right] \tag{3.4}$$

Where, L_f denotes a load factor, Avg_L denotes an average load, $\max_D$ denotes a maximum demand. A load factor less than "1" is said to be a controller that has better load balancing capability.

Bandwidth is defined as the maximum rate of data transfer capacity in a specific amount of time from the controller to the switches.

$$\alpha_{bw} = \left[\frac{\max_D(bits)}{S}\right] \tag{3.5}$$

Where, α_{bw} denotes a Bandwidth, $\max_D(bits)$ denotes a maximum rate of data transfer in terms of bits, S denotes a second. Therefore, the bandwidth is measured in terms of Mega or Gigabits per second (Mbps or Gbps).

Fault tolerance in the SDN is the capability of a controller to continue operating properly when the failure occurs. The failure rate of controller is estimated based on amount of failures that occur to the entire operating time. It is mathematically formulated as given below,

$$R_f = \frac{Number\ of\ failures}{Total\ operating\ time} \tag{3.6}$$

Where, R_f denotes a failure rate. If the failure rate is minimum and then the controller has better fault tolerance capability. The data transmission rate is defined as the amount of data transmitted over a channel within a particular unit of time. The units used for this are $bits/s$.

Based on the above-said parameters, the fitness is estimated based on the regression function. Deming regression is a machine learning technique that helps to analyze the given input (i.e., controllers) and finds the best fit from the populations by satisfying the multi-objective functions. The regression estimation is expressed as given below,

$$Y_i = \vartheta_0 + \vartheta_1[MO(C)], where\ MO(C) \in \alpha_{lat}, \alpha_{load}, \alpha_{bw}, \alpha_{ft}, \alpha_{DTR} \tag{3.7}$$

Where, Y_i denotes an output of regression function, ϑ_0 and ϑ_1 indicates the regression coefficients, $MO(C)$ denotes multiobjective estimation. Based on the regression analysis, the controller has minimum latency and higher load balancing capacity, bandwidth, fault tolerance and data transmission rate are chosen as the optimal for controller placement. Based on the analysis, the fitness is computed as given below,

$$F = \arg\min \alpha_{lat} \&\& \arg\max \alpha_{load}, \alpha_{bw}, \alpha_{ft}, \alpha_{DTR} \tag{3.8}$$

Where, F indicates a fitness function, $\arg\min$ denotes an argument of a minimum function, $\arg\max$ denotes an argument of maximum function.
Based on the fitness measure, there are four swarming behaviors of dragonflies are estimated in the search space. The four behaviors are separation, alignment, cohesion, and attraction towards the food source that helps to find the global optimal solution among the population.
Initially, the separation process is carried out to find out the current position of dragonflies and the position of neighboring dragonflies.

$$\beta_1 = -\sum_{j=1}^{n}(x_{it} - x_{jt}) \tag{3.9}$$

Where, β_1 indicates a separation of the dragonflies, x_{it} denotes a current position of a dragonfly, x_{jt} symbolizes a position of the neighboring dragonflies, "n" is a number of neighboring dragonflies in search space.
The second process is an alignment that specifies the movement velocity of dragonflies towards that of the neighboring dragonflies.

$$\beta_2 = \frac{1}{n}\sum_{j=1}^{n}\tau_j(t) \tag{3.10}$$

Where, β_2 indicates an alignment, $\tau_j(t)$ stand for a velocity of neighboring dragonflies, "n" denotes neighboring dragonflies.
Thirdly, the cohesion process is carried out to find the tendency of dragonflies towards the center of the mass of their neighborhood.

$$\beta_3 = \frac{1}{n}\sum_{j=1}^{n}[x_{jt} - x_{it}] \tag{3.11}$$

From (3.11), β_3 represents a cohesion process of the dragonfly, x_{jt} indicates a position of the neighboring dragonfly, x_{it} be a position of a current dragonfly, n is the number of neighborhoods.
Finally, the attraction process towards the food source is estimated based on the current position of the food source and the position of the dragonfly.

