Multi-objective Hybrid Evolution with Information Entropy Awareness for Controller Placement

Xing Li
School of Telecommunications Engineering
Xidian University
Xi'an, China
xli_2018@stu.xidian.edu.cn

Zhanqi Xu
School of Telecommunications Engineering
Xidian University
Xi'an, China
zqxu @mail.xidian.edu.cn

Fan Yang
School of Telecommunications Engineering
Xidian University
Xi'an, China
fany@mail.xidian.edu.cn

Yunbo Li
China Mobile Communication Corp. Research Institute
CMCC
Beijing, China
liyunbo@chinamobile.com

***Abstract*—In software defined wide-area networks, the number and location of controllers in the optimization model have a significant impact on network performance. Compared with the traditional single-objective model or quasi-multi-objective model that can be transferred to the single-objective model, multi-objective models can provide more comprehensive solutions to the problems by concerning controller deployment at one time, which makes network operators use different solutions to accommodate various scenarios better. In this paper, an actual multi-objective model is built to optimize controller deployment by considering deployment cost, load difference, and propagation delay. To solve this model, we propose an algorithm by specially designing the hybrid initialization method to generate an initial population that balances diversity and convergence. After that, we design the mechanisms of encoding conversion, information entropy awareness, hybrid evolution, and perturbation modification. These mechanisms are particularly constructed for the proposed algorithm to solve the problems in the evolution process and to improve the global search ability of the algorithm for obtaining superior Pareto sets. Finally, we validate the effectiveness and generality of the proposed algorithm by comparing its Pareto sets with those of other algorithms in Internet2 OS3E network from various aspects.**

Keywords—Multi-Objective Optimization (MOO); Pareto Front (PF); Software Defined Networking (SDN); Controller Placement Problems (CPPs); Hybrid Evolutionary Algorithm

I. INTRODUCTION

The Wide-Area Network (WAN) based on Dense Wavelength Division Multiplexing (DWDM) has the advantage of high transmission capacity. However, with the development of mobile services and the popularity of 5G technology, Internet applications are gradually migrating to the cloud. The cloud data center network needs the WAN with high bandwidth, low latency, and high quality, which leads to increasing obvious defects such as long cycles of WAN services provisioning [1]. As a result, Software Defined Networking (SDN) is applied to meet the requirements of WAN services and operations [2], in which the solution to Controller Placement Problems (CPPs) is of great significance.

Ahmadi *et al.* [3] used Non-dominated Sorted Genetic Algorithm (NSGA) to optimize the load balance of controllers and reduced the propagation delay, which is classified into the delay between controllers, as well as the delay between controllers and switching nodes. Liao *et al.* [4] computed the mutation probability of the evolutionary algorithm by particle swarm optimization and used the algorithm to decrease the propagation delay and optimize the load difference of the controllers. However, the models in both studies ignored the deployment cost, leading to the reduction in the practicality of results due to high cost. Reference [5] used improved NSGA (NSGA-II) to optimize the load balance and the average network connectivity, providing an idea to reduce the deployment cost.

Pareto-based Optimal COntroller-placements (POCO) [6] is a framework to solve the CPPs. Samir *et al*. [7] and Jalili *et al.* [8] embedded different evolutionary algorithms into POCO and obtained some favorable solutions to the CPPs in a network. However, the optimization objectives of POCO are not comprehensive, with only two objectives optimized at the same time, and the efficiency of its solving algorithm needs to be improved.

In order to obtain the Controller Deployment Schemes (CDSs) of CPPs, the optimization models need to be built first and solved by optimization algorithms. This paper solves CPPs by an actual multi-objective optimization model and relevant algorithm to obtain a set of optimal CDSs with different objectives such as cost, load difference, and latency. Therefore, the operators can select the most suitable CDS according to the needs and preferences of actual application, so as to achieve the best balance between cost and multiple network performance. However, the ordinary optimization models and their corresponding solving algorithms tend to focus on the tradeoff between different network performance, such as latency and load balance, without incorporating the impact of cost on the deployment scheme, which results operators from obtaining lower-cost CDSs with excellent network performance.

This paper improves the existing models and algorithms to adapt to realistic CDSs. The multi-objective optimization model is built to optimize the control network by jointly optimizing propagation delay, load difference, and deployment cost. As a tri-objective optimization model, the solution to the model is more challenging compared to the optimization models in [6][7]. The proposed model is solved by the Multi-Objective Hybrid Evolution with Information Entropy Awareness (MOHEIEA), aiming to improve the global efficiency of solutions through various improvement mechanisms.

