A quantum controller deployment algorithm based on
improved NSGA2 algorithm

Run Zhe Lin, Liang Rui Tang, Fan Bing
North China electric power university, Beijing, 102206, China
1094271020@qq.com

Abstract. In order to be able to apply quantum encryption communication to large-scale networks, the adoption of SDN networking in quantum key distribution networks is considered to be a feasible solution. This paper discusses the optimization scheme of deploying quantum encryption controller in QKD under various optimization objectives, and proposes a controller deployment algorithm based on improved NSGA2 algorithm according to network topology characteristics, which is intended to guarantee the algorithm. At the same time of effectiveness, the convergence speed of the algorithm is improved.

1. Introduction
With the accelerated development of the Internet, network information security has become increasingly important. The emergence of quantum computers has greatly challenged the confidentiality of the classic public keys that are currently used. With the continuous improvement of quantum secure communication technology, how to apply quantum encryption technology to large-scale communication networks, to ensure the communication security of the network has become an important research direction. In order to meet the network control requirements of large-scale quantum key distribution networks, relevant researches have proposed to refer to the networking mode of SDN in traditional networks in the quantum key distribution network to improve the network operation efficiency. Literature [1] gives a specific scheme of how to use SDN for networking in QKD and defines it as SQN. Like the data forwarding controller in SDN, the deployment of quantum encryption controllers in SQN plays a key role in the QKD network. Optimizing its deployment location is a noteworthy direction.

At present, there are few researches on the optimization of the deployment location of quantum encryption controller, so it is necessary to learn from the traditional SDN network and put forward a plan for the deployment of SQN controller based on the characteristics of quantum encryption. In the traditional SDN network, there are some related literatures to study the deployment principle of SDN controller. The deployment plan of SDN controller takes time delay as the primary consideration. Literature [2] uses game theory to discuss how to balance the interaction time delay between SDN controllers and the time delay between SDN controller and the node it controls to obtain the integrated optimal deployment plan of SDN controller. In addition to time delay, controller load is also an important factor. Literature [3] optimizes the load balance of SDN controller and reduces the number of controllers needed. The multi-controller deployment problem proposed in the above literature is a NP problem, and the solution algorithm is mostly based on BIP model. Although this algorithm can achieve good results, it is only applicable to small networks. In the face of large networks, it is difficult to solve the problem of too long running time.

Therefore, this paper proposes a heuristic algorithm based on genetic algorithm for multi-objective
controller deployment. On the one hand, the algorithm considers both the delay and the load balancing of the quantum cryptographic controller and uses the NSGA2 genetic algorithm to find the optimal quantum controller deployment scheme considering the two objectives. On the other hand, in order to speed up the convergence of the algorithm, this paper improves the variation process of genetic algorithm for the characteristics of SQN network. Using the topological characteristics of the network, the cross-variation of the two chromosomes in the traditional genetic algorithm is changed from numerical variation to chromosome cross-variation according to the position of the node represented by the chromosome in the network. The improved algorithm improves the stability of the excellent solution, and improves the convergence speed of the genetic algorithm in the SQN network environment. The algorithm is intended to ensure that the algorithm can find the location of an excellent quantum cryptographic controller and improve the speed of the algorithm.

2. Deployment scheme of quantum encryption controller

2.1. Algorithm model

As mentioned above, the load balancing of quantum encryption controller and the delay of quantum encryption controller network are considered when quantum encryption controller is deployed. This problem is described by BIP model:

Firstly, the network $G (V, E)$ is given. $V$ represents the nodes in the network, $E$ represents the edges of the network, and $N$ represents the number of controllers in the SQN network.

The symbols that appear in the algorithm are shown in the following table:

| symbol | Semantics |
|--------|-----------|
| $|V|$ | Total number of network nodes |
| $N$ | Total number of network controllers |
| $x_{ij}$ | Indicates whether node $j$ is assigned to a binary variable of a quantum encryption controller deployed on node $i$ |
| $y_i$ | Indicates whether the quantum encryption controller is deployed on a binary variable on node $i$ |
| $d_{ij}$ | Time delay from node $j$ to quantum encryption controller $i$ |
| $\sigma$ | The number of controls per quantum encryption controller can exceed the threshold |
| $\varepsilon$ | Increased network overhead per hop delay |

The performance indicator $FIT_1$ calculates the variance of the sum of the delays of each quantum encryption controller to all the nodes it controls, and uses this to control the average delay of each quantum encryption controller to its node:

$$FIT_1: \quad r = \frac{1}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} x_{ij} y_i d_{ij}$$