$$\beta_4 = \mid x_f - x_{it} \mid \tag{3.12}$$

From (3.12), β_4 represents an attraction towards a food source, x_f indicates a position of the food source, x_{it} be a current position of the dragonfly.
The position of current dragonfly gets updated along with their neighborhoods,

$$\mathcal{X}_{i(t+1)} = \mathcal{X}_{i(t)} + \nabla\mathcal{X}_{i(t+1)} \tag{3.13}$$

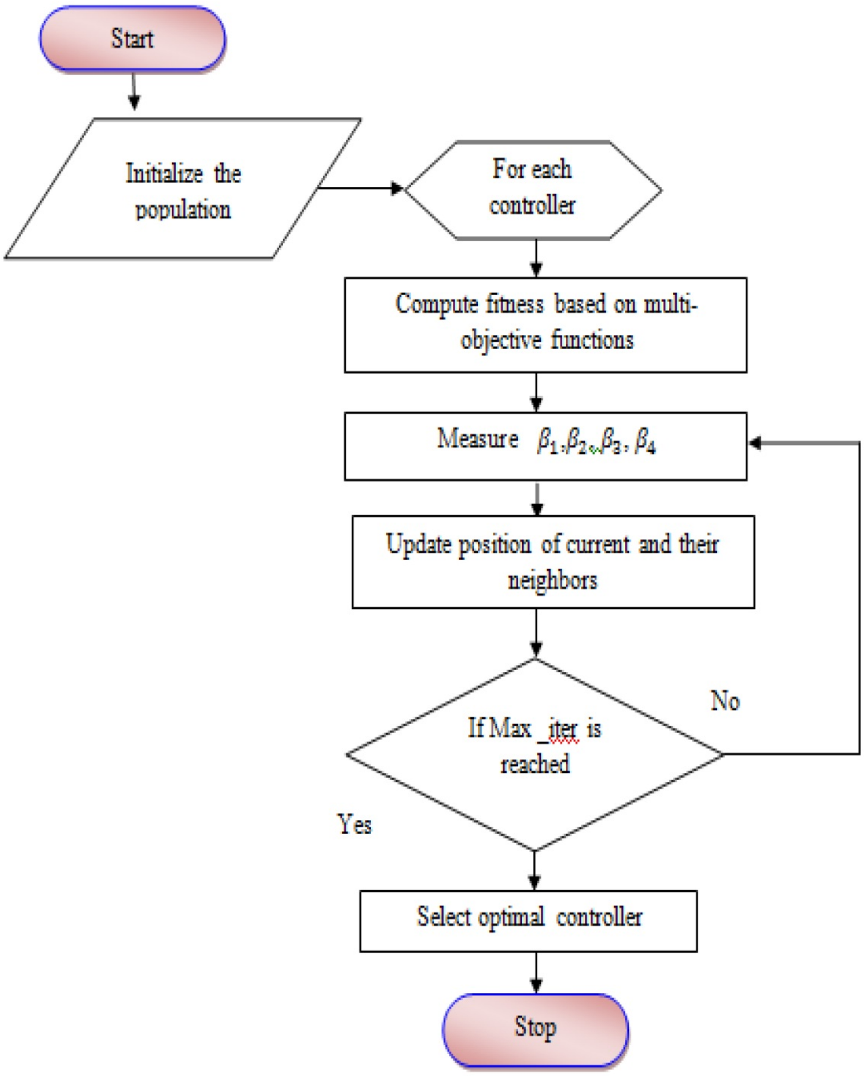

Figure 2: Flow diagram of Demming Regressive Multiobjective Dragonfly optimization

From (3.13), $\mathcal{X}_{i(t+1)}$ symbolizes the updated position of the dragonfly, $\mathcal{X}_{i(t)}$ is the current position of a dragonfly, $\nabla\mathcal{X}_{i(t+1)}$ indicates a step vector that is used to find the movement direction of the dragonfly.

$$\nabla\mathcal{X}_{i(t+1)} = \omega_1\beta_1, \omega_2\beta_2, \omega_3\beta_3, \rho_f\beta_4 + \theta * x_{(t)} \tag{3.14}$$

From (3.14), ω_1 indicates a weight of separation function (β_1), ω_2 is the weight of alignment function β_2, ω_3 is the weight of cohesion β_3, ρ_f stand for a food vector, β_4 is the attraction towards a food source, θ indicates an inertia weight that helps to controls the convergence behavior of optimization,$x_{(t)}$ indicates a position of the dragonfly at time "t". Based on the updated results, the global best solution is identified.

