

A Cooperative Evolution of Multiple Operators Based Adaptive Quantum Genetic Algorithm for Network Coding Resources Optimization

Hongbo Xu, Shasha Wang, Zhijian Qu*

School of Computer Science and Technology, Shandong University of Technology, Zibo, China Email: axuhongbo@126.com, 1981777829@qq.com, *zhijianqu@sdut.edu.cn

How to cite this paper: Xu, H.B., Wang, S.S. and Qu, Z.J. (2019) A Cooperative Evolution of Multiple Operators Based Adaptive Quantum Genetic Algorithm for Network Coding Resources Optimization. *Journal of Computer and Communications*, **7**, 147-161.

https://doi.org/10.4236/jcc.2019.77014

Received: June 10, 2019 **Accepted:** July 13, 2019 **Published:** July 16, 2019

Copyright © 2019 by author(s) and Scientific Research Publishing Inc. This work is licensed under the Creative Commons Attribution International License (CC BY 4.0). http://creativecommons.org/licenses/by/4.0/

Open Access

Abstract

In order to optimize the network coding resources in a multicast network, an improved adaptive quantum genetic algorithm (AM-QEA) was proposed. Firstly, the optimization problem was translated into a graph decomposition problem. Then the graph decomposition problem was represented by the binary coding, which can be processed by quantum genetic algorithm. At last, a multiple-operators based adaptive quantum genetic algorithm was proposed to optimize the network coding resources. In the algorithm, the individual fitness evaluation operator and population mutation adjustment operator were employed to solve the shortcomings of common quantum genetic algorithm, such as high convergence rate, easy to fall into local optimal solution and low diversity of the population in later stage. The experimental results under various topologies show that the proposed algorithm has the advantages of high multicast success rate, fast convergence speed and strong global search ability in resolving the network coding resource optimization problems.

Keywords

Network Coding, Quantum Genetic Algorithm, Multicast Networks, Optimization

1. Introduction

Multicast allows the information transmitted from single point to multiple points in networks. Compared with the traditional unicast and broadcast methods, multicast dramatically improves transmission efficiency. Multicast technology can provide critical technical support for most services, such as distance learning, video on demand, data distribution in the data center and etc. [1]. Network coding is an effective technique for improving network throughput, and it also can bring many benefits for the multicast transmission. It allows intermediate network nodes to perform arbitrary mathematical calculations to combine (encode) the input packets, and then output the encoded packets to the downstream nodes [2]. The key problem network coding based multicast technology is how to quickly and efficiently determine the coding nodes of each route in the network. Therefore, the point-to-multipoint multicast rate can reach the upper limit specified by Shannon's "maximum stream-minimum cut" theorem [3]. As an emerging technology, network coding technology has received much attention. However, this problem belongs to the NP-hard problem, and there is currently no clear and efficient solution.

The research on network coding technology initially focused on creating different coding constructs. For example, a polynomial time algorithm for network coding was proposed by P. Sander *et al.* [4], an algebraic construction method for network coding was given by R. Koetter and M. Medard of MIT [5] [6]. Recently, research on network coding technology has become more diverse. For example, Sudipta Sengupta *et al.* [7] studied coding in wireless networks. In order to solve the Network Coding problem for Critical Infrastructure Networks, Rakesh Kumar *et al.* [8] proposed an architecture that realizes linear NC by decomposing the linear NC functions. Imad El Qachchach *et al.* [9] implemented an efficient multi-source network coding by using low-rank parity check code.

In 1996, A. Narayanan and M. Moore first combined the quantum computing theory with evolutionary algorithms and proposed the concept of the quantum genetic algorithm [10]. The quantum genetic algorithm has the advantages of small population size, maintaining population diversity, and being easy to parallelize. Compared with traditional genetic algorithms, it can solve the shortcomings of the traditional genetic algorithm such as sensitivity to population size, more iterations, easy falling into the local optimum solution, and slow convergence speed.