The performance indicator $FIT_2$ calculates the variance of the sum of the delays of each of the quantum encryption controllers to the nodes it controls, and uses this to express the degree of equalization of each quantum encryption controller load:

$$FIT_2: d_{\text{var}} = \sum_{i=1}^{M} y_i \left( \frac{e}{N} \sum_{i=1}^{N} x_{ij} y_i - \frac{e}{N} \sum_{i=1}^{N} \sum_{j=1}^{M} x_{ij} y_i y_i \right)^2$$

The algorithm optimizes the target performance index $FIT_1$ and the performance index $FIT_2$ at the same time, so that the final quantum controller deployment scheme satisfies min $FIT_1$ and min $FIT_2$. The quantum encryption controller deployment in this network meets the following constraints:

- A node is only allowed to access a quantum encryption controller:

$$\sum_{i=1}^{M} x_{ij} = 1 \quad \forall j \in V$$
There are only $N$ quantum encryption controllers in the network:

$$\sum_{i=1}^{V} y_i = N$$  \hspace{1cm} (4)

Each node can only connect to a node that is set to a quantum encryption controller:

$$x_{ij} \leq y_j \quad \forall i, j \in V$$  \hspace{1cm} (5)

In order to limit the maximum control delay of the quantum encryptiongraphic controller, the maximum delay of each quantum encryptiongraphic controller to its control node is:

$$d_{max} = \max_{y_i, j} \{ x_{ij}d_{ij} \}$$  \hspace{1cm} (6)

The delay of each quantum encryption controller to its control node meets the following conditions:

$$x_{ij}d_{ij} \leq d_{max} \quad \forall i, j \in V$$  \hspace{1cm} (7)

The number of controllers controlled by each quantum encryption controller cannot exceed the upper limit of the number of controls:

$$\sum_{i=1}^{V} x_{ij} \leq \frac{V}{N} + \sigma_j$$  \hspace{1cm} (8)

The multi-objective BIP model described above can be solved by a general optimizer such as Gurobi and CPLEX. However, from the perspective of efficiency of the solution, the above method is only used for small networks. For large-scale networks, we use a heuristic algorithm with low time complexity to solve. How to use this heuristic algorithm will be explained in the next section.

### 2.2. Heuristic algorithm based on genetic algorithm NSGA2

The heuristic algorithm in this paper is called multi-objective quantum encryption controller deployment algorithm (MO-QECPA). In this algorithm, node index coding is adopted, and chromosomes correspond to the deployment location of quantum encryption controller. After determining the deployment location of quantum encryption controller, Greco is adopted to solve the problem of node allocation in the network[4]. The fitness set of chromosomes is $\text{Fits}=\{\text{FIT}_1, \text{FIT}_2, \text{FIT}_3\}$. In order to calculate the fitness of chromosomes, Greco distribution nodes are similarly used in the fitness function.

The main process of MO-QECPA is described in algorithm 1. First, the node is encoded using its index (line 2). An initial population containing all genes is then generated (line 3). For example, when $|V|=10$, the initial population should include all indexes of the node (1,2,3,..., 10). For each member of the population, nodes are assigned according to Greco(line 4) and their fitness set is calculated. Circulation (6-21) describes the genetic process of NSGA2 [5]. If there is no replacement best solution after $|V|*N2$ iterations, end the loop.

| Algorithm 1 MO-QECPA |
|---|
| **Input:** $G(V,E)$ |
| **Output:** Placement solution |
| 1: **procedure** MO-QECPA |
| 2: **Encode** nodes by their indices |
| 3: **Initialize** $P_{t} \leftarrow$ population. $Q_{t} \leftarrow []$ |
| 4: $\text{Fits}_{t} \leftarrow$ Fitness($P_{t}$) |
| 5: $m \leftarrow 0$ |
| 6: **while** $m \leq |V| * N2$ do: |
| 7: \hspace{0.5cm} $R_{t} \leftarrow Q_{t} \cup P_{t}$ |
| 8: \hspace{0.5cm} $F_{t} \leftarrow \text{fast_nondominate_sort}( \text{Rt} )$ |
| 9: \hspace{0.5cm} $P_{t+1} \leftarrow []$ |
| 10: \hspace{0.5cm} $i \leftarrow 0$ |
11:   crowding_distance_sort($F$) 
12:   while len($P_{t+1}$) + len($F[i]$) < $N$ do 
13:      $P_{t+1} = P_{t+1} \cup F[i]$ 
14:      $i \leftarrow i+1$ 
15:   $Fits_{p_{t+1}} \leftarrow \text{Fitness}(P_{t+1})$ 
16:   $Q_t = \text{Make_new_generation}(P_{t+1}, m)$ 
17:   $Fits_{Q_t} \leftarrow \text{Fitness}(Q_t)$ 
18:   $m \leftarrow m+1$ 
19:   $P_t \leftarrow P_{t+1}$ 
20: Get the best member from the $P_t$ 
21: end procedure