The rest of this paper is organized as follows: Section II presents the multi-objective model and analyzes the holding relationship among the objectives. Section III introduces the algorithm designed specially and analyzes the effectiveness of various improvement mechanisms. Section IV evaluates the solution performance of MOHEIEA from multiple perspectives. Finally, a conclusion is given in Section V.

II. Proposed Model

This section combines the needs of operators and users with the constraints of network resources, from which three optimization objectives are abstracted, including the propagation delay of the control network, load difference, and deployment cost. Finally, the constraints are added to build the model. The proposed model and the designed algorithm of our work are presented in Fig. 1.

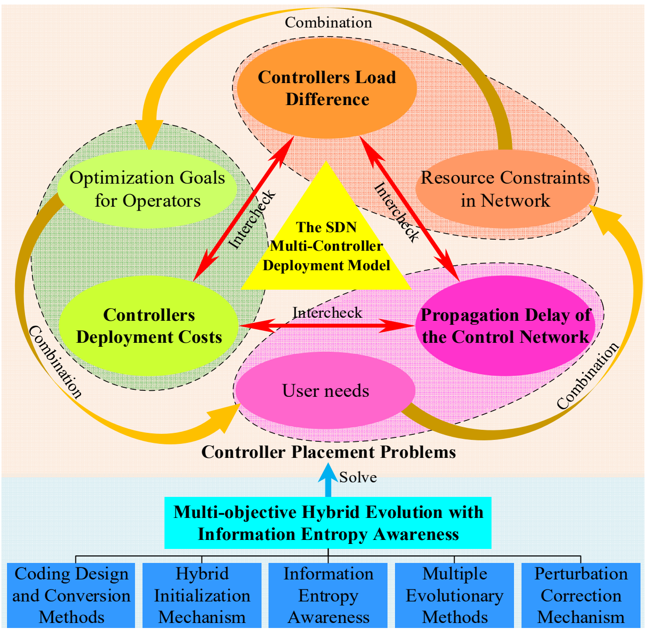


Fig. 1. Model and algorithm used in this paper

A. Symbols list

Table I gives a description of the notation in the model.

TABLE I. Summary of Notations and Symbols

Notation	Definition
$\boldsymbol{G}=(\boldsymbol{N},\boldsymbol{E})$	$\boldsymbol{G}$ is the network topology, where $\boldsymbol{N}$ denotes the network node set, and $\boldsymbol{E}$ represents the directly connected link set.
v	The propagation speed of optical signals in the network, which equals 2×10^8 m/s.
m	The total number of controllers in the network of a CDS.
$\boldsymbol{W}=[w_i]_m$	$\boldsymbol{W}$ represents the controller load set, where w_i means the number of switching nodes in the network managed by the SDN controller i.
$\boldsymbol{P}=[p_i]_{\lvert N\rvert}$	$\boldsymbol{P}$ means the controller position set, and if a SDN controller is deployed on node i, p_i=1, otherwise p_i=0.
$\boldsymbol{C}=[c_{ij}]_{\lvert N\rvert\times\lvert N\rvert}$	The control set is $\boldsymbol{C}$, and if switching node j is controlled by the SDN controller of node i, then c_{ij}=1, otherwise c_{ij}=0.
$\boldsymbol{S}=[s_{ij}]_{\lvert N\rvert\times\lvert N\rvert}$	The shortest distance set is abstracted as $\boldsymbol{S}$, where s_{ij} denotes the shortest distance between node i and node j.
β_1,β_2	Weight factor of propagation delay.

B. Optimization objectives

As shown in (1), the optimization objective of the model is to minimize the deployment cost, load difference, and propagation delay. The deployment cost is defined in (2), which represents the number of controllers in the network. In (3), the load difference represents the extreme difference in the number of switching nodes controlled by the controllers. The equation (4) shows the propagation delay obtained by weighting the maximum propagation delay between controllers and switching nodes as well as the maximum propagation delay between controllers. Both propagation delays in (4) are important in the control network, so β_1=β_2=0.5 is set to measure the control network delay in a comprehensive way.