Quantum genetic algorithms have a wide range of applicability. In recent years, it has been the focus of research on solving optimization problems. For example, Lijun Mao *et al.* [11] applied the quantum genetic algorithm to the cloud computing job scheduling. Ye Zhang *et al.* [12] used it to solve the observing and downloading integrated scheduling problem of earth observation satellite. Kong Haipeng *et al.* [13] presented an adaptive double chain quantum genetic algorithm (ADCQGA) for solving constrained optimization problems. Kim *et al.* [14] first introduced genetic algorithm to solve the network coding resources optimization problem in a multicast network. However, due to the instability of genetic algorithm, the effect is general. Based on Kim's method, Qu *et al.* [15] [16] proposed a genetic algorithm to optimize network coding resources with transmission restriction in a multicast network.

The key problems of the quantum genetic algorithm are to determine the quantum bit coding, the quantum revolving gate mechanism, and the population

fitness evaluation function. For each individual, the evolutionary parameters should be considered carefully because the adjustment of the rotation angle is flexible. If the rotation angle is too small, the convergence speed of the algorithm is slow, and the rotation angle is too large, which makes the algorithm easy to fall into the local optimal solution.

To optimize the network coding resource in a multicast network, an adaptive quantum genetic algorithm based on the multi-operator coordination mechanism was presented. The algorithm can solve the problem of easy to fall into the optimal local solution by improving the diversity of the population in the later evolution stage.

2. Problem Formulation

In order to facilitate the research, this paper simplifies the network coding resources problem to be studied as follows: assume that the communication network is represented by a directed graph G(V, E) with all sides being unit flows. The edge with capacity K can be replaced with the side of the K unit flow in this graph. Then the single source multicast problem can be represented by a quaternion (G, s, T, R), where G represents a directed topology, s is a multicast source node, $T = \{t_1, t_2, \dots, t_n\}$ is the set of destination nodes, and R is the multicast rate of node s to all target nodes. We combine the coded nodes linearly. A coding strategy that the multicast rate from s to all target nodes is R called a feasible linear network coding. For a particular (G, s, T, R), we hope that the last established multicast tree has as little coding as possible while achieving the given multicast rate R.

We use the number of encoded edges to measure the task quantities of the encoding operation. As shown in **Figure 1**, a coding node v has three input edges and two exit edges, where $y_1 = x_1 + x_2$, $y_2 = x_1 + x_2 + x_3$. If at least two input edges need to output information from an output edge simultaneously, then this output edge is defined as the coded edge. Therefore, the node has two coding edges and thus has two encoding operations. For a particular node in (*G*, *s*, *T*, *R*), it can be considered a potential coding node if and only if its number of input edges $m \ge 2$ and the exit edge $n \ge 1$.

In order to facilitate the design of the fitness function, we split all potential coding nodes. Assuming that an encoding node has *m* input edges and *n* output



Figure 1. Coding node v.

edges, we introduce *p* auxiliary nodes $\{u_1, u_2, \dots, u_p\}$ to connect with the nodes of the input links, and introduce *q* auxiliary nodes $\{w_1, w_2, \dots, w_q\}$ is connected with the nodes of the output links. Then all traffic flowing through the node can be represented as a linear combination between the secondary node u_i and the secondary node w_j . The topological maps of each algorithm in this paper are all decomposed topological maps. For example, the topological map after the decomposition of the coding node v in **Figure 1** is as shown in **Figure 2**.

3. Algorithm Description

3.1. Algorithm Flow

The whole algorithm includes three parts: the rotation angle adaptive adjustment mechanism, the multi-operator synergy mutation mechanism, and the population fitness evaluation. The overall algorithm flow can be described, as shown in **Figure 3**.