Figure 2 reveals the flow process of Demming Regressive Multiobjective Dragonfly optimization for finding the optimal controller. After finding the optimum controller, equation (3.1) is applied for performing the controller placement in SDN. The algorithmic process of the proposed DRMDOCP technique is described as follow,

Algorithm 3.2 describes the step-by-step process of Demming Regressive Multiobjective Dragonfly

// Algorithm 1 Demming Regressive Multiobjective Dragonfly optimization

Input: dataset, Number of controllers $C = \varphi_1, \varphi_2, \ldots \varphi_n$,
Output: Find optimized controllers for placement

Begin
Initialize the population of controllers $C = \varphi_1, \varphi_2, \ldots \varphi_n$
For each controller φ_i
  Compute multiple objective functions $\alpha_{lat}, \alpha_{load}, \alpha_{bw}, \alpha_{ft}, \alpha_{DTR}$
  Apply Demming Regression **to** find the best fit
  Measure the fitness 'F'
  Calculate $\beta_1, \beta_2, \beta_3, \beta_4$
if $\left(F(\varphi_i) < F(\varphi_j)\right)$ **then**
  Update the position of dragonfly '$x_{i(t+1)}$'
End if
If (Max _iter is reached) **then**
  Obtain global best solution
end if
return (optimum placement)
End

optimization for controller placement. Initially, the numbers of the controller's populations are generated in search space. For each controller, the multi-objective function is measured. Then the regression is applied for analyzing the estimated multi-objective functions. Based on the regression analysis, the fitness is estimated. After that, there are four various principles of the swarm behaviors of the dragonfly optimization that are calculated for finding the global best based on position updates. This process gets iterated until it reaches the maximum iterations. Finally, the optimum placement is said to be obtained. Based on the analyzed results, the optimum number of controllers are selected and placed into the network to improve the overall network performance. Therefore, it is highly preferred to design an efficient network for improving the network throughput and delivery ratio and minimizing the latency.

4. Simulation Setting

Extensive simulation of the DRMDOCP and existing GSOCCPP [19], VBO [17] is carried out in an NS-2 simulator using a GBN network topology taken from Networking Training Datasets. The simulation results are evaluated using different parameters namely packet delivery ratio, packet loss rate, throughput and average latency.

5. Performance Results and Discussion

The simulation results of the DRMDOCP and existing GSOCCPP [19], VBO [17] are discussed with respect to various performance metrics such as average latency, throughput, and packet drop rate and packet delivery ratio. Through the simulation analysis, it is verified that the proposed DRMDOCP technique in this paper is better than the previous methods.

- **Packet delivery ratio:** It is referred as amount of packets effectively delivered from source to the destination pair in network. The formula for calculating the Packet delivery ratio is formulated as given below,

$$R_{PD} = \left(\frac{ND}{N}\right) * 100 \tag{5.1}$$

Table 1: simulation parameters and values

Simulation Parameters	Values
Network Simulator	NS2.34
Number of nodes (switches) in GBN network topology	17
Number of data packets	25,50,75,100,125,150,175,200,225,250
Number of controllers	5
Simulation time	300sec
Protocol	DSR
Number of runs	10

Where, R_{PD} denotes packet delivery ratio, N denotes number of data packets, ND denotes number of packets successfully delivered. It is measured in terms of percentage (%).

- **Packet drop rate:** It is referred as amount of packets dropped from source to destination pair in network. It is measured as given below,

$$Packet\ drop\ rate = \left(\frac{number\ of\ packets\ dropped}{N}\right) * 100 \tag{5.2}$$

Where N denotes the number of data packets. It is measured in terms of percentage (%).

- **Throughput:** It is referred as amount of data packets (i.e. size) successfully delivered from source to the destination pair in a specific time. Throughput is calculated as given below,

$$Throughput = \left(\frac{Amount\ of\ data\ packets\ delivered\ (bits)}{time\ (sec)}\right) * 100 \tag{5.3}$$

The throughput is estimated in terms of bits per second (bps).

