

## Research Article

# A Novel Hybrid Optimization Algorithm for Scalable Video Coding in an SDN

Li Wang<sup>1</sup> and Xiaokai Wang<sup>2</sup>

<sup>1</sup>School of Mathematics & Physics, Jinzhong University, Jinzhong 030619, Shanxi, China

<sup>2</sup>School of Physics and Electronic Engineering, Shanxi University, Taiyuan 030006, Shanxi, China

Correspondence should be addressed to Li Wang; [xgwangli@jzxy.edu.cn](mailto:xgwangli@jzxy.edu.cn)

Received 28 June 2020; Revised 8 September 2020; Accepted 15 September 2020; Published 6 October 2020

Academic Editor: Georgios I. Giannopoulos

Copyright © 2020 Li Wang and Xiaokai Wang. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Scalable Video Coding (SVC) is a powerful solution to video application over heterogeneous networks and diversified end users. In the recent years, works mostly concentrate on transported layers or path for a single layer in the Software-Defined Network (SDN). This paper proposes the Novel Hybrid Optimization Algorithm for Scalable Video Coding (NHO-SVC) based on Genetic Algorithm to select the layer and path simultaneously. The algorithm uses the 0/1 knapsack programming model to set up the model, predicts the network states by the Autoregressive Integrated Moving Average Model (ARIMA), and then, makes decision based on Genetic Algorithm.

## 1. Introduction

In the recent years, Scalable Video Coding (SVC) [1] has arisen as a powerful solution over heterogeneous networks and diversified end users, such as Wireless Multimedia Communication, Video on Demand, Remote Monitoring, and Digital Video Broadcasting. However, due to the diversity of users and the heterogeneity of the network, the video server is required to provide differentiated services, which is difficult to run under the TCP/IP network architecture. To face the advance in mobile 3D display, article [2] proposed a QoE-oriented transcoding approach, which uses a linear mean opinion score interpolation method to further reduce the cumbersome manual work of preparing QoE patterns. In another scene of dynamic traffic steering, article [3] tried to provide a framework for addressing several well-known limitations of video streaming to control forwarding paths of on-demand live video streams.

SVC is an extension of the H264/AVC standard, which can produce a base layer and some enhancement layers by one encoding according to the client request and network environment. When the network is poor, only the base layer will be provided to meet the basic needs. Waiting for the

environment turn to better, enhancement layers will be gradually added on the basis of the base layer. SVC is adapted to the heterogeneous and time-varying network, but its flexibility is limited in the TCP/IP network due to the lack of the real-time information about network devices.

Facing this problem, the Software-Defined Network (SDN) [4] has arisen with the separation of the control plane and data plane which can implement some richer functions. Especially, the OpenFlow has been proposed as a reliable realization of the SDN [5]. The controller of the SDN obtains the global network topology and link states to programme the transmission path, while the switch simply forwards data according to the forwarding rules.

This article designs a novel algorithm for the transmission of scalable video in the SDN. Its innovation is that it can simultaneously and efficiently plan the number of layers and each transmission path. This article is organized as follows. Section 2 introduces some related algorithms for planning video layers and paths. Section 3 explains the network model corresponding to this algorithm. Section 4 specifies the mathematical model and solution of the algorithm. Section 5 compares experiments with three typical mainstream algorithms, and Section 6 summarizes the full article.

## 2. Related Work

However, recent work almost concentrates on adjusting the number of layers or optimizing the path for the single layer, which is very inefficient and wastes bandwidth. In [6], the decision model is established on the weighted average of delay and packet loss rate to programme the transmission path. Article [7] is based on transmission of different wavelet information, which is mainly aimed at the wireless network. Article [8] works on the client with the dynamic adaptive streaming over HTTP (DASH) with SVC and uses a buffer-aware layer selection algorithm to choose some layers to transmit. In [9, 10], the multidomain SVC optimization model is established to seek the best path. Unfortunately, if we only select the appropriate number of layers, it will cause low transmission efficiency. On the other hand, if we only plan a suitable path for the base layer without dynamically adjusting the number, this may cause bandwidth waste.

Inspired by article [11], we try to combine the dynamic adjustment for the scalable video layer and the selection of layer path. In [11], the joint decision problem is formulated as a Markov Decision Process (MDP) and solved with Q-Learning, while article [12] solves it with the Artificial Fish Swarm Algorithm (AFSA). However, the MDP requires the correlation between different states and, then, to calculate the probability transfer matrix, which performs poorly in the face of complex and changing network environments, and Q-Learning needs a lot of resources to search and store Q-Table. AFSA is a simulation algorithm for solving complex optimization problems proposed by simulating natural ecosystem mechanisms, but its disadvantages are low convergence accuracy and poor efficiency.