3.2. Rotation Angle Adaptive Adjustment Mechanism

The method of population update in this algorithm uses a quantum revolving door update strategy. Its update matrix can be expressed as follows.

$$\begin{bmatrix} \alpha_i' \\ \beta_i' \end{bmatrix} = \begin{bmatrix} \cos(\theta_i) & -\sin(\theta_i) \\ \sin(\theta_i) & \cos(\theta_i) \end{bmatrix} \begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix}$$
(1)

In Equation (1), (α_i, β_i) is the *i*-th qubit in the chromosome and θ_i is a rotation angle. The adaptive adjustment mechanism modifies the rotation angle of individual evolution according to the fitness of different individuals. The rotation angle lookup table used in this algorithm is shown in **Table 1**.

In **Table 1**, f(x) represents the fitness value of the individual x, x_i^j represents the *i*-th position of the *j*-th individual, represents the *i*-th position of the optimal individual in the current population, and $S(\alpha_i^j, \beta_i^j)$ represents the rotation angle in the polar coordinates. θ^j is the rotation angle step used by the *j*-th individual. Its definition is as follows.

$$\theta^{j} = \begin{cases} \frac{f_{j} - f_{\min}}{f_{\max} - f_{\min}} \cdot (K_{2} - K_{1}) + K_{1}, & f_{\max} \neq f_{\min} \\ K_{1}, & f_{\max} = f_{\min} \end{cases}$$
(2)



Figure 2. Split coding node v.



Figure 3. The flow chart for the AM-QGA.

In the current population, the *i*-th evolutionary rotation angle step and rotation direction of the *j*th individual can be calculated by the Equation (3).

$$\Delta \theta_i^j = \theta^j \cdot S\left(\alpha_i^j, \beta_i^j\right) \tag{3}$$

3.3. Fitness Function Design

For the convenience of solving problems, we translate the minimization problem

X_i^j	h	$\Delta heta_i^j$ –	$Sig(oldsymbol{lpha}_i^j,oldsymbol{eta}_i^jig)$					
	ν_i		$\alpha_i^j \beta_i^j > 0$	$\alpha_i^j \beta_i^j < 0$	$\alpha_i^j = 0$	$\beta_i^j = 0$		
0	0	$\theta_{I}^{j}=0$	-	_	-	_		
0	0	$\theta_2^j = 0$	_	_	-	_		
0	1	$\theta_3^j = 0$	+1	-1	0	±1		
0	1	$\theta_4^j = 0$	-1	+1	±1	0		
1	0	$\theta_{5}^{j}=0$	-1	+1	±1	0		
1	0	$\theta_6^j = 0$	+1	-1	0	±1		
1	1	$\theta_7^j = 0$	-	_	-	_		
1	1	$\theta_8^j = 0$	-	_	-	_		

Table 1. Rotation angle step lookup table.

into the maximization problem. For a single chromosome *X*, the fitness formula is designed as follows:

$$Fitness(X) = \begin{cases} W - need, f = 1\\ W - n_{\max}, f = 0 \end{cases}$$
(4)

In this equation, n_{max} is the number of all coding edges in the topology map corresponding to chromosome *X*, and *need* is the number of coding edges in the generated multicast tree, and *W* is a constant much larger than n_{max} .

The variable need can be calculated by the following steps. First, we generate the topological map corresponding to chromosome X and then use the topology map as input. Second, we use the dinic algorithm to solve the maximum flow (s, t) of the source point s to all other target nodes $t \in T$. If there is any flow (s, t) less than multicast rate *R*, this indicates that the topology can't meet the condition. That is, if f = 0, then need = n_{max} , otherwise f = 1. For all target nodes $t \in T$, we use the Dijkstra algorithm to solve the path set $P(s,t) = \{P_1(s,t), P_2(s,t), \dots, P_R(s,t)\}$. For each target node, we run R times Dijkstra algorithm. It is necessary to ensure that for every $P_i(s,t)$ path set, for any $i \neq j$, $P_i(s,t) \cap P_i(s,t) = \emptyset$. We record the passing edges in the tag array when we run the Dijkstra algorithm, which ensures that the repeated edges will not be taken during the next iteration of the algorithm. The path set P is all sides of the multicast tree. In this multicast tree, if the number of incident edges of a node is not less than 2, the corresponding output edge of the node is the coding edge. In the process of the Dijkstra algorithm, we could define two sets for all potential coding nodes. In this way, we record the edges emitted from the point, and the edges injected into the point, respectively. The need can be quickly calculated through these two sets. The whole process of the fitness evaluation function is shown in Figure 4.