- **Average latency:** It is defined as time interval to transfer the data packets from controller to switches.

$$L_{Avg} = N * t[TD] \tag{5.4}$$

Where, "L''_{Avg} denotes an average latency, N denotes the number of data, and $t[TD]$ denotes a time consumed for performing the data transmission. It is measured in terms of milliseconds (ms).

Table 2 provides the analysis of the comparative results of three heuristic-based optimization solutions namely DRMDOCP and existing GSOCCPP [14], VBO [5] for controller placement. The observed comparative results indicate that the performance of DRMDOCP is better in terms of achieving a higher packet delivery ratio. In simulation setting, every algorithm is achieved by 10 runs, and numbers of switches are same for all the controllers. In the first run, 25 data packets are considered. By constructing the graphical model, the delivery ratio between the controllers and switches is estimated. With the application of DRMDOCP, 21 data packets are successfully delivered from source to destination pair of nodes. Therefore, the delivery ratio of DRMDOCP is 84% and the

Table 2: comparative results of Packet delivery ratio

Number of data packets	Packet delivery ratio (%)		
	DRMDOCP	GSOCCPP	VBO
25	84	80	76
50	88	84	80
75	90.66	86.66	84
100	88	85	83
125	90.4	88	84
150	88	85.33	82
175	91.42	88.57	85.71
200	89	86	83.5
225	91.11	88.88	84.44
250	90	86	82

Table 3: comparative results of Packet drop rate

Number of data packets	Packet drop rate (%)		
	DRMDOCP	**GSOCCPP**	**VBO**
25	16	20	24
50	12	16	20
75	9.33	13.33	16
100	12	15	17
125	9.6	12	16
150	12	14.66	18
175	8.57	11.42	14.28
200	11	14	16.5
225	8.88	11.11	15.55
250	10	14	18

delivery ratio of existing GSOCCPP [19], VBO [17], 80% and 76%. Likewise, the nine various runs are carried out with the various counts of the input. The obtained results of DRMDOCP technique are compared with existing methods. The average of ten comparison results is taken into consideration of final results. The average results prove that the DRMDOCP increases the performance of packet delivery ratio by 4% and 8% than the conventional methods [17, 19].

Figure 3 depicts the convergence plots of packet delivery ratio versus amount of data packets ranges from 25 to 250. As shown in the plot, the number of data packets is considered as input and the packet delivery ratio is observed as output. The output of the three optimization techniques DRMDOCP and existing GSOCCPP [19], VBO [17] are represented by three different colors namely orange, red and green. From the observed results, the packet delivery ratio is found to be improved by using the DRMDOCP than the other methods. The significant reason is to apply the Regressive Multiobjective Dragonfly Optimization technique. The proposed optimization technique finds the optimal placement of controllers among the switches in the network. The optimum controller placement increases the data delivery between the source and destination pair.

The table 3 values indicate that the packet drop rate versus a number of data packets using three different optimization techniques namely DRMDOCP, GSOCCPP [19], VBO [17]. For the simulation intentions, the number of data packets is taken as input in the ranges from 25 to 250. The observed result validates that the packet drop rate using the DRMDOCP technique is decreased when compared to the existing optimization techniques. However, with the '25' number of data