Therefore, we have designed a model to optimize layers and paths simultaneously, that is, the Novel Hybrid Optimization Algorithm for SVC (NHO-SVC). NHO-SVC is designed in the SDN, so it can take advantage of the controllability and programmability of OpenFlow. The core work is to analyse the state of the network and make plans for the number of layers and paths. We encode the network system on the 0/1 multiknapsack programming model, which means that all feasible end-end paths are calculated in advance and the selected path are set to 1. The states of network at the next slot are given by the Autoregressive Integrated Moving Average Model (ARIMA), which is used with the Genetic Algorithm to solve the model. According to the experimental results, the packet loss rate and delay are reduced for the whole network resources to achieve effective configuration.

## 3. System Architecture

In this section, we present the SDN architecture shown in Figure 1. The OpenFlow controller realizes the control function through the global network view, and the OpenFlow switch forwards packets according to control instruction to represent data plane. The traffic flow is divided into two parts, the scalable video flow providing guaranteed Quality of Service (QoS) and the non-scalable video flow discarded when the bandwidth is insufficient.

The architecture includes four parts: the SVC video server, receiver, OpenFlow controller, and OpenFlow switch. The scalable video is encoded and stored in the server and selectively forwarded by the OpenFlow controller.

In SVC, the enhancement layer cannot be encoded independently, and it must be coded on the previous layer. The strategy of the number of layers is to calculate which adjacent layers will be transmitted, and there will be no situation where only the enhancement layer is transmitted but the base layer is not transmitted.

## 4. System Module

**4.1. Problem Model.** Suppose that SVC video server can transmit  $L$  layers at the time slot  $t$ ,  $w_i$  is the required bandwidth for  $i$  layer, and  $v_i$  is the income to transmit  $i$  layer,  $1 \leq i \leq L$ . There are  $N$  network nodes between the SVC video server and receiver, forming  $S$  paths.

**4.1.1. Objective Function.**

$$F = \max \sum_{j=1}^s \sum_{i=1}^L Z_{ij}^t \times v_i, \quad (1)$$

$$Z_{ij}^t = \begin{cases} 1, & \text{choose path } j \text{ for layer } i \text{ at slot } t, \\ 0, & \text{otherwise.} \end{cases}$$

There are four constraints in this model.

Flow constraint: assuming  $P_j$  is the set of path  $j$ ,  $Q_{jk}^t$  is the available bandwidth of the link  $k$  in path  $j$  at the time slot  $t$ .

$$H_{ik}^t = \begin{cases} 1 \exists j, & \text{s.t. } Z_{ij}^t = 1, \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

$$\sum_{i=1}^L H_{ik}^t \times w_i \leq Q_{jk}^t \forall j \in \{1, \dots, S\}, \quad k \in P_j.$$

Delay constraint:

$$d_{jt} \leq D, \quad (3)$$

where  $d_{jt}$  is the delay of path  $j$  at the time slot  $t$  and  $D$  is the maximum acceptable delay.

The layer number of restrictions: each layer can be transmitted only once or none.

$$\sum_{j=1}^s Z_{ij}^t \in \{0, 1\}, \quad 1 \leq i \leq L. \quad (4)$$

The transmission sequence of restrictions: due to the characteristics of SVC video, only the base layer can be independently decoded, the enhancement layer must

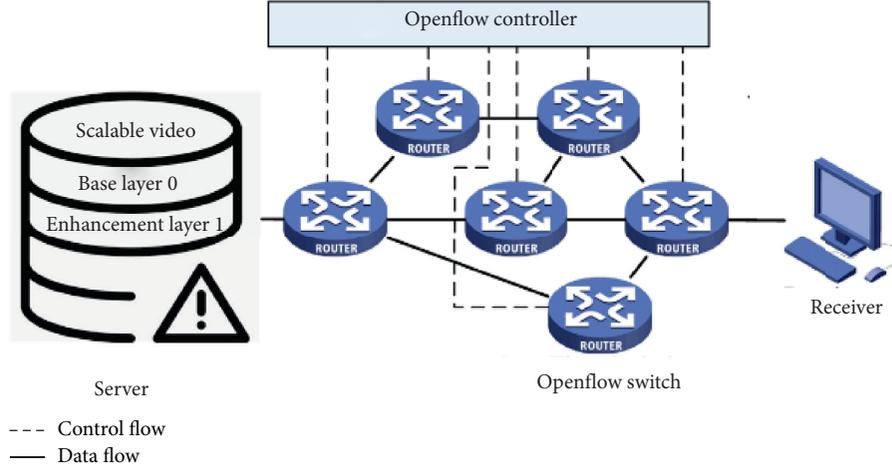


FIGURE 1: System architecture of SVC in the SDN.

