Multiconstrained routing based on artificial bee colony algorithm and dynamic fireworks algorithm

XianWei Zhang1,a, Youbing Hu2,b
1School of computer science, Xi’an Shiyou University, Xian, China
2School of computer science and technology, Xidian University, Xian, China
axwzhang@xsyu.edu.cn, bhuuyoubing@aliyun.com

Abstract. In this paper, a mathematical model for multiconstrained routing optimization problem is established. The multi-objective optimization problem is transformed into a single-objective optimization problem by adding a penalty function. Then the artificial bee colony algorithm (ABC) is used for route search. Because the ABC algorithm is easy to fall into the local optimal deficiencies, the dynamic fireworks algorithm (dynFWA) is introduced for local search, which can ensure fast global search and fast guarantee. In the process of searching for the optimal solution, the success rate is improved by about 1.05% compared with the PSO_ACO algorithm optimized by the ant colony algorithm, which is 6.18% higher than the standard PSO algorithm and the standard ABC algorithm. The minimum average cost of the search is about 0.53% higher than the PSO_ACO algorithm, which is about 1.87% higher than the other two algorithms. The simulation results show that the algorithm can effectively solve the multiconstrained routing problem under large-scale networks.

1. Introduction
With the rapid development of network technology, users have higher requirements for network quality of service (QoS). The most fundamental and important thing in the network is the routing problem, which we can abstract into the shortest path problem in reality [1][2]. The QoS routing problem is to find a solution that optimizes multiple constraints. However, multi-objective optimization problems have always been a problem. At present, many algorithms are introduced to find the optimal solution problem under multiconstraint conditions [3]. There are many ways to solve multi-objective optimization problems in mathematics [6][7], usually using mathematical programming methods and nonlinear programming methods. These mathematical methods are simple to implement, but are affected by the initial point or the objective function, and are easy to fall into the local optimal solution, which cannot meet our needs. Then a new ABC_dynFWA algorithm is proposed, which is used to implement QoS routing. The algorithm is evaluated by simulation experiments and compared with ABC algorithm, PSO algorithm and PSO_ACO algorithm [8].

The rest of this paper is organized as follows: Section 2 builds a mathematical model of multiconstrained QoS routing [11][12]. The third section gives the design and implementation steps of the ABC_dynFWA algorithm. The simulation results and analysis are given in the fourth section. The fifth section summarizes.
2. Building a mathematical model of QoS routing

We abstract the network space into a network topology map. Looking for multiconstrained QoS routes in the network is abstracted to find a path in the network topology map, which satisfies all the given constraints, and the path is still optimal. G(V, E), where V is the set of nodes and E is the set of links. From the source point s to the link $e_{sd}$ of the target node d, the constraints we usually consider include delay $D_{sd}$, bandwidth $B_{sd}$, jitter $J_{sd}$, packet loss rate $L_{sd}$ and cost $\text{cost}(e_{sd})$, and $e_{sd}$ satisfies multiple constraints simultaneously. Cost $(e_{sd})$ must be minimized.

1. In path p, the delay of the link needs to be less than or equal to the delay of the QoS routing requirement $D_{req}$: $\sum_{e_{sd} \in p} D_{sd} \leq D_{req}$. Where $D_{sd}$ is the delay of the link $e_{sd}$ in the path p.

2. In path p, the bandwidth of the link needs to be greater than or equal to the QoS routing requirement block $B_{req}$: $\min\{B_{sd}\}_{e_{sd} \in p} \geq B_{req}$. Where $B_{sd}$ is the bandwidth of the link $e_{sd}$ in the path p.

3. In path p, the jitter of the link needs to be less than or equal to the jitter of the QoS routing requirement $J_{req}$: $\sum_{e_{sd} \in p} J_{sd} \leq J_{req}$. Where $J_{sd}$ is the jitter of the link $e_{sd}$ in the path p.

4. In path p, the packet loss rate of the link needs to be less than or equal to the packet loss rate loss rate $L_{req}$ of the QoS routing requirement: $\prod_{e_{sd} \in p} (1 - L_{sd}) \leq L_{req}$. Where $L_{sd}$ is the packet loss rate of the link $e_{sd}$ in the path p.

5. The link $e_{sd}$ needs to satisfy both the above four constraints and the minimum cost$\text{cost}(e_{sd})$: $\min\{\text{cost}(e_{sd})\}$. Where $\text{cost}(e_{sd})$ is the minimum cost of the entire link.