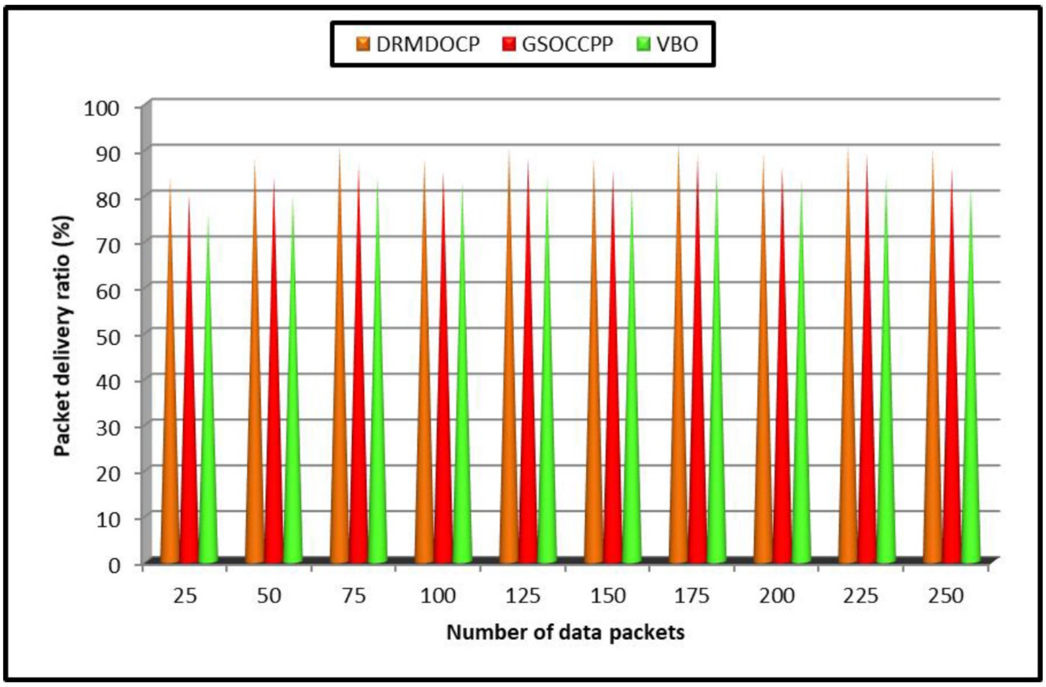

Figure 3: Convergence plots of packet delivery ratio

packets considered for simulation, the packet delivery ratio being '16%' and the observed packet drop rate of existing [17, 19] are '20%', and 24%. The average of ten various results indicates that the overall average rate of packet drop rate of DRMDOCP is considerably reduced by 23% when compared to GSOCCPP [19], and 38% when compared to existing VBO [17].

Figure 4 displays convergence plots of packet drop rate of the three methods. As shown in the plot, the orange color denotes packet drop rate of proposed DRMDOCP and red and green color cone denotes packet drop rate of existing GSOCCPP [19] and VBO [17] respectively. The graphical plot confirms that the DRMDOCP reduces the packet drop rate than the other two existing methods. The major reason is to perform the optimum placement of the controllers in the network. Controller's placement is performed depending on higher bandwidth and balancing load capability. This helps to minimize the packet drop between the sources and the destination pair of nodes.

Table 4 above shows the performance of the throughput for all three methods. Taking the as the

Table 4: comparative results of throughput

Size of data packets (KB)	Throughput (bps)		
	DRMDOCP	GSOCCPP	VBO
15	175	160	147
30	226	205	193
45	344	310	280
60	420	390	350
75	560	480	430
90	620	560	520
105	710	670	630
120	820	770	730
135	962	880	830
150	1058	960	880