Since the index of the nodes in the network does not directly reflect the position of the node in the network and the relationship with other nodes, the quantum controller deployment position represented by the child chromosome generated in the classic way (line 16) may appear at any position, and cannot preserve the excellent characteristics of the parent chromosome, it expands the scope of the search and reduces the efficiency of the algorithm.

Similar to the traditional numerical crossover, the chromosome crossover algorithm in this paper believes that in a finite network, such a solution should appear in the path of the node with the two nodes indexed as the source destination. This is the classical algorithm in a continuous numerical interval. The way to find a better feasible solution is mapped in a network where there is a fixed connection between nodes. At the same time, the shortest path is used to represent the path between two nodes, which is a classic algorithm that maps values continuously in the network. Through these two mappings, the crossover algorithm in this paper is more suitable for the network environment.

Algorithm 2 describes how to add a new chromosome crossover algorithm designed in this paper to the NSGA2 algorithm. First, from the binary tournament algorithm from $P_{t+1}$ from the selection of excellent chromosomes (line 3), then the cycle (line 6-20) according to the selected excellent chromosomes for crossover and mutation: first randomly select two of the selected excellent chromosomes. The chromosome is then judged whether or not the chromosome is crossed (the chromosome crossover probability is $rate_1$), and the crossover operation is performed at the first position of the chromosome (the probability of crossing a certain position of the chromosome is $rate_2$). After determining the position at which the intersection is made, the Dijkstra algorithm is used to calculate the shortest path between the nodes represented by the genes at the two locations, and the shortest path contains the set of links as path. Nodes other than the source destination node are then randomly selected from the set path as a result of the intersection. When the random selection is specifically performed herein, the probability that the node closer to the source destination node is selected is higher. If a chromosome does not cross-over with other chromosomes, it is judged whether or not the mutation operation is performed (the probability of chromosome mutation is $rate_3$). Then, if the number of iterations reaches half of the upper limit, the chromosomal mutation probability is increased by a factor of two. The variation in this paper is the variation in classical genetic algorithms.

Table 3 Make_new_generation algorithm

Algorithm 2 make_new_generation

Input: $P_{t+1}$
Outpu: $Q_t$
1: procedure make_new_generation
2: Initialize $Q_t \leftarrow []$, $C_t \leftarrow []$, $rate_1$, $rate_2$, $rate_3$, $D$
3: $C_t \leftarrow \text{binary_tournament}(P_{t+1})$
4: $i \leftarrow 0$, $j \leftarrow 0$
5: $J \leftarrow \text{Size}(C_t)$
6: while $i \leq J$ do
7: member_1, member_2 $\leftarrow \text{random_selection}(C_t)$
8: if random_number(0,1) $<$ $rate_1$
9: end if
5: while \( j \leq \text{Size}(\text{member}_1) \)
6: \( Q[t][j] \leftarrow \text{member}_1 \)
7: if random_number(0,1) < rate_3
8: \( \text{path} \leftarrow \text{dijkstra}(\text{member}_1[j], \text{member}_2[j]) \)
9: \( Q[t][j] \leftarrow \text{random_selection} \)(path)
10: if \( m < D/2 \)
11: if random_number(0,1) < rate_2
12: \( Q[t][j] \leftarrow \text{variation}(\text{member}_1) \)
13: elseif
14: if random_number(0,1) < 2*rate_2
15: \( Q[t][j] \leftarrow \text{variation}(\text{member}_1) \)
16: end procedure

3. Simulation analysis
This paper selects 91 node topologies of Qinhuangdao power grid to verify the effectiveness of the algorithm. In the process of network simulation, 10,000 requests, which are completely random in both source and destination nodes, are generated in the network.

3.1. Algorithm convergence rate comparison
Firstly, in the simulation environment of this paper, the improved NSGA2 algorithm and the common NSGA2 algorithm are used to select the optimal deployment location of the quantum encryption controller in 91 nodes of Qinhuangdao Power Grid, and the difference in convergence speed of the algorithm is compared. For comparison purposes, three performance functions are considered during the running of the algorithm, but only the value of the performance function \( F_{IT1} \) during each iteration is compared.