rely on prior layer, so the SVC layer must be continuous and the base layer of each frame must be transmitted to ensure the continuity of video playback. If  $Z_{ij}^t = 1, 1 \leq j \leq S$ , then  $\exists i' = \max\{i-1, 1\}$ , such that  $Z_{i'n}^t = 1, 1 \leq n \leq L$

**4.2. Model Solution.** The solution of the problem model is NP-complete theoretically with complexity  $O(2^{LS})$ . This article takes Genetic Algorithm [13] to simulate mutation, crossover, and selection. It encodes the possible solution into a vector called an individual, and each element of the vector is called a gene. The evaluation of the Fitness Function for each individual is used in the genetic operator so as to get a new group.

**4.2.1. Encoding.** Taking into account the characteristics of optimization, we directly map the solution space to the gene space represented with matrix  $T$ , whose size is  $L \times S$ .  $L$  indicates the total number of layers, and we calculate  $S$  paths from server to receiver. So, the equation,  $T_{ij}^t = 1$  means that the data of layer  $i$  will be transmitted through the path  $j$ .

#### 4.2.2. Fitness Function

The Fitness Function is defined as follows:

$$f(Z) = \sum_{j=1}^S \sum_{i=1}^L Z_{ij}^t \times v_i - \max\{v_i\} \times \sum_{j=1}^S G_j(T) - \max\{v_i\} \times \sum_{i=1}^L H_i(T), \quad (5)$$

where  $G_j(T)$  is the penalty function for insufficient bandwidth of path  $j$  and  $H_i(T)$  is the penalty function for repeating. Penalty functions are defined as follows:

$$G_j(T) = \max\left(\alpha_j \sum_{k \in P_j} \left(\sum_{i=1}^L H_{ik}^t \times w_i - Q_{jk}^t\right), 0\right), \quad (6)$$

$$H_i(T) = \max\left(\beta_i \left(\sum_{j=1}^S T_{ij}^t - 1\right) \times w_i, 0\right),$$

where  $\alpha_j$  and  $\beta_j$  are penalty coefficients between 0 and 1. In experiments, coefficients will be initialized to 0.5 and, then, further optimized by Algorithm 1.

#### 4.2.3. Genetic Operator

**Selection:** it is based on the assessment of fitness of the individuals and uses random selection in roulette mode proportional to the fitness value.

**Crossover:** the gene space exists as a matrix equivalent to  $L \times S$ . The crossover mode is set to one-point crossover, that point  $k \in \{0, 1, \dots, LS-1\}$  is randomly selected, and all the bits after  $k$  are exchanged by their parents.

**Mutation:** one or more genes are selected randomly and, then, mutated with probability  $P_m$ , that is, to negate the value of the current gene.

**4.2.4. Solution Algorithm.** In the algorithm, the following variables are defined:

$P_c$ : probability of crossover

$P_m$ : probability of mutation

$M$ : population size

$G$ : number of iterations to terminate evolution

Action: decision results

**4.3. ARIMA Prediction Model.** In the architecture, the base layer of the scalable video is QoS flow, and others are non-QoS flow. In order to guarantee the transmission reliability,

```

Input: Link_States, Pc, Pm, M, G
Output: Action
Initialize: Action = [0, 0]
//The first dimension represents the number of layers, and the second dimension represents the path.
Random generate Pop1 of M individual
do {
Calculate  $f(z)$  based on equation (5) for all in Pop1;
Generate Pop2 of zero population;
Add best two individuals into Pop2;
do {
Selection;
if (random (0, 1) < Pc) Crossover;
if (random (0, 1) < Pm) Mutation;
Add new individuals into Pop2;
} until (population of Pop2 == M)
Pop1 = Pop2;
} until (best  $f(z)$  keep unchanged for 5 times or iteration num > G)
if (best individual != [0, 0])
return Action = best individual;
else
return Action = [z, 0];

```

ALGORITHM 1: The novel hybrid optimization algorithm for scalable video coding.