3.4. Multi-Operator Synergy Mutation Mechanism

The adaptive quantum genetic algorithm can assign different rotation angle steps



Figure 4. Fitness assessment function flow chart.

according to the characteristics of each individual, which greatly improves the convergence speed of the algorithm. However, in the later stage of the algorithm, the algorithm is easy to fall into the optimal local solution due to the decline of population diversity. Based on this, the multi-operator synergy mutation mechanism is added. The current population status is evaluated by multiple operators, which increases the diversity of populations in the late stage of the algorithm. And the algorithm can jump out of the optimal local solution with a large

probability. The stability of the algorithm is enhanced.

The individual similarity evaluation operator X_{sim} is used to evaluate the differences between individuals in the current population. As shown in the Equation (5), where d_{max} and d_{min} respectively represent the largest and smallest individual between the current population and the optimal individual. The d_{avg} is the average value of the Hamming distance between all individuals in the current population.

$$X_{sim} = \begin{cases} \frac{d_{avg} - d_{\min}}{d_{\max} - d_{\min}}, & d_{\max} \neq d_{\min} \\ 0, & d_{\max} = d_{\min} \end{cases}$$
(5)

The larger the similarity X_{sim} , the more the majority of the current population is farther from the optimal solution. In this case, the larger mutation probability can be used. The smaller the similarity X_{sim} , the more the current population distance solution is closer. The individual mutation probability can be reduced to make the algorithm converge faster.

The role of the individual fitness assessment operator is to evaluate the performance of each in the current population. The f_{max} and f_{min} respectively represent the optimal fitness and the worst fitness value of the current population. The f_i indicates the fitness value of the current individual.

$$y_{fit} = \begin{cases} \frac{f_{\max} - f_i}{f_{\max} - f_{\min}}, & f_{\max} \neq f_{\min} \\ 0, & f_{\max} = f_{\min} \end{cases}$$
(6)

The larger the operator y_{fit} , the worse the performance of the current individual. the greater the probability of mutation should be given to the current individual so that it will mutate more quickly. The smaller the operator y_{fit} , the better the performance of the current individual. A smaller probability of mutation should be given to the current individual. It could maintain its excellent traits.

The population variation adjustment operator $F_{acc}(n)$ is a function related to evolutionary algebra to avoid the premature occurrence of the algorithm. When the algorithm falls into the optimal local solution, the operator can increase the mutation probability and make the algorithm jump out of the optimal local solution.

$$F_{acc}(n) = \begin{cases} F_{acc}(n-1) + C \cdot \frac{s-n}{s}, & \text{where } f_{\max}(n) = f_{\max}(n-T) \wedge n > T \\ F_{acc}(n-1), & \text{where } f_{\max}(n) \neq f_{\max}(n-T) \wedge n > T \\ 0, & \text{where } n \le T \end{cases}$$
(7)

The *n* is the current evolutionary algebra, *s* is the maximum evolution algebra, *T* is the algebra of the optimal solution in the iterative process of the algorithm, $C(0 \le C \le 1)$ is the adjustment constant, f_{\max} is the optimal adaptation in the nth generation population degree. It can be known from the definition that when the optimal fitness in the population does not change *T* generation conti-

nuously, the mutation probability will increase correspondingly in the T + 1 generation.

