Computer Communications 000 (2015) 1–10

Contents lists available at ScienceDirect

Computer Communications

journal homepage: www.elsevier.com/locate/comcom

A controllable chaotic immune algorithm for risk-aware routing in DiffServ networks

Bing Fan [a,*], Ying Zeng [b], Kangming Jiang [b], Liangrui Tang [a]

[a] State Key Laboratory of Alternate Electrical Power System with Renewable Energy Sources, North China Electric Power University, Beijing, China
[b] Electric Power Dispatching and Control Center of Guangdong Grid, Guangzhou, China

ARTICLE INFO

Article history:
Received 22 December 2014
Revised 26 July 2015
Accepted 8 November 2015
Available online xxx

Keywords:
DiffServ network
Risk-aware routing
Chaotic immune algorithm
Controllable evolutionary strategy
Path generation method

ABSTRACT

An integrated routing risk model is constructed, which takes into account the effects of unicast routing on DiffServ network risk consisting of the impacts of interrupted services on network users and path availability. With the objective of minimizing integrated routing risk, a novel controllable chaotic immune routing algorithm (CCIRA) is proposed. Due to the inefficiency of traditional path generation methods, a path generation method based on chaotic search and dynamic adjacency matrix is proposed, improving the generation efficiency of available solutions of routing optimization algorithms. An evolutionary strategy which combines dynamic vaccination and free mutation is used in order to ensure the population diversity and the global convergence of CCIRA. Chaotic search is introduced to population initialization, vaccination and free mutation in order to overcome the uncertainty of the optimization process and optimization results in traditional evolutionary algorithms due to the crossover and mutation strategies being based on random numbers. Simulation results prove that CCIRA is highly efficient and practical. Combining the integrated routing risk model and CCIRA, the risk control performance of our risk-aware routing algorithm is also proved to be superior by the comparison with other algorithms.

1. Introduction

No matter how ingenious new techniques are, network unit failures will always happen [1] naturally or artificially, causing different degrees of impact on network users. Network risk can be characterized by failure probability and the network loss (i.e., the impacts of interrupted services on network users) caused by failures. In DiffServ networks, traffic entering a network is classified and possibly conditioned at the boundaries of the network, and assigned to different behavior aggregates in terms of service level agreement (SLA) [2]. The interruption of a service with high service level will cause greater impact on the network users then the interruption of a service with low service level. Therefore, in addition to service bandwidth, a new metric called service importance, is introduced in this paper to characterize the distinction between traffic with different service levels. The higher the service level, the larger the service importance value. It is worth noting that service with high service importance may have a small bandwidth. In other words, service importance and bandwidth have no direct relationship, which leads us to measure the network loss by different metrics. In this paper, a DiffServ network is abstracted as different logical layers based on different metrics of network risk. We defined three network logical layers: the service layer, transport layer and physical topology layer, and the risk of a link in these three layers is depicted by the service importance metric, service bandwidth metric and path availability metric, respectively. The goal of our research is finding a unicast routing which can minimize the network risk in the three logical layers.

From different perspectives, network risk is studied based on different metrics. Focusing on the business aspect, a risk-aware design and management of resilient networks is proposed in [3], which measured network risk by Value-of-Risk, the maximum penalty to a single service or a whole network with a given confidence interval, due to SLA violation, and presented five risk mitigation strategies considering different trade-offs between budget for risk mitigation and Value-of-Risk. In addition to considering penalty defined in SLA, literature [4] characterized network risk by the product of the penalty per unit time and the probability of network-element failures caused by disasters, and proposed a heuristic algorithm based on finding shortest paths by transforming the penalty, probability of link failures, and free wavelength number into link cost. Penalty is usually determined based on service importance and service bandwidth respectively corresponding to the risk metrics of service layer and transport layer in our research.

* Corresponding author. Tel.: +8613810402555; fax: +861061773844.
E-mail address: bbqice@163.com (B. Fan).

http://dx.doi.org/10.1016/j.comcom.2015.11.003
0140-3664/

Path availability is also a common network risk metric. In order to find the maximum available path under multiple link failures, a series of algorithms are proposed in [5] based on shared risk link groups (SRLGs), which transforms a link belonging to multiple SRLGs into multiple links each belonging to one SRLG and finds a shortest path covered by a SRLG set with the maximum availability using polynomial algorithms or heuristic algorithms. In [6], a risk-aware routing method in optical mesh networks was proposed, which characterizes the quality of a network's optical-layer routing by SLA violation risk instead of statistical path availability because of the inefficiency of path availability alone as the routing metric, and transforms the risk into the failure arrival rate of reference links for calculating the low-risk paths by Dijkstra algorithm. Path availability corresponds to the risk metric of physical topology layer in our research if link availabilities are similar to each other.

Other than metric-based network risk research, a dynamic risk-aware routing for OSPF resilient networks is proposed in [7], which takes advantage of existing failure prediction technologies to anticipate failures and prompt traffic flow to avoid the failures by assigning a high weight to the links related to these failures. Network availability and routing oscillations using this routing mechanism are estimated based on an analytical model, and the results show that the gain is proportional to the ratio of correctly identified failures to the number of all predictions. Jeon et al. [8] proposed a fully distributed algorithm for minimum delay routing under heavy traffic based on Dijkstra algorithm, which is essentially the use of all or part of the link information to achieve network load balancing and minimize the network risk. In [9,10], power communication network routing algorithms were proposed, which can reduce data transmission risk by considering two indexes: 'service risk degree' and 'service risk balance degree'. NSGAII [11] in [9] and an improved Dijkstra algorithm in [10] are used to optimize routing so as to minimize the two indexes.

Network risk should include two factors: failure probability and network loss caused by failures, so risk-aware routing algorithms considering only one factor are one-sided. The approaches taking penalty as network loss are not always optimal due to service provisioning being led by business rather than technological conditions [3]. Additionally, in some special DiffServ networks (e.g., power communication networks), network loss can't be measured by penalty but by actual impacts on users. Therefore, we depict the impact by service importance in the service layer and occupied bandwidth in the transport layer, while depicting the probability by path availability in the physical topology layer, and present a cross-layer integrated routing risk model.

The routing problem considering integrated network risk can be thought of as a multi-constrained routing problem which is a NP-complete problem [12]. Many researchers have used evolutionary algorithms to solve the multi-constrained routing problem such as hybrid genetic algorithm (GA) [13], quantum GA [14,15], chaotic GA [16], immune GA [17] and NSGAII [18,19]. One of the problems with using evolutionary algorithms in routing is the efficiency of generating available solutions (paths). If fixed length binary or natural number encoding mode is used to generate solutions randomly, a large number of unavailable solutions will be generated because most network topology graphs are not complete graph, which reduce the efficiency of evolutionary algorithms seriously. Some scholars have to research routing algorithms based on complete graphs [18]. In [16] and [17] respectively, methods based on tabu and chaotic search using variable