the link states need to be acquired in real time during the decision process. Due to the characteristics of high self-similarity, time-varying, and nonlinear, the ARIMA model [14] is adopted in this paper to predict the available bandwidth at the next slot. Firstly, the ARIMA model transforms the nonstationary time series into the stationary time series, then to establish the prediction model by regression of the dependent variable's lag value and the present and lag value of random error term, and finally, we can take it as  $Q_{jk}^t \sim \text{ARIMA}(p, d, q)$ . Among them, the set  $\{Q_{jk}^t\}$  is constituted by the historical available bandwidth of the link  $k$  of the path  $j$ .  $p$  represents the autoregressive process of order,  $d$  represents the sliding average process of order, and the  $q$  represents the difference order.

Difference operator:

$$\begin{aligned}
\Delta Q_{jk}^t &= Q_{jk}^t - Q_{jk}^{t-1} = Q_{jk}^t - LQ_{jk}^t = (1-L)Q_{jk}^t, \\
\Delta^2 Q_{jk}^t &= \Delta Q_{jk}^t - \Delta Q_{jk}^{t-1} \\
&= (1-L)Q_{jk}^t - (1-L)Q_{jk}^{t-1} \\
&= (1-L)^2 Q_{jk}^t, \\
\Delta^d Q_{jk}^t &= (1-L)^d Q_{jk}^{t-1},
\end{aligned} \tag{7}$$

where  $L$  is the backward shift operator. The  $d$ -order difference is applied for  $Q_{jk}^t$  to get stationary sequence  $w_t = \Delta^d Q_{jk}^t = (1-L)^d Q_{jk}^t$  so as to establish tARIMA  $(p, d, q)$  model. The model form is

$$\begin{aligned}
w_t &= \phi_1 w_{t-1} + \phi_2 w_{t-2} + \dots + \phi_p w_{t-p} + \delta + u_t + \theta_1 u_{t-1} \\
&\quad + \theta_2 u_{t-2} + \dots + \theta_q u_{t-q},
\end{aligned} \tag{8}$$

recorded as follows:

$$\Phi(L)\Delta^d x_t = \delta + \Theta(L)u_t, \tag{9}$$

where

$$\begin{aligned}
\Phi(L) &= 1 - \phi_1 L - \phi_2 L^2 - \dots - \phi_p L^p, \\
\Theta(L) &= 1 + \theta_1 L + \theta_2 L^2 + \dots + \theta_q L^q, \\
\{u_t\}, \quad t &= 0, \pm 1, \pm 2, \dots
\end{aligned} \tag{10}$$

The set  $\{u_t\}$  is the white noise sequence and  $\delta$  is constant. The next is the equation section.

## 5. Experimental Results

In order to evaluate the performance of NHO-SVC in SDN, there are two experiments designed in this section. Firstly, we verify the feasibility of NHO-PM and, then, test the effectiveness of NHO-SVC compared with the other three SVC algorithms proposed in article [12, 15].

**5.1. Experimental Setup.** The simulation network is built on Mininet [16] with Open vSwitch [17] as the switch and POX as the controller including three parts described below. In actual application scenarios, considering the existence of video caching and avoiding excessive fluctuations, the optimization algorithm is executed every minute. In the experiment, in order to make the data fluctuations more obvious, it is executed every second.

**Prediction module:** it chooses the optimal model to predict the value at the next slot utilizing 60 sets of historical data. To improve the efficiency, the prediction thread is established for each link.

Decision-making module: the initial population size is 64, the crossover rate and the mutation rate are 0.8 and 0.1, respectively, and the iteration termination condition is that the optimal adaptation value is continuous 5 times unchanged or iterated over 1000 times. To accelerate iteration, every optimal individual is directly chosen to the next generation. In addition, a function about the mandatory transmission of base layer is added to decision-making module that enabled when the decision is [0, 0] that means none of layers will be transmitted. Thereby, the mandatory transmission can ensure the continuity of transmission in the premise of minimizing the loss of non-QoS flow so that the scalable video can be smoothly played in the receiver.

- (1) Main module: it presides over the OpenFlow network, listens to the device data, communicates with other modules, and sends decision results to the servers.
- (2) In this simulation, we establish three UDP transmission links according to the network topology of Figure 1, with the maximum bandwidth of each link set to 1024 kbps. Encoding a scalable video into two layers, the resolution of base layer of is  $320 \times 160$ , and the resolution of enhancement layers is  $630 \times 360$ . The average code rate of the base layer is 91 kbps, PSNR = 37.759 dB, and the average code rate of the enhancement layer is 124 kbps, PSNR = 50.511 dB.