The mutation probability of each individual in each evolution is determined by the individual similarity evaluation operator, the individual fitness evaluation operator, and the population variation adjustment operator. The formula for calculating the mutation probability is shown in Equation (8).

$$p_n^i = \begin{cases} p_0 \cdot y_{fit}^n \cdot x_{sim} + F_{acc}(n), & f_{max} \neq f_{min} \\ 0, & f_{max} = f_{min} \end{cases}$$
(8)

In Equation (8), p_n^i is the mutation probability of the *i*-th individual of the *n*-th generation population, p_0 is the initial mutation probability constant determined according to different problems.

The multi-operator synergistic mutation mechanism mainly includes individual similarity evaluation operator, individual fitness evaluation operator, and population variation adjustment operator. Their combined action is to determine the mutation probability of each in the current population.

4. Simulations and Analysis

In order to verify the effectiveness of the algorithm, this paper compares GA, QGA, and AM-QGA, and carries out the following experiments.

The software and hardware environment used in the experiment is: CPU: Intel(R) Core(TM) i5-6200U CPU4 core 2.30 GHz; memory: 4 G; Windows10; Codeblocks 16.0; GCC 5.4.0.

Firstly, we experimented with a small number of cases, as shown in **Figure 5**. The capacity of each link in the figure is unit capacity. The *s* is the source node, and t_1 , t_2 , t_3 , and t_4 are destination nodes. According to the maximum flow minimum cut theorem, the multicast throughput between *s* and t_1 , t_2 , t_3 , and t_4 is two units of capacity. In this experiment, the multicast throughput is set to 2 units. In **Figure 5**, the potential coding nodes are nodes 7, 8, 9, 10, 12, n_{max} is 9. And the required number of chromosome coding bits is 18. The minimum



Figure 5. Topology-Network.

number of encodings required for **Figure 5** using the exhaustive method is 0. In this experiment, the population size of GA, QGA, and AM-QGA was 30, and the termination algebra was 200.

In the evaluation of algorithm performance, three evaluating indicators are the average optimal algebra, average optimal coding times, and the minimum number of coding times in ten experiments. They show as follows: FBG (first best solution generation), BSR (best solution success ratio), and MOP (minimum operation). The experimental results are shown in **Table 2**.

It can be seen from **Table 2** that the three algorithms in this paper are the same as the results obtained by the violence calculation, so the correctness of the three algorithms can be proved.

In order to ensure the experimental results, this paper generated a larger scale topological map according to the literature [1] strategy. As shown in **Figure 6**, the original topology map is a directed topology map of one source point and two target nodes.

We copy **Figure 6** into two copies and use the target starting point as the source point of the duplicated image to obtain the topological map, as shown in **Figure 7**. We named it 3-copy. Similarly, we get 7-copy, 15-copy, and 31-copy topologies, respectively. The total nodes, the number of edges, the number of target nodes, the number of potential coding nodes, the number of individual chromosome bits, the maximum number of coding operations, and all possible coding operands for these topological maps are shown in **Table 3**.

For the four topological maps in the above table, the same index is used for evaluation. The population size of GA, QGA, and AM-QGA is set to 30, and the ending algebra MAXGEN is 500. The average result of 10 experiments is obtained. The experimental results are shown in **Table 4**.

Algorithm	FBG	BSR	МОР
GA	8.1	100%	0
QGA	4.7	100%	0
AM-QGA	2.2	100%	0

Table 2. Topology-Network algorithm operation result table.



Figure 6. Original network.



Figure 7. 3-copy network

Table 3. Network parameters.

Networks	Nodes	Links	Sinks	Code_Nodes	Bits	Max_Code_Links	Possibilities
3-сору	19	30	4	7	34	15	32768
7-copy	43	70	8	19	98	43	8.79609e+012
15-copy	91	150	16	43	226	99	6.33825e+029
31-copy	187	310	32	91	482	211	3.29101e+63

Table 4. Simulation results.