![Figure 1 Comparison of the convergence speed of the algorithm](image)

As shown in the figure, in the case of the same initial population, in the initial iteration, due to the chromosome crossing method of the traditional NSGA2 algorithm, the result of the initial crossover is better than the improved NSGA2 algorithm. However, after the algorithm in this paper determines the excellent chromosomes, it can search for better individuals in the neighboring nodes of these nodes according to the location of the network nodes represented by the chromosomes and preserve the excellent characteristics of the original chromosomes. The number of iterations is about 60 times. A relatively stable and excellent chromosome population has been obtained. At the same time, in order to avoid falling into the local optimum situation, the probability of chromosomal mutation is increased when the number of iterations exceeds half of the set value. Therefore, after the number of iterations is about 100 generations, the optimal value of the average delay of the optimal individual is further updated, but the result obtained from the 60th generation is not much different. However, the traditional NSGA2 algorithm has an iteration number of 80-120 generations, and the optimal solution is still changing, and the optimal solution is not as optimal as the algorithm proposed in this paper. Through simulation, the improved NSGA2 algorithm in this paper can effectively improve the efficiency of convergence in the algorithm and obtain better chromosomes.
3.2. Comparison of controller deployment schemes derived from the algorithm

The difference between the quantum encryption controller deployment scheme based on the algorithm proposed in this paper and the quantum encryption controller deployment scheme derived from the delay BIP model-I and the two BIP models-II of the load are compared. At the same time, in order to prove that the delay obtained by the genetic algorithm is within a reasonable range, the comparison algorithm uses IBM CPLEX to solve the BIP model.

The comparison result of the total delay in the 91-node topology of Qinhuangdao Power Grid is shown in Fig. 2(a), and the comparison result of the equalization result of the quantum encryption controller receiving request is shown in Fig. 2(b). For the sake of comparison, the results are normalized:

Figure 2 shows the results of the algorithm comparison: (a) quantum encryption controller average delay and (b) load balancing coefficient comparison chart

The result shows quantum encryption controller deployment scheme obtained by using the BIP algorithm for delay is smaller in the total delay of response to 10000 encryption requests than the total delay of the deployment scheme obtained by the algorithm herein. When the controller deployment scheme only considers the time delay, the BIP can get the optimal result on the response delay. However, as can be seen from Figure 2(a), the difference between the two schemes in the delay results is small, with a maximum of 7%. At the same time, it can be seen from Fig. 2(b) that the algorithm proposed in this paper has a small difference in the balance between the load balance of the quantum controller control and the BIP algorithm for load balancing, only about 10%. The algorithm proposed in this paper sacrifices the performance of a small number of quantum controller deployment schemes, but it can effectively improve the computational speed of the algorithm.

4. Conclusion

This paper proposes a heuristic algorithm for multi-target controller deployment based on the improved NSGA2 genetic algorithm. It is verified by simulation that it can guarantee an excellent quantum controller deployment scheme under the double target condition. At the same time, according to the characteristics of SQN network, the chromosomal variation method is improved, and the convergence speed of the algorithm is improved. This algorithm improves the speed of the algorithm while obtaining similar results compared with the BIP algorithm. At the same time, it also has better performance in large networks.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Grant No. 51677065).
References

[1] W. Yu, B. Zhao and Z. Yan. (2017) Software defined quantum key distribution network, 2017 3rd IEEE International Conference on Computer and Communications (ICCC). Chengdu. pp. 1293-1297.

[2] Ksentini A, Bagaa M, Taleb T, et al. (2016) On using bargaining game for Optimal Placement of SDN controllers. IEEE International Conference on Communications, IEEE. pp. 1-6.

[3] Yao G, Bi J, Li Y, et al. (2014) On the Capacitated Controller Placement Problem in Software Defined Networks. IEEE Communications Letters. 18(8): 1339-1342.

[4] Hu Y, Luo T, Beaulieu N C, et al. (2017) The Energy-Aware Controller Placement Problem in Software Defined Networks. IEEE Communications Letters. 21(4): 741-744.

[5] P. Xiaoying, Z. Jing, C. Hao, C. Xuejing and H. Kaikai. (2015) A differential evolution-based hybrid NSGA-II for multi-objective optimization. 2015 IEEE 7th International Conference on Cybernetics and Intelligent Systems (CIS) and IEEE Conference on Robotics, Automation and Mechatronics (RAM). Siem Reap. pp. 81-86.