**5.2. Performance Index.** The following indexes are defined to evaluate the performance of NHO-SVC:

- (1) Packet loss rate (PLR) and average packet loss rate
- (2) The packet loss only happens in non-QoS flow, and the influence of SVC can be reflected by PLR  $L$  and average PLR  $\bar{L}$  that calculated as

$$L(i) = \frac{C_{\text{QoS}} + C_{\text{non-QoS}} - BW}{C_{\text{non-QoS}}}, \quad (11)$$

$$\bar{L} = \frac{\sum_{i=1}^n L(i)}{n},$$

where  $C_{\text{QoS}}$  denotes SVC video data,  $C_{\text{non-QoS}}$  denotes non-QoS video data,  $BW$  denotes the total link bandwidth,  $L(i)$  denotes the PLR at the time slot  $i$ , and  $n$  denotes the total number of slots

**5.2.1. Peak Signal to Noise Ratio (PSNR).** The number of layers is continuously adjusted during transmission, resulting in fluctuations of the received video quality. The average PSNR can objectively evaluate video quality as defined:

$$\bar{P} = \frac{\sum_{i=1}^F P(l_i)}{F}. \quad (12)$$

$F$  denotes the total number of frames in the video sequence and  $P(l_i)$  denotes the PSNR value corresponding to the  $l_i$  layer in the  $i$  frame.

**5.3. Feasibility Verification.** Article [18] has designed a heuristic ARIMA model to make prediction, that is ARIMA (5, 1, 7).

In this section, we first compare NHO-PM with ARIMA (5, 1, 7) to verify the feasibility and, then, test the effect of NHO-SVC with and without NHO-PM.

The evaluation indexes of the prediction results are defined as follows:

Root Mean Square Error (RMSE):

$$\text{RSME} = \sqrt{\frac{1}{n} \sum_{t=1}^n (y_t - \hat{y}_t)^2}. \quad (13)$$

Average Absolute Error (MAE):

$$\text{MAE} = \frac{1}{n} \sum_{t=1}^n |y_t - \hat{y}_t|. \quad (14)$$

Mean Absolute Relative Error (MAPE):

$$\text{MAPE} = \frac{1}{n} \sum_{t=1}^n \frac{|y_t - \hat{y}_t|}{y_t} \times 100\%, \quad (15)$$

where  $y_t$  represents the actual value and  $\hat{y}_t$  represents the predicted value.

Utilizing the 69 sets of traffic data collected from the network server in [18], the first 48 sets are used as training data to get the ARIMA (5, 1, 7) model, and the last 21 sets are used as the test set. Similarly, NHO-PM takes the 48 historical data as training data to predict the residual data. In Figure 2, we counted 21 sets of network traffic under different algorithms. It shows that both NHO-PM and ARIMA (5, 1, 7) can track the actual value, while NHO-PM is closer to the actual value.

After that, we further compare normalized error coefficients of the two algorithms, shown as Figure 3, and it can be obviously seen that NHO-PM is superior in MAE, RMSE, and MAPE. Because NHO-PM can update the model with time series, while ARIMA (5, 1, 7) is fixed, using historical data failed to take advantage of the latest acquisition.

The changes of the PLR with and without NHO-PM are shown in Figure 4. The maximum PLR without NHO-PM is 11.8363% occurring at the 27nd slot, and the average PLR is

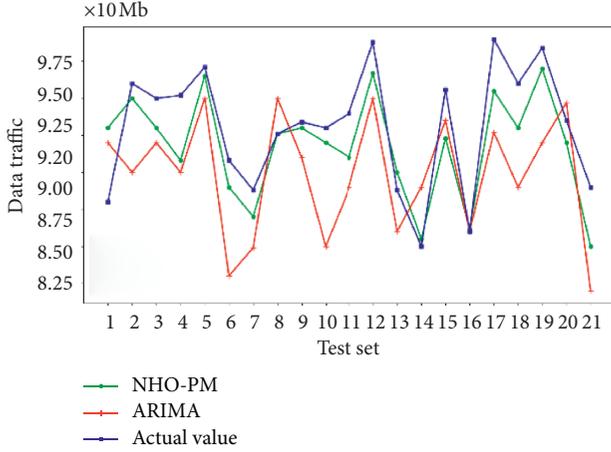


FIGURE 2: Feasibility verifying test by counting network traffic.