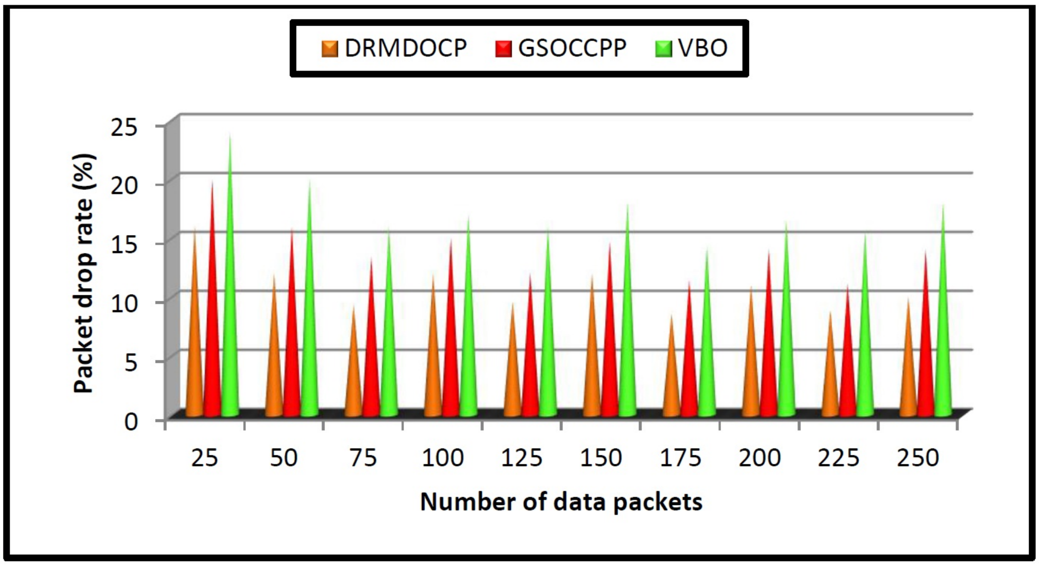

Figure 4: Convergence plots of packet drop rate

Table 5: comparative results of average latency

Number of switches	Average latency (ms)		
	DRMDOCP	GSOCCPP	VBO
2	0.2	0.4	0.7
4	0.45	0.7	0.85
6	0.51	0.8	0.95
8	0.7	0.95	1.1
10	1.1	1.4	1.6
12	1.5	1.7	1.9
14	1.9	2.1	2.3
16	2.3	2.5	2.7

metric to show the performance of the DRMDOCP against the existing method. While considering the size of the data packet is 15KB and the throughput of the DRMDOCP model is 175bps. Then the throughput of the existing GSOCCPP [19] and VBO [17] are 160bps, 147bps respectively. In this way, the throughput gets increased while increasing the size of data packets from source node. Throughput of proposed DRMDOCP is considerably increased by 10% and 19% than the existing methods.

Figure 5 demonstrates the convergence plot of throughput with the size of the data taken in the ranges from 15KB to 150KB based on the three different methods namely DRMDOCP, GSOCCPP [19], and VBO [17]. As shown in the figure, the throughput increased with an increasing number of sizes of data packets. Compared to other existing optimization methods, the proposed multi-objective optimization provides superior performance in terms of achieving higher throughput. The reason behinds this improvement is to find the higher bandwidth capacity and the data transmission rate of the controller. This helps to improve the number of data packets effectively delivered from source to destination pair in specific time.

Finally, table 5 and figure 6 given above show the average latency comparisons made for three different optimization methods namely,DRMDOCP, GSOCCPP [19], and VBO [17] respectively. The