$$min\ F(x)=(f_M, f_D, f_L)^T \tag{1}$$

$$f_M = m \tag{2}$$

$$f_D = \max_{i=1,2,\cdots,m} w_i - \min_{j=1,2,\cdots,m} w_j \tag{3}$$

$$f_L = \frac{\beta_1}{v}\cdot \max_{i,j\in N}\left(s_{ij}\cdot c_{ij}\right) + \frac{\beta_2}{v}\cdot \max_{i,j\in N}\left(s_{ij}\cdot p_i\cdot p_j\right) \tag{4}$$

Increasing the number of controllers can reduce the distance between the controllers, the distance between the controllers and the switching nodes they control, and the load on high-load controllers. It is beneficial for propagation latency and load difference optimization, but it leads to relatively higher deployment costs. The increase of the load difference means that the maximum number of switching nodes controlled by a single controller increases, reducing deployment cost and propagation delay. For reducing the propagation delay, the distance between the controllers and the switching node needs to be reduced, so it is necessary to increase the number of controllers or the load difference. In summary, the three objectives in this model are interlocked and form an actual multi-objective optimization model, which can be solved by multi-objective evolutionary algorithms.

C. Restrictive conditions

Constraints are used to restrict the properties of the final CDSs, and the mathematical expressions of the limits in the model are given below:

$$\sum\nolimits_{i=1}^{|N|} p_i = m \tag{5}$$

$$\sum\nolimits_{i=1}^{|N|} c_{ij} = 1,\quad j=1,2,\cdots,|N| \tag{6}$$

$$m \le {|N|}/{2} \tag{7}$$

Equation (5) indicates that the number of controllers in the network is m, and each node deploys one controller at most. Equation (6) means that each switching node can be controlled by only one controller. Equation (7) is used to limit the deployment cost, denoting that the number of controllers is not greater than half of the number of nodes in the network.

III. Algorithm and Analysis

Since the CPPs are NP-hard and difficult to solve [9], we propose MOHEIEA and analyze its performance in this section. Different improvement mechanisms are added at different stages of the evolution in MOHEIEA, aiming to improve the global optimal seeking ability of the algorithm to obtain a superior Pareto set.

A. Coding design

Coding design includes the coding method and the encoding conversion mechanism. The former is applied to the coding of individuals, while the latter is used to solve the problems that arise in evolution.

1) The coding method

The binary fixed-length encoding is used in our work, as shown in Fig. 2 , the length of individuals is the number of network nodes |N|, and encoded content is used to mark the deployment status of the controller, where 1 indicates that the controller is deployed at that node and 0 is opposite.

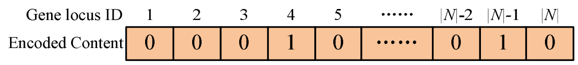


Fig. 2. Individual encoding method

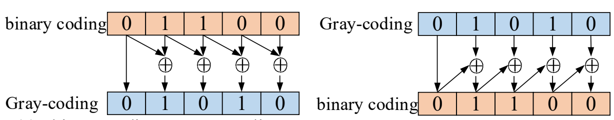
Furthermore, each switching node is controlled by the closest controller.

2) The encoding conversion mechanism

The ordinary evolutionary algorithm lacks search ability, and its offspring are generated by random crossovers or mutations from parents, which is not conducive to population search for superiority solutions in the neighborhood of the current solution. In addition, binary coding suffers from the Hamming cliff problem because the Hamming distance between some individuals is too large to be spanned by ordinary crossover and mutation, which dramatically limits the optimization capability of the algorithm.Therefore, the encoding conversion mechanism is added to MOHEIEA to cope with the problems.

$$\begin{cases} g_1 = b_1 \ , \ \ g_i = b_{i-1} \oplus b_i \\ b_1 = g_1 \ , \ \ b_i = b_{i-1} \oplus g_i \end{cases} \tag{8}$$

Suppose the binary coding of an individual is $B_N=b_1b_2\cdots b_{|N|}$, and $G_N=g_1g_2\cdots g_{|N|}$, i=2,3, ···, $|N|$ means the corresponding Gray-coding. The way of mutual conversion is shown in (8).



(a) binary coding to Gray-coding (b) Gray-coding to binary coding

Fig. 3. Encoding conversion mechanism

Fig. 3 presents an example of the encoding conversion mechanism. Converting individuals to Gray-coding before the crossover and mutation operations can amplify the practical significance of the correspondence between individuals with small coding differences and overcome the Hamming cliff problem. As a result, the algorithm's optimization-seeking ability is enhanced. Moerover, individuals are convertedto binary coding after the crossover and mutation operations to facilitate the calculation of optimization objectives and the selection of individuals.