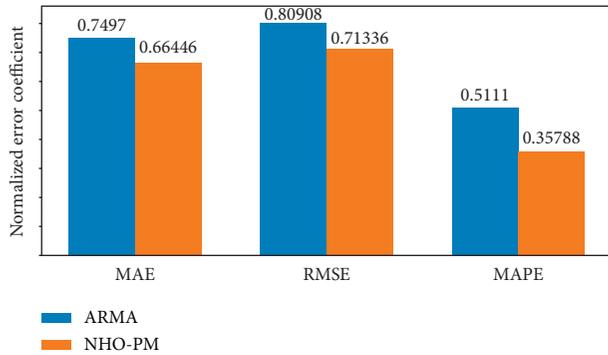


FIGURE 3: The comparison of normalized error coefficients.

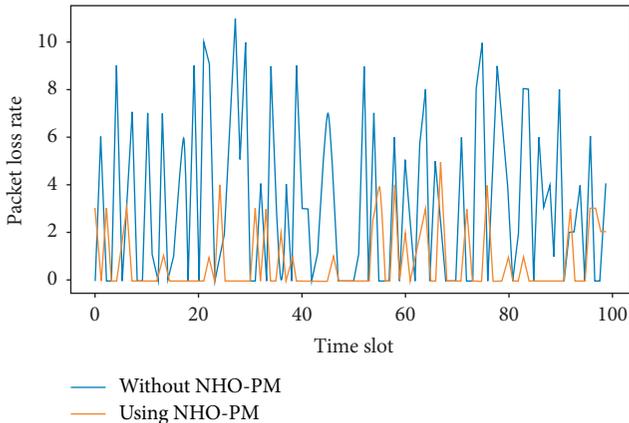


FIGURE 4: Packet loss test with and without NHO-PM.

3.25%; the maximum PLR with NHO-PM is 5.2345% occurring at the 67th slot, and the average PLR is 0.73%. It can be clearly seen that NHO-PM has a significant effect on reducing the PLR.

Figure 5 further shows the selected paths for the base layer and enhancement layers under the NHO-PM algorithm. The vertical axis represents optional paths, and the

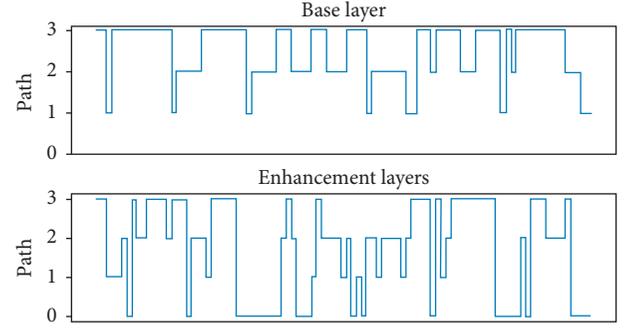


FIGURE 5: The transmission paths of each layer.

horizontal axis represents layers of different time slots. When enhancement layers are marked as path 0, it means only the base layer is transmitted at that slot. This diagram visually shows the changing state of the number of layers and paths.

**5.4. Contrast Test.** We compare NHO-SVC with the other three SVC algorithms at the average PLR and the average PSNR.

**5.4.1. Algorithm 1.** It transmits both the base and enhancement layers simultaneously and randomly selects a unique path for each layer from three paths.

**5.4.2. Algorithm 2 ([15]).** It guarantees the transmission of base layer, while the enhancement layers try its best to provide service. In addition, the path of the base layer can be adjusted, but enhancement layers are always transmitted from the shortest path. The objective function is

$$\text{Minimize } \sum_{j=1}^J \{K_1 L_j + K_2 PLB_j\}, \quad (16)$$

where  $L_j$  is the delay of the path  $j$  and  $PLB_j$  is the PLR of the enhancement layer on the path  $j$ . There are three provided paths ignoring the delay on the path, setting  $K_1 = 0$ .

**5.4.3. AFSA ([12]).** It establishes a prediction model based on the MDP and solves it with the Artificial Fish Swarm Algorithm.

As shown in Figure 6, we show the comparison of different transmission algorithms.