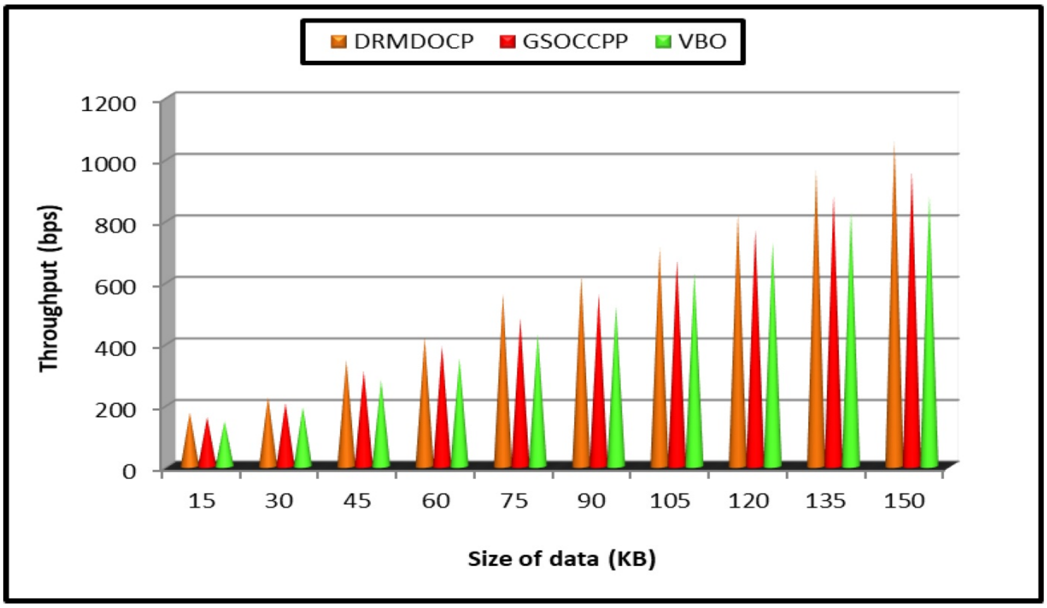

Figure 5: Convergence plots of throughput

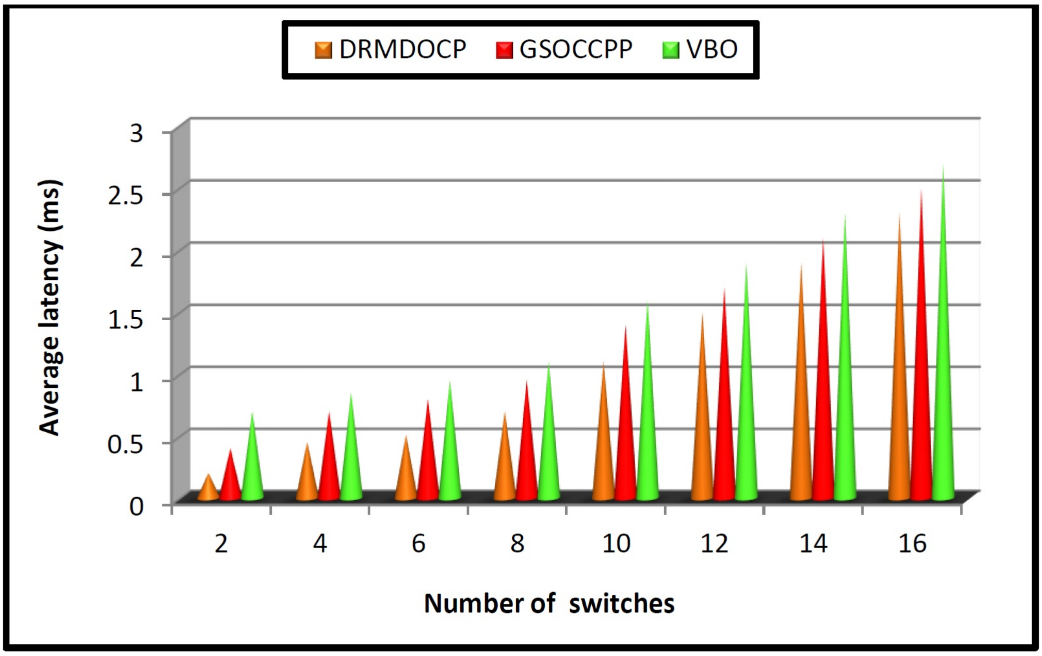

Figure 6: Convergence plots of average latency

above figure is clarifying that the rate of average latency is directly proportional to the unmanned vehicles involved in the simulation. In other words, increasing the number of switches causes an increase in the latency rate also. However, '2' switches are considered for simulation, the average latency being '0.2ms' using DRMDOCP and an average latency of existing GSOCCPP [19], and VBO [17] being '0.4ms' and '0.7ms'. From this result, it is inferred that the average latency is said to be minimized using DRMDOCP upon comparison with [19] and [17]. The reason behind the improvement is due to the application of the Deming Regression-based Multiobjective Dragonfly optimization technique. The regression function accurately analyzes the objective of the controller having the best fault tolerance capability as well as load balancing capacity. This helps to minimize the latency of data packet transmission from source to destination pair.

6. Conclusion

Placement of controllers is a significant process in the large-scale SDN. The performance of SDN is maximized and controller placement reduces the average latency of SDN. The regression-based optimization technique is used to improve throughput and minimize latency. The optimal solution for placement of controllers under multi-objective functions achieved through the Deming regressive multi-objective dragonfly optimization. Also, the proposed DRMDOCP techniques have experimented with aid of conventional works based on the GBN network topology. The performance of DRMDOCP technique is analyzed using factors namely packet delivery ratio, packet drop rate, throughput, and average latency and compared with state-of-the-art works. The simulation results indicate that the DRMDOCP has improved the performance of data packet delivery, throughput, and minimization of latency and drop rate when compared to the state-of-the-art works.