B. Population initialization

MOHEIEA generates the initial population through cluster initialization and uniform initialization, aiming to balance the diversity and convergence of the initial population.

1) Cluster initialization

Cluster initialization generates the number of cluster centers randomly according to their range, and then obtains the location of cluster centers by the k-means clustering algorithm. Finally, the initial population is obtained.

The clustering initialization uses various attributes of the network, aiming to reduces the propagation delay and obtain the initial population with an optimal objective value. However, the cluster initialization has difficulty in accommodating the diversity of populations, and often produces the population with high similarity, leading to local optimal solutions when the population evolves.

2) Uniform initialization

Uniform initialization generates the number of controllers randomly according to their range, then deploys the controllers at random nodes in the network to get an individual, and finally repeats the above operations and obtains the initial population with more diversity.

Uniform initialization makes the initial population distributed as uniformly as possible throughout the variable space to increase the diversity of the population. However, this initialization mechanism does not incorporate the properties of the network, and therefore the obtained initial population can hardly have a superior objective value.

C. Information entropy awareness mechanism

The information entropy of the Pareto set in the objective space can be used to perceive the measures like the distributivity and convergence of the current Pareto set, so as to select the most appropriate evolutionary method in the hybrid evolutionary mechanism.

The information entropy is calculated by the influence function and the density function. The influence function is used to measure the influence degree between any two individuals in the Pareto set, and the commonly used Gaussian influence function [10] is presented in (9):

$$\psi\left(d\left(x_i,x_j\right)\right)=\frac{1}{\sigma\sqrt{2\pi}}e^{-\frac{d\left(x_i,x_j\right)^2}{2\sigma^2}}, \quad i,j=1,2,\cdots,|\Omega| \tag{9}$$

where σ means the standard deviation, and setting σ=1 enables the value of the influence function to be in an appropriate range, d indicates the Euclidean distance of individuals x_i and x_j in the objective space, as well as the size of the Pareto set is $|\Omega|$.

The density of an individual is the sum of the influence function values of all individuals in the Pareto set for the individual. If $\boldsymbol{\Omega}$ denotes the Pareto set and the size of the Pareto set is $|\Omega|$, then the density value of the individual y is $D(y)$ [10], as given in (10).

$$D(y)=\sum_{i=1}^{|\Omega|}\psi\left(d\left(x_i,y\right)\right) \tag{10}$$

To get the information entropy, the objective space should be normalized and divided into several grids. If the number of objects is three, then the space can be divided into $a_1\times a_2\times a_3$ grids. If there are $|\Omega_{ijk}|$ non-dominated solutions in the (i, j, k) grid space, the density function [10] (12) and information entropy [10] (13) are calculated as below:

$$D_{ijk}=\sum_{y=1}^{|\Omega_{ijk}|}D(y) \tag{11}$$

$$\rho_{ijk}=\frac{D_{ijk}}{\sum_{z_1=1}^{a_1}\sum_{z_2=1}^{a_2}\sum_{z_3=1}^{a_3}D_{z_1z_2z_3}} \tag{12}$$

$$H=-\sum_{z_1=1}^{a_1}\sum_{z_2=1}^{a_2}\sum_{z_3=1}^{a_3}\rho_{z_1z_2z_3}In\left(\rho_{z_1z_2z_3}\right) \tag{13}$$

The better the distributivity of the Pareto set, the larger the information entropy. Since the three optimization objectives are in the same priority, a_1, a_2, and a_3 should be set to the same value. Moreover, too large or too small values of a_1, a_2, and a_3 are not conducive to measuring the distributivity of the Pareto set. In our work, a_i=4, i=1,2,3, is set to complete the grid division according to the distributivity of the Pareto set.

D. Hybrid evolutionary mechanism

In this paper, setting the population size Num=200, the maximum generation G_{max}=100, the crossover probability P_c=0.95, and the mutation probability P_m=0.3 according to simulations, we can get a better Pareto set in a short time. MOHEIEA contains two evolutionary methods to adapt to population characteristics. On the basis of Multi-Objective Evolutionary Algorithm based on Decomposition (MOEA/D), the evolution focuses on enhancing the convergence of populations, while the evolution based on NSGA-II aims at improving the distributivity of populations.