**(1). Average PLR.** The average PLR of these algorithms is, respectively, 0.73%, 8.76%, 5.43%, and 2.17%. NHO-SVC is significantly lower than others because when the link bandwidth is insufficient, it can not only reduce the number of layers but also adjust the path; moreover, the mandatory transmission of the base layer is the choice of minimal impact to other non-SVC flow. Comparing ALGORITHM 1 with 2, the PLR of ALGORITHM 2 is smaller because its transmission path can adaptively change and enhancement layers can be duly discarded, while ALGORITHM 1 simply

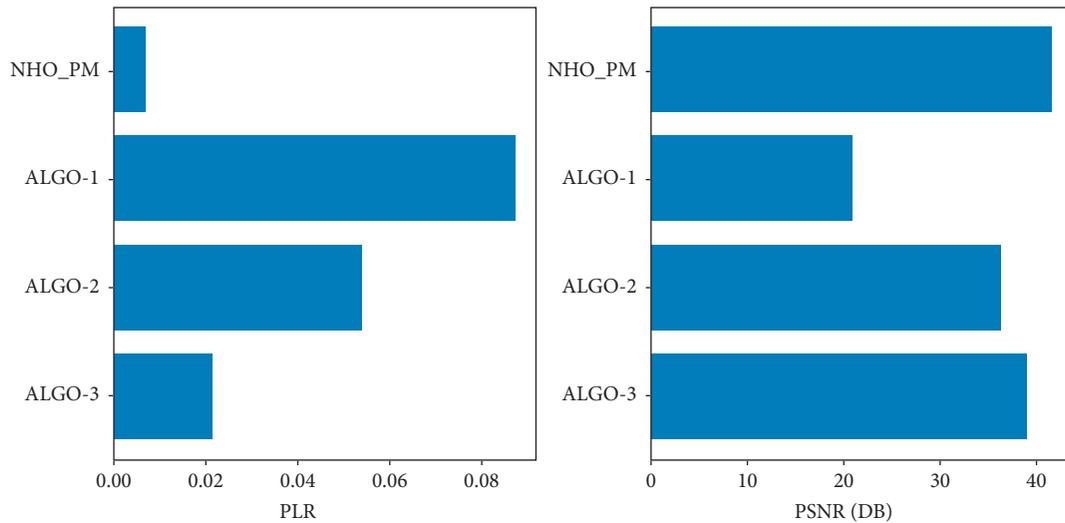


FIGURE 6: The comparison of different transmission algorithms.

assigns a random path. The performance of AFSA is also excellent, but it is still inferior to NHO-SVC because AFSA uses the action corresponding to the optimal value at the next slot when updating the AFSA, which will overestimate the sampled action, and for the unsampled action, that will not be the best.

(2). *Average PSNR.* The average PSNR of these algorithms is, respectively, 41.573 dB, 20.933 dB, 36.345 dB, and 38.955 dB. Analysing the reason, ALGORITHM 1 randomly selects a unique path to transmit the base layer or enhancement layers, which cannot guarantee QoS when the bandwidth is insufficient, so the SVC video quality is the worst. Several other three algorithms are close, of which NHO-PM performs best, ALGORITHM 2 is the second, and AFSA is relatively poor. NHO-PM can dynamically adjust the number of layers, and ALGORITHM 2 may also discard enhancement layers, so it is slightly lower in the PSNR. In addition, ALGORITHM 2 always sends two layers of the SVC video, so the video quality received in some slots is better than NHO-SVC. But, in other slots, NHO-SVC can transmit enhancement layers without packet loss while these may be lost in ALGORITHM 2; on average, the PSNR related to network status of NHO-SVC is better. The strategy obtained by AFSA is deterministic, in these experiments, when the neural network did not converge at the optimal position by a nonlinear approximation, which may lead to a larger deviation between the predicted value and the true value. In NHO-SVC, it is to evaluate the states of network at the next slot, and finally, it uses the Genetic Algorithm to solve the mode, which is very concise and efficient. In conclusion, under the guarantee of QoS, NHO-SVC can sacrifice the video quality with moderate network resources.

(3). *Convergence.* The transmission performance for the scalable video is basically close, so we further compare their convergence. The results are shown in Figure 7; both NHO-PM and AFSA can stably converge to the optimal value, while convergence speed of NHO-PM is slightly faster than that of AFSA.