References

[1] J. Ali, B.-H. Roh and S. Lee, *QoS improvement with an optimum controller selection for software-defined networks*, PLoS ONE 14(5) (2019) 1–37.

[2] W.H.F. Aly, *Generic controller adaptive load balancing (GCALB) for SDN networks*, J. Comput. Net. Commun. 2019 (2019) 1–9.

[3] P. Aravind, G.P.S. Varma and P.V.G.D.P. Reddy, *Simulated annealing based optimal controller placement in software-defined networks with capacity constraint and failure awareness*, J. King Saud Univ. Comput. Info. Sci. (2021) 1–13.

[4] R. Chaudhary and N. Kumar, *PARC: placement availability resilient controller scheme for software-defined datacenters*, IEEE Trans. Vehicular Technol. 69(8) (2020) 8985–9001.

[5] A. Dvir, Y. Haddad and A. Zilberman, *The controller placement problem for wireless SDN*, Wireless Networks 25 (2019) 4963–4978.

[6] Y. Fan, L. Wang and X. Yuan, *Controller placements for latency minimization of both primary and backup paths in SDNs*, Comput. Commun. 163 (2020) 35–50.

[7] M. He, A. Varasteh and W. Kellerer, *Toward a flexible design of SDN dynamic control plane: an online optimization approach*, IEEE Trans. Network and Service Manag. 16(4) (2019) 1694–1708.

[8] A.A.Z. Ibrahim, F. Hashim, N.K. Noordin, A. Sali, K. Navaie and S.M.E. Fadul, *Heuristic resource allocation algorithm for controller placement in multi-control 5G based on SDN/NFV architecture*, IEEE Access 9 (2020) 2602–2617.

[9] A. Jalili and M. Keshtgari, *A new reliable controller placement model for software-defined WANs*, J. AI and Data Mining 8(2) (2020) 1–10.

[10] A. Jalili, M. Keshtgari, R. Akbari and R. Javidan, *Multi criteria analysis of controller placement problem in software defined networks*, Computer Commun. 133 (2019) 115–128.

[11] 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 8 (2020) 151668–151680.

[12] Y.P. Llerena and P.R.L. Gondim, *SDN-controller placement for D2D communications*, IEEE Access 7 (2019) 169745–169761.

[13] L. Mamushiane, J. Mwangama and A. Lysko, *Controller placement optimization for software defined wide area networks (SDWAN)*, ITU J. Future and Evolving Technol. 2(1) (2021) 1–21.

[14] S. Petale and J. Thangaraj, *Failure-based controller placement in software defined networks*, IEEE Trans. Network and Service Manag. 17(1) (2020) 503–516.

[15] G. Ramya and R. Manoharan, *Enhanced optimal placements of multi-controllers in SDN*, J. Ambient Intell. Humanized Comput. 2020 (2020) 1–18.

[16] A.K. Singh, S. Maurya, N. Kumar and S. Srivastava, *Heuristic approaches for the reliable SDN controller placement problem*, Trans. Emerging Tel. Technol. 31(2) (2020) 1–18.

[17] A.K. Singh, S. Maurya and S. Srivastava, *Varna-based optimization: a novel method for capacitated controller placement problem in SDN*, Front. Comput. Sci. 14 (2020) 1–25.

[18] S. Tahmasebi, N. Rasouli, A.H. Kashefi, E. Rezabeyk and H.R. Faragardi, *SYNCOP: An evolutionary multi-objective placement of SDN controllers for optimizing cost and network performance in WSNs*, Computer Networks 185 (2021) 1–42.

[19] S. Torkamani-Azar and M. Jahanshahi, *A new GSO based method for SDN controller placement*, Comput. Commun. 163 (2020) 91–108.

[20] A. Zilberman, Y. Haddad, S. Erlich, Y. Peretz and A. Dvir, *SDN wireless controller placement problem-the 4G LTE-U case*, IEEE Access 9 (2021) 16225–16238.