1) Evolution Based on NSGA-II

The encoding conversion mechanism is added to this evolutionary method before and after the evolution to enhance the optimal seeking ability. The evolution based on NSGA-II first obtains the mating pool by tournament selection and selects the next-generation population based on non-dominated sorting and crowding degree after crossover and mutation operations. The non-dominated sorting operation can promote the selection of individuals with superior objective values, while the calculation of the crowding degree facilitates the choice of individuals with better distributivity. This evolution method not only improves the distributivity of the Pareto set, but also preserves the better feasible solution in the next-generation population to reinforce the optimization capabilities.

2) Evolution Based on MOEA/D

This evolutionary method adds the encoding conversion mechanism before and after evolution. Through the differential evolution formula in (14), individuals r_1 and r_2 in the neighborhood and individual p in the Pareto set are randomly selected for evolution. After the multipoint mutation, the original individual r is replaced by the Chebyshev decomposition.

$$offspring = r + F_1 \cdot (r_1 - r_2) + F_2 \cdot (p - r) \tag{14}$$

where (r_1−r_2) represents the perturbation to generate a random evolutionary direction that can prompt the population to jump out of the local optimum and avoid evolutionary retardation. (p−r) can provide a better evolutionary direction for individual r. To prompt the population to evolve in a superior direction, F_2>F_1 should be ensured, where F_1=0.3, F_2=0.6 are taken as the best according to the results. The evolution based on MOEA/D can search for better feasible solutions in the neighborhood, enhancing the convergence of the Pareto set.

3) The combination of evolutionary methods

In summary, the two evolutionary methods are combined in the evolution process through information entropy. The Pareto set of the current population has poor distributivity in the objective space when the obtained information entropy is less than 0.4. Therefore, the evolution based on NSGA-II is used to improve the distributivity of the Pareto set when the value of information entropy is less than the threshold τ (τ = 0.4). Otherwise, the evolution based on MOEA/D is applied to promote the convergence of the Pareto set.

E. Perturbation modification mechanism

The number of controllers is more than half of the number of nodes in illegal individuals, which have low practicality and are likely to have negative impact on the evolution of the population. Therefore, after each generation of evolution, the illegal individuals in the population are modified according to the procedure shown in **Mechanism**.

Mechanism: Perturbation modification mechanism

Input: $|N|$: number of nodes, x: illegal individual
Output: x': legal individual
/* Compute the coefficient ς of the illegal individual x. */
Compute ς = 1/sum(x);
/* Generate perturbation vector. */
v_p = rand(1, $|N|$);
for i = 1 to $|N|$ **do**
    /* Correct the i_{th} gene locus of the perturbation vector. */
    $v_p(i) = v_p(i) > \varsigma$? -1 : 0;
    /* Modify the i_{th} gene locus of the illegal individual. */
    $x'(i) = v_p(i) + x(i)$;
    $x'(i) = (x'(i) == -1)$? 0 : $x'(i)$;
end for
Get the modified legal individual x'.

The rules for modifying v_p are presented by Mechanism. Since each gene locus of v_p obeys a uniform distribution of (0,1), then the distribution rule of the number of -1 is in (15).

$$P(k) = C_k^{|N|} \cdot (1-\varsigma)^k \cdot (\varsigma)^{|N|-k}, \quad k = 0,1,2,\cdots,|N| \tag{15}$$

It can be seen from (14) that if the number of -1 in v_p conforms to the Bernoulli distribution of degree $|N|$, then its mathematical expectation is $E=|N|\cdot(1-\varsigma)$. The increase in the number of 1 in the illegal individual leads to the reduction of its coefficient ς. Therefore, the number of -1 in its perturbation vector is increased, and the number of 1 in the individual in the correcting process is further reduced. Therefore, the mechanism can modify the illegal individual into a better ordinary individual according to the features of the illegal individual.

F. Algorithm procedure of MOHEIEA

The MOHEIEA generates an initial population with a balance of diversity and optimization objectives by the two initialization methods. Various mechanisms such as encoding conversion, information entropy awareness, hybrid evolution, and perturbation modification are added to MOHEIEA to improve the performance of solutions.

1) Time-complexity analysis

The time complexity of MOHEIEA is denoted as $O(|N|\cdot G_{max}\cdot Num)$ that dominated by $|N|$ and evolutionary process, in which G_{max} is the maximum generation and Num is the population size.