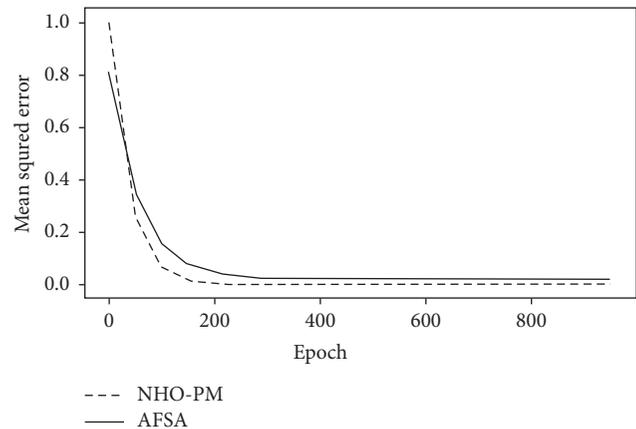


FIGURE 7: The convergence of NHO-PM and AFSA.

## 6. Conclusions

Based on the controllability and programming of SDN, NHO-SVC is proposed in the paper to make a strategic decision about the number of layers and the path of each layer for the SVC video at the next slot simultaneously. In the face of this problem, we should use concise and efficient means to solve it from the overall perspective. The experimental results show the proposed algorithm can concisely and effectively guarantee the quality of service supporting for SVC.

## Data Availability

The data used to support the findings of this study can be obtained from the corresponding author upon request.

## Conflicts of Interest

The authors declare that there are no conflicts of interest regarding the publication of this paper.

## Acknowledgments

This work was supported by the Shanxi Province Science and Technology Research Projects (No. 20110321025-02) and Shanxi International Science and Technology Cooperation Project (No. 2014081027).

## References

- [1] P. Yin and F. Gomila, "Multi-view video coding using scalable video coding," 2008.
- [2] Y. Liu, S. Ci, and H. Tang, "QoE-oriented 3D video transcoding for mobile streaming," *ACM Multimedia*, vol. 8, no. 3, 2012.
- [3] Q. Wang, K. Xu, and R. Izard, *GENI Cinema: An SDN-Assisted Scalable Live Video Streaming Service*, International Conference on Network Protocols, Berlin, Germany, 2014.
- [4] Y. Li and M. Chen, "Software-defined network function virtualization: a survey," *IEEE Access*, vol. 3, pp. 2542–2553, 2015.
- [5] K. Suzuki, K. Sonoda, and N. Tomizawa, "A survey on OpenFlow technologies," *IEICE*, vol. 97, no. 2, pp. 375–386, 2014.
- [6] D. E. Henni, A. Ghomari, and Y. Hadjadj-Aoul, *Videoconferencing over OpenFlow Networks: An Optimization Framework for QoS Routing*, IEEE CIT/IUCC/DASC/PICOM, Berlin, Germany, 2015.
- [7] H. Yang, L. Qing, X. He, and S. Xiong, "Scalable distributed video coding for wireless video sensor networks," *IEICE Transactions on Information and Systems*, vol. 2, no. 1, pp. 20–27, 2018.
- [8] S. Mori and M. Bandai, *A Quality-Level Selection for Adaptive Video Streaming with Scalable Video Coding*, IEICE, Berlin, Germany, 2018.
- [9] J. Samuelsson, J. Enhorn, and R. Sjöberg, "Hybrid codec scalable video," 2019.
- [10] H. H. LEE, J. W. Kang, and J. S. Choi, "Method for coding and decoding scalable video and apparatus using same," 2019.
- [11] Y. Yue, Y. Ran, and S. Chen, "Joint routing and layer selecting for scalable video transmission in sdn," IEEE GC Workshops, Berlin, Germany, 2015.
- [12] Z. Liu, "SVC video transmission optimization algorithm in software defined network," *China Communications*, vol. 15, no. 10, pp. 143–149, 2018.
- [13] S. Mirjalili, "Genetic Algorithm," *Handbook of Evolutionary Algorithms and Neural Networks*, vol. 50, pp. 43–55, 2019.
- [14] G. P. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, 2019.
- [15] C. A. Hasrouty, C. Olariu, and V. Autefage, *SVC Videoconferencing Call Adaptation and Bandwidth Usage in SDN Networks*, GLOBECOM, Berlin, Germany, 2017.
- [16] B. Lantz and B. O'Connor, "A mininet-based virtual testbed for distributed SDN development," *ACM SIGCOMM Computer Communication Review*, vol. 45, no. 4, pp. 365–366, 2015.
- [17] B. Pfaff, J. Pettit, and T. Koponen, "The design and implementation of open vswitch," *USENIX*, vol. 15, pp. 117–130, 2015.
- [18] K. Xue, Z. Z. Li, and L. Liu, "Network traffic prediction based on ARIMA model," *Microelectronics & Computer*, vol. 21, no. 7, pp. 84–87, 2004.