Multicast scenarios	Algorithms	FBG	BSR	МОР
	GA	10.3	100%	0
3-сору	QGA	6.9	100%	0
	AM-QGA	4.5	100%	0
	GA	293.9	100%	0
7-copy	QGA	96	100%	0
	AM-QGA	15.9	100%	0
	GA	-	0%	36
15-copy	QGA	247.9	90%	0
	AM-QGA	134.1	100%	0
	GA	-	0%	94
31-сору	QGA	343.1	80%	1
	AM-QGA	241.8	100%	1

Figures 8-11 show the relationship between the optimal coding operands and evolutionary algebras of different algorithms in four different multicast scenarios. From the data in the table, it can be found that the convergence rate of AM-QGA



Figure 8. 3-copy network evolution result graph.



Figure 9. 7-copy network evolution result graph.



Figure 10. 15-copy network evolution result graph.



Figure 11. 31-copy network evolution result graph.

is the fastest, and it has the best global search ability and anti-early maturity ability. The convergence performance of the OGA is second only to AM-OGA. GA has the worst convergence performance and anti-local search ability. Because AM-QGA adopts adaptive adjustment mechanism and multi-operator co-evolution mechanism, the search efficiency of the algorithm is much improved. And AM-QGA is not easy to fall into local optimum. It has the best performance among the three algorithms. Because QGA uses qubit coding, population diversity is superior to a genetic algorithm, so its algorithm performance is better than the GA algorithm. Figures 8-11 show the relationship between the optimal operation codes calculated by different algorithms and their evolutionary algebras. It can be seen that when the GA and QGA calculate the big data graph, the optimal solution cannot be found after 500 generations sometimes. This shows that the quantum genetic algorithm is very easy to fall into local optimum, although it converges quickly. By comparison, the BSR of AM-QGA is 100%. It can be seen that AM-QGA has strong global search ability in resolving the network coding resources problem. It can also maintain the diversity of the population well in the later period of the algorithm so that it can easily jump out of the optimal local solution. It can be concluded that AM-QGA has better stability and better global convergence performance after fully considering individual distribution in the population and adjusting the mutation probability.

5. Conclusion

Network coding technology has changed the traditional IP multicast mode which can only store and forward information. It effectively increases the multicast throughput, improves the bandwidth utilization efficiency, and has broad application prospects. In order to solve the problem that QGA is likely to fall into local optimum solution with high convergence rate, the AM-QGA was proposed. It mainly consists of three parts: a cooperative decision-making mechanism based on individual similarity evaluation operator, individual fitness evaluation algorithm, and population mutation adjustment operator. The AM-QGA is used to optimize the network coding resources in multicast network. The AM-QGA has the advantages of fast convergence, the high success rate of multicast, and a strong ability to get rid of the local optimum solution. However, due to the high complexity of the evaluation function, the algorithm has the problem that the time complexity constant is too large and the running time is long. The next research direction is to optimize the evaluation function algorithm or use parallel operations to increase the speed of the algorithm.

Acknowledgements

This work was supported by Natural Science Foundation of Shandong Province (NO. ZR2016FM18); A Project of Shandong Province Higher Education Science and Technology Program (NO. J16LN20).