2) Space-complexity analysis

In model (1-7), at most $|N|/2$ controllers are supposed to deploy in the network if the number of deployed controllers is denoted *as* m. When $1 \le m \le |N|/2$, the objective space consists of $\sum_{m=1}^{|N|/2} \binom{m}{|N|/2} \approx 2^{|N|-1}$ distinct solutions, which is a large number even for small networks. For example, for Internet2 OS3E topology [3] with 34 network nodes, the number of solutions is $2^{33} \approx 8.59 \times 10^9$. However, MOHEIEA can obtain a superior set of CDSs through population evolution. The space complexity of MOHEIEA is denoted as $O(|N|\cdot Num)$ that dominated by $|N|$ and the population.

Algorithm: MOHEIEA

Input: ***G***: the network topology; *Num*: population size; P_c: crossover probability; P_m: mutation probability; G_{max}: maximum generation; λ: weight vectors set; δ: neighborhood size

Output: $\boldsymbol{\Omega_{fin}}$: The final Pareto set

{Step I: Initialization}

/* Generate initial population. */
P_1 with size *Num*/2 is generated by cluster initialization.
P_2 with size *Num*/2 is generated by uniform initialization.
The combined populations P_1 and P_2 are the initial populations.

{Step II: Evolution}

for $i = 1$ **to** G_{max} **do**
  /* Information entropy awareness mechanism. */
  Calculate the information entropy of the current Pareto set;
  /* The encoding conversion mechanism. */
  The coding method of all individuals within the population is converted to Gray-coding.
  /* Hybrid evolutionary mechanism. */
  if information entropy > τ
    Perform evolution based on MOEA/D;
  else
    Perform evolution based on NSGA-II;
  end if
  /* The encoding conversion mechanism. */
  The coding method of all individuals within the population is converted to binary coding.
  /* Perturbation modification mechanism. */
  Modify all illegal individuals in the current population into legal individuals by **Mechanism**.
end for

{Step III: Stopping Evolution}

Output $\boldsymbol{\Omega_{fin}}$;

IV. Experimental Study

This section evaluates the performance of MOHEIEA and compares it with MOEA/D, NSGA-II, and algorithms in [7]. All algorithms were implemented in MATLAB and run on an Intel Core i5 (3.30 GHz) with 8 GB of memory. In the experiments, Internet2 OS3E were chosen, which are widely used in current studies [3][4]. For simulation purposes, nodes with unclear locations in the network were removed. The distance between two nodes was calculated by Haversine's formula [7].

A. Performance metrics of Pareto set

Since this multi-objective model focuses on a practical problem and the reference set is not available, the performance metrics without a reference set were chosen, which include Hypervolume, Spacing, and Spacing metric [10].

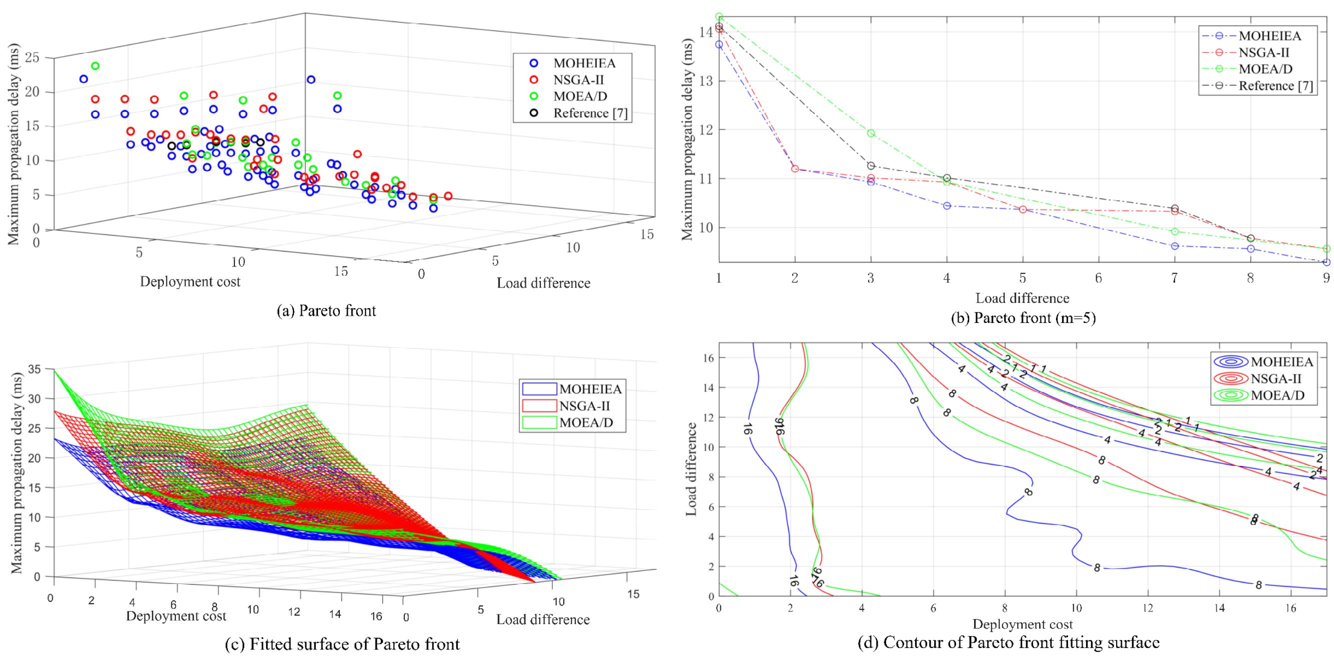
1) Hypervolume: It is used to measure the hypervolume enclosed by the Pareto front and the reference points. The larger its value, the better the comprehensive performance of the algorithm.