The task of QoS routing finds a balance among multiple constraints, optimizes resource allocation, finds a path that satisfies multiple QoS parameter constraints, and finds the best path to provide quality of service guarantee to users.

Multi-objective optimization is often difficult, and ABC and FWA algorithms usually have feasible solutions in a continuous space. QoS routing can only find a balance among all resources, and it is impossible to find a solution that makes all constraints optimal. Therefore, the multi-objective optimization problem is transformed into a single-objective optimization problem by the penalty function. At this point, it is only necessary to solve the feasible solution of this single-objective optimization problem, and make it satisfy all the constraint parameters, and find out that one of the optimal links $e_{sd}$ is the target link.

The penalty function is shown in equation (1):

$$
\rho(x) = \alpha \times \max \{|\beta_{wr} - \sin(\beta_{wr})|, 0\} + \beta \times \max \{\sum_{wr} \beta_{wr} - \beta_{wr}^0, 0\} + \theta \times \max \{\prod_{wr} (1 - J_{wr}) = J_{wr}^0, 0\} + \tau \times \max \{\sum_{wr} J_{wr} - J_{wr}^0, 0\}
$$

Where $\alpha$, $\beta$, $\theta$ and $\tau$ are weighting factors for adjusting bandwidth, delay, packet loss rate and jitter. The network's requirements for different requirements can be changed by adjusting the weighting factors of the parameters.

The fitness value is calculated by the formula (2) when the ABC algorithm performs the search:

$$
f(x) = \omega \times \rho(x) + \min \{\sum_{i \in p} \text{cost}(e_{si})\}
$$

Where $\omega$ is a parameter that adjusts the weight ratio between the penalty function and the cost function.

3. ABC_dynaFWA algorithm design

In the ABC_dynaFWA algorithm, each node in the network topology map is abstracted into a honey source. First, the ABC algorithm is used to search from the source point, and each searched node that satisfies multiple constraints is encoded. Whenever the ABC algorithm searches for a honey source with better fitness value than the historical honey source, it uses this honey source as the explosion point of the FWA algorithm, explodes according to formula (3), and explodes according to formula (4) and formula (5). The number and range of sparks produced are limited. The core fireworks that will be produced after the explosion are retained, and the non-core fireworks are discarded. The core fireworks
perform a local search, while the global search relies on leading bees and following bees to search. This will ensure the fastest local search and ensure the fastest global search. When the destination node is searched, the node encoded in the search process is decoded as a link, the fitness value of each such link is calculated, and the link with the smallest fitness value is selected, which is the optimal link.

\[ S_i = m \ast \frac{Y_{\text{max}} - f(x_i) + \varepsilon}{\sum_{j=1}^{m}(Y_{\text{max}} - f(x_j)) + \varepsilon} \]  

(3)

\( S_i \) indicates the number of sparks generated by the i fireworks explosion. m is used for the total number of sparks now produced. \( f(x_i) \) is the fitness value. \( Y_{\text{max}} \) represents the worst fitness value. \( \varepsilon \) is a small constant introduced to prevent the denominator from being zero.

\[ s_j = \begin{cases} \text{round}(a, m), & \text{if } s_i < \text{am} \\ \text{round}(b, m), & \text{if } s_i > \text{bm}, a < b < 1 \\ \text{round}(a, a), & \text{otherwise} \end{cases} \]  

(4)

\[ A_j = \hat{A} \ast \frac{f(x_j) - Y_{\text{min}} + \varepsilon}{\sum_{i=1}^{n}(f(x_i) - Y_{\text{min}}) + \varepsilon} \]  

(5)

The meaning of the representation of the parameter is the same as the formula (3), and \( Y_{\text{min}} \) indicates the best fitness value. \( \hat{A} \) indicates the maximum guaranteed margin.

ABC_dynFWA algorithm step:

S1: Initialize the settings. The SN feasible solutions are randomly generated, and the fitness function values are calculated according to formula (2). SN is equal to the number of employed bees or the number of honey sources, and a feasible solution is randomly generated as shown in equation (6):

\[ x_{jd} = L_d + \text{rand}(0, 1) \ast (U_d - L_d) \]  

(6)

Where \( U_d \) and \( L_d \) represent the upper and lower limits of the solution space.

S2: Hire bees choose honey sources based on greedy strategies. The honey source is both the next hop node of the current node or a neighboring node.

S3: Observe the bee to randomly select the honey source by roulette, and generate a new honey source near the honey source according to formula (7), and calculate the fitness value of the new honey source. If the newly calculated honey source is better than the historical value, the greedy selection method is used to replace the current honey source with the new honey source.

\[ V_{ijd} = x_{ijd} + \rho(x_{ijd} - x_{jbd}) \]  

(7)

Where \( \rho \) is a random number between [-1, 1], \( j \neq i \), randomly selecting a honey source that is not equal to i among SN honey sources.

S4: Each time the best honey source is selected, the current honey source is the initial position of the FWA algorithm. According to the formula (3) explosion, formulas (4) and (5) are used to limit the number and amplitude of sparks generated after the explosion.

S5: At this time, the FWA algorithm is used for dynamic local search, and the ABC algorithm performs global search. If the target node is searched, the node that passes from the source node to the destination node is encoded and recorded, and Step 1 is skipped for the next iteration. If no target node is found, skip Step 3 to search.

S6: Decode all feasible solutions generated during the search process.

The algorithm ends if the number of iterations of the algorithm reaches the set maximum value or the searched optimal solution reaches a given threshold.
4. Simulation evaluation

This paper uses the improved Salama algorithm [18] to randomly generate a network topology map with 50 nodes, as shown in Fig 4.1 (a). In the network topology diagram, there is a road strength between any two nodes, and the probability is 0.5. There are four constraints on the link between any two points: bandwidth, jitter, delay, and packet loss rate, which are used to indicate the metric of the link. Using OPNET and MATLAB, randomly select starting point 8 and destination node 48, and run the algorithm to simulate.

First, we analyze and analyze the success rate of the algorithm under different network nodes, as shown in Fig 4.1 (b). In the figure, at each node, iteration is performed 100 times, and after each iteration, a success rate is obtained, and then an average success rate is obtained. When the number of network nodes is small, the success rates of the ABC_dynFWA algorithm, the PSO algorithm, the ABC algorithm, and the PSO_ACO algorithm are basically the same. As the number of network nodes increases, the success rate of the four algorithms decreases. This is because the number of network nodes increases, the path from the source point to the destination node increases, and the probability of the algorithm falling into local optimum during the search process increases, so the success rate has dropped. However, the success rate of the ABC_dynFWA algorithm is also the highest, followed by the success rate of the PSO_ACO algorithm optimized by the ACO algorithm. The success rate of the ABC algorithm and the PSO algorithm are basically the same. When the number of network nodes is increased to 500 nodes, the average success rate of the four algorithms is significantly different, but the success rate of the ABC_dynFWA algorithm is also the highest.

Secondly, we set different constraints to test the minimum average cost of the four algorithms to find the optimal path. The weight coefficients of each constraint are randomly set, but the sum of the four constraint weights is 1. Here we set the bandwidth weight to 0.5, the delay weight coefficient to 0.2, the jitter weight coefficient to 0.2, and the packet loss rate to 0.1. We set a lower set of QoS constraints (25, 22, 17, 0.3), representing the four constraints of bandwidth, jitter, delay, and packet loss rate. Fig. 4.2(a) shows the minimum average cost of the link we obtained by running the four algorithms under this set of ground constraints. From Fig 4.2(a), we can analyze that as the number of iterations increases, the average cost of the four algorithms decreases, which is the characteristic of heuristic route search. Among them, the fastest decline is the PSO_ACO algorithm and the ABC_dynFWA algorithm. The single ABC_dynFWA algorithm also improves the role of the PSO_ACO algorithm by 0.84%. The performance of the standard PSO algorithm and the standard ABC algorithm is basically the same in the search minimum evaluation cost. The ABC_dynFWA algorithm is about 2.17% higher than the standard PSO algorithm and the standard ABC algorithm. Therefore, under a set of low constraints, the ABC_dynFWA algorithm is significantly better than the other three algorithms in finding the average cost.
When we keep the weight coefficients of the four parameters unchanged and set a higher set of QoS constraints (80, 14, 12, 0.07), we run four algorithms to find the minimum average cost. As shown in Fig. 4.2(b), the minimum average cost of the four algorithms at the beginning is improved because the bandwidth has the largest weight coefficient and the minimum average cost increases as the bandwidth increases. With the increase of the number of iterations, the minimum average cost of the four algorithms in the search is decreasing, and the decline is obvious at the beginning, but the iteration tends to be gentle when it is 50 times. This is also the characteristic of the heuristic swarm intelligence algorithm. It can be clearly seen from the figure that the performance of the ABC_dynFWA algorithm is the best, which is about 0.53% higher than the PSO_ACO algorithm, and the minimum average cost compared to the standard PSO algorithm and the standard ABC algorithm search is increased by about 1.87%.

5. Conclusions

This paper analyzes the existing multi-constrained QoS routing by referring to the data, and converts the multi-objective optimization problem into a single-objective optimization problem through the penalty function according to the characteristics of QoS multi-constrained routing. A new QoS routing algorithm is designed. The algorithm uses the ABC algorithm for global search. When searching for a better food source, the FWA algorithm is used for dynamic search. The dynFWA algorithm can quickly perform local search. This not only ensures the global search, but also guarantees the local search and solves the problem that the ABC algorithm is easy to fall into the local optimal solution during the search process. Finally, a large number of simulation experiments were carried out by combining OPNET and MATLAB. In terms of success rate, the dynFWA algorithm has better performance than the other three algorithms. In terms of finding the minimum average cost, the algorithm is about 0.53% higher than the PSO_ACO algorithm, and the performance of the algorithm is the best compared with the other two algorithms. The evaluation results of simulation experiments show that the proposed algorithm has better performance and feasibility in obtaining large-scale multi-constrained QoS routing.

Acknowledgments

This work is supported by the Special research project of shaanxi education department (No. 16JK1596).

References

[1] Bertsekas D P. Proper Policies in Infinite-State Stochastic Shortest Path Problems[J]. IEEE Transactions on Automatic Control, 2017, PP(99):1-1.
[2] Akbar, Ali, et al. "Shortest-path problem solving in the installation of data/internet network using Apriori algorithm." International Conference on Informatics & Computing 2017:1-5.
[3] Peng C, Zhang L, Pang Z, et al. Fuzzy QoS-driven service selection method for group user[J]. International Journal of Innovative Computing Information & Control Ijicic, 2014, 10(6):2251-2262.
[4] Tanaka K, Taniguchi T, Wang H O. Generalized Takagi-Sugeno fuzzy systems: rule reduction and
robust control}[C] IEEE International Conference on Fuzzy Systems, 2000. Fuzz IEEE. IEEE, 2000:688-693 vol.2.

[5] Pang, Xiaofang. Improved bias proportional navigation with multiple constraints for guide ammunition[J].

[6] Liu Qunfeng. Several Grid-Based Algorithms for Optimization Problems 201

[7] Yoshida K, Okamoto T, Koakutsu S. Equilibrium Solution Search on a Selfish Routing Problem with Multiple Constraints Using the Variable Metric Gradient Projection Method[C] IEEE International Conference on Systems, Man, and Cybernetics. IEEE, 2016:3023-3029.

[8] Er. Navpreet Singh. Evaluating the performance of density grid-based clustering using ABC technique for efficient routing in WSNs// 2017 7th International Conference on Cloud Computing, Data Science & Engineering - Confluence

[9] Kaur H, Sharma S. Analysis of metrics: Improved hybrid ACO-PSO based routing algorithm for mobile ad-hoc network[C]// International Conference on Parallel. 2016:703-708.

[10] Asha G R, Gowrishankar. A hybrid approach for cost effective routing for WSNs using PSO and GSO algorithms[C]// International Conference on Big Data, Iot and Data Science. 2017:1-7.

[11] Luo, Changqing, et al. "Parallel Secure Outsourcing of Large-scale Nonlinearly Constrained Nonlinear Programming Problems." IEEE Transactions on Big Data PP.99(2018):1-1.

[12] Dazhi E, Zhang M. Application of an Intelligent Algorithm in Estimating the Fire Site[J]. Bulletin of Science & Technology, 2015.

[13] Li C, Lu Z, Hu Y, et al. A Parallel Hybrid Intelligent Algorithm for Fuzzy Mean-CVaR Portfolio Model[C] IEEE, International Conference on High PERFORMANCE Computing and Communications; IEEE, International Conference on Smart City; IEEE, International Conference on Data Science and Systems. IEEE, 2018.

[14] Sun, Xiaoyan, et al. "Interactive genetic algorithm assisted with collective intelligence from group decision making." Evolutionary ComputationIEEE, 2012:1-8.

[15] Tan Ying. Fireworks Algorithm: Springer, 2015

[16] Ni M, Wu X, Zheng Y, et al. A Method Based on Penalty Function and Integer Programming for QoS Routing Problem[C]// 2010 International Conference on Multimedia Technology. 2010:1-3.

[17] Linsker R. An iterative-improvement penalty-function-driven wire routing system[J]. Ibm Journal of Research & Development, 2010, 28(5):613-624.

[18] Zhou L. Design and Implement of Random Network Topology Using Waxman-salama Model[J]. Journal of Hunan Institute of Science & Technology, 2008.