Conflicts of Interest

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

References

- [1] Williamson, B. (2000) Developing IP Multicast Networks (Volume One). Cisco Press, Indianapolis.
- [2] Qu, Z., Liu, X., Zhang, X., Xie, Y. and Li, C. (2015) A Hamming-Distance-Based Adaptive Quantum-Inspired Evolutionary Algorithm for Network Coding Resources Optimization. *The Journal of China Universities of Posts and Telecommunications*, 22, 92-99. https://doi.org/10.1016/S1005-8885(15)60657-4
- [3] Li, S.R., Yeung, R.W. and Cai, N. (2003) Linear Network Coding. *IEEE Transactions on Information Theory*, 49, 371-381. <u>https://doi.org/10.1109/TIT.2002.807285</u>
- Sanders, P., Egner, S. and Tolhuizen, L. (2003) Polynominal Time Algorithms for Network Information Flow. 15*th ACM Symposium on Parallel Algorithms and Architectures*, San Diego, 7-9 June 2003, 286-294. <u>https://doi.org/10.1145/777463.777464</u>
- [5] Kotter, R. and Medard, M. (2002) Beyond Routing: An Algebraic Approach to Network Coding. *IEEE INFOCOM*, New York, 23-27 June 2002, Vol. 1, 122-130.
- Kotter, R. and Medard, M. (2003) An Algebraic Approach to Network Coding. *IEEE/ACM Transactions on Networking*, 11, 782-795. <u>https://doi.org/10.1109/TNET.2003.818197</u>
- Sengupta, S., Rayanchu, S. and Banerjee, S. (2010) Network Coding-Aware Routing in Wireless Networks. *IEEE/ACM Transactions on Networking*, 18, 1158-1170. https://doi.org/10.1109/TNET.2010.2042727
- [8] Kumar, R., Babu, V. and Nicol, D. (2018) Network Coding for Critical Infrastructure Networks. *IEEE 26th International Conference on Network Protocols*, Cambridge, 24-27 September 2018, 436-437. <u>https://doi.org/10.1109/ICNP.2018.00061</u>
- [9] El Qachchach, I., Habachi, O., Cances, J.P., et al. (2018) Efficient Multi-Source Network Coding Using Low Rank Parity Check Code. IEEE Wireless Communica-

tions and Networking Conference, Barcelona, 15-18 April 2018, 1-6. <u>https://doi.org/10.1109/WCNC.2018.8377229</u>

- [10] Narayanan, A. and Moore, M. (1996) Quantum Inspired Genetic Algorithms. *IEEE International Conference on Evolutionary Computation*, Nagoya, 20-22 May 1996, 41-46.
- [11] Mao, L., Wang, X. and Li, J. (2018) Cloud Computing Resource Scheduling Research Based on the Improved Quantum Genetic Algorithm. *Joint International Advanced Engineering and Technology Research Conference*, Xi'an, 26-27 May 2018, 380-385. <u>https://doi.org/10.2991/jiaet-18.2018.67</u>
- [12] Zhang, Y., Hu, X., Zhu, W. and Jin, P. (2018) Solving the Observing and Downloading Integrated Scheduling Problem of Earth Observation Satellite with a Quantum Genetic Algorithm. *Journal of Systems Science and Information*, 6, 399-420. <u>https://doi.org/10.21078/JSSI-2018-399-22</u>
- Kong, H., Li, N. and Shen, Y. (2015) Adaptive Double Chain Quantum Genetic Algorithm for Constrained Optimization Problems. *Chinese Journal of Aeronautics*, 28, 214-228. <u>https://doi.org/10.1016/j.cja.2014.12.010</u>
- [14] Kim, M., Aggarwal, V., O'Reily, U.M., et al. (2007) Genetic Representations for Evolutionary Minimization of Network Coding Resources. Applications of Evolutionary Computing. LNCS 4448. Springer, Berlin, 21-31. https://doi.org/10.1007/978-3-540-71805-5_3
- [15] Qu, J.Z., Liu, X.H. and Fu, J. (2012) Genetic Algorithm-Based Network Coding Resources Optimization in Multimedia Network. *Proceedings of the* 2012 *International Conference on Systems and Informatics*, Yantai, 19-20 May 2012, 1547-1550. <u>https://doi.org/10.1109/ICSAI.2012.6223333</u>
- [16] Qu, Z.J., Liu, X.H., Huang, J.J., et al. (2012) Genetic Local Search Algorithm for Network Coding Resources Minimization. Proceedings of the 2012 IEEE International Conference on Computer Science and Automation Engineering, Vol. 2, Zhangjiajie, 25-27 May 2012, 782-786. <u>https://doi.org/10.1109/CSAE.2012.6272882</u>