2) Spacing: It indicates the standard deviation of the minimum distance from each Pareto solution to other Pareto solutions in the Pareto set. The smaller the spacing value, the better the uniformity of the Pareto set.

3) Spacing metric: It measures the distributivity of the Pareto set. The smaller the spacing metric value, the better the distributivity of the Pareto set.

B. Presentation of the Pareto set

The Pareto sets of MOHEIEA and other comparative algorithms are calculated, compared, and shown by various measures in this subsection. The comparative algorithms in this paper are the algorithm in [7], NSGA-II and MOEA/D. The algorithm in [7] is based on the POCO framework, which facilitates finding the Pareto set for the bi-objective CPPs models. NSGA-II and MOEA/D are two multi-objective evolutionary algorithms that are mostly used for solving MOO problems due to their excellent solution performance. MOHEIEA builds on NSGA-II and MOEA/D to evolve populations for features of the current Pareto set and thus to better favour population evolution.



(a) Pareto front
(b) Pareto front (m=5)
(c) Fitted surface of Pareto front
(d) Contour of Pareto front fitting surface

Fig. 4. Pareto sets of MOHEIEA and other algorithms (Internet2 OS3E)

Fig. 4 shows the Pareto sets solved by MOHEIEA and other algorithms in the Internet2 OS3E network, which has 34 nodes and 42 links. Fig. 4(a) shows that the Pareto front solved by MOHEIEA is closer to the ideal points, and MOHEIEA can obtain a solution with lower propagation delay than other algorithms when the deployment cost and load difference are the same.

According to Fig. 4(b), the distributivity and convergence of the Pareto set obtained by MOHEIEA are better than those by other algorithms when the deployment cost is determined (m=5).To show the results more intuitively, Fig. 4(c) presents the fitted surface of the Pareto set. Compared with the fitted surfaces of NSGA-II and MOEA/D, the fitted surface of MOHEIEA is closer to the ideal point, indicating that a Pareto set with better convergence can be obtained by MOHEIEA.

To increase the credibility of the results, the contours of the fitted surface are shown in Fig. 4(d). We can draw the conclusion that the contours of MOHEIEA approximate the contours of other algorithms when the maximum propagation delay is short. However, the contours of MOHEIEA are better than the contours of other algorithms when the maximum propagation delay becomes long. The result indicates that MOHEIEA can significantly reduce the propagation delay of the control network at the same deployment cost and load difference.

MOHEIEA avoids falling into local optimums while searching for optimal feasible solutions near the current Pareto solution through such improvements as encoding conversion, information entropy awareness, hybrid evolution, and perturbation modification. As a result, MOHEIEA has a strong global and local optimality-seeking ability.

To quantitatively compare the performance of the Pareto sets obtained by MOHEIEA and the performance of those by other algorithms, three performance metrics of the Pareto sets in Fig. 4(a) and Fig. 4(b) are listed in Tables II and III.

Table II proves that MOHEIEA can reach the optimum in all three metrics, which indicates that the Pareto front obtained by MOHEIEA achieves excellent results in uniformity, distributivity and convergence under the three objectives. As shown in Table III, MOHEIEA is not optimal in all three metrics, but it makes a good tradeoff between the three metrics and still achieves optimal results by combining all optimization metrics. This indicates that MOHEIEA can find the Pareto set with superior performance under two objectives.

TABLE II. PERFORMANCE INDICATORS OF PARETO SET IN FIG. 4(A)

Algorithm	Performance Indicators		
	Hypervolume	*Spacing*	*Spacing metric*
MOHEIEA	**0.8603**	**0.0659**	**0.6466**
MOEA/D	0.7260	0.0796	0.6863
NSGA-II	0.6688	0.0771	0.6922

TABLE III. PERFORMANCE INDICATORS OF PARETO SET IN FIG. 4(B)

Algorithm	Performance Indicators		
	Hypervolume	*Spacing*	*Spacing metric*
MOHEIEA	**0.6711**	0.0663	**0.2938**
MOEA/D	0.5566	0.0875	0.8240
NSGA-II	0.6402	**0.0567**	0.6186
Reference [7]	0.6507	0.0753	0.4246

According to the simulation results of Internet2 OS3E network, it can be found that MOHEIEA can obtain the Pareto set with better convergence and distribution than MOEA/D and NSGA-II. Firstly, MOHEIEA improves the algorithm's optimization-seeking ability when solving the Hamming cliff problem through the encoding conversion mechanism. Secondly, the information entropy awareness mechanism is used to measure the distributivity of the current Pareto set and select a more appropriate evolutionary method. The hybrid evolutionary mechanism can select the most suitable evolution according to the characteristics of the current Pareto set, which has more advantages than MOEA/D and NSGA-II. The perturbation modification mechanism drives the population to evolve toward a lower deployment cost, which facilitates the applicability of the CDSs. By combining the above factors, MOHEIEA can obtain the set of CDSs with better convergence and distribution in network.

V. CONCLUSION

The proposed MOHEIEA obtains excellent solution performance through improved mechanisms such as encoding conversion and can be used to solve CPPs. Compared with other algorithms, MOHEIEA has better generality and global search capability of finding Pareto sets with superior distributivity, convergence, and uniformity.

ACKNOWLEDGMENT

This work is supported by National Key Research and Development Program of China (2019YFB1803605, 2020YFB1805601) and National Natural Science Foundation of China (61572391).

REFERENCES

[1] J. Bukhari and W. Yoon, "Design of scalable SDN based eMBMS/WLAN network architecture assisted by Fog Computing," *2018 International Conference on Information and Communication Technology Convergence (ICTC)*, Jeju, 2018, pp. 216-218.

[2] Y. P. Llerena and P. R. L. Gondim, "SDN-Controller Placement for D2D Communications," *in IEEE Access*, vol. 7, pp. 169745-169761, 2019.

[3] Vahid Ahmadi, Mostafa Khorramizadeh, "An adaptive heuristic for multi-objective controller placement in software-defined networks," Computers & Electrical Engineering, vol. 66, pp. 204-228, 2018.

[4] L. Liao and V. C. M. Leung, "Genetic algorithms with particle swarm optimization based mutation for distributed controller placement in SDNs," *2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, Berlin, 2017, pp. 1-6.

[5] J. Sanner, Y. Hadjadj-Aoul, M. Ouzzif and G. Rubino, "An evolutionary controllers' placement algorithm for reliable SDN networks," *2017 13th International Conference on Network and Service Management (CNSM)*, Tokyo, 2017, pp. 1-6.

[6] D. Hock, S. Gebert, M. Hartmann, T. Zinner and P. Tran-Gia, "POCO-framework for Pareto-optimal resilient controller placement in SDN-based core networks," *2014 IEEE Network Operations and Management Symposium (NOMS)*, 2014, pp. 1-2.

[7] M. Samir, E. Samir, M. Azab, M. R. M. Rizk and N. Sadek, "Towards Optimal Placement of Controllers in SDN-enabled Smart Grid," *2018 IEEE 9th Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, Vancouver,BC, 2018, pp.95-100.

[8] A. Jalili, V. Ahmadi, M. Keshtgari and M. Kazemi, "Controller placement in software-defined WAN using multi objective genetic algorithm," *2015 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI)*, Tehran, 2015, pp. 656-662.

[9] G. Wang, Y. Zhao, J. Huang and W. Wang, "The Controller Placement Problem in Software Defined Networking: A Survey," in *IEEE Network*, vol. 31, no. 5, pp. 21-27, 2017.

[10] Zheng Jinhua and Zou Juan, Multi-objective Evolutionary Optimization. Beijing China, CA: Science Publishing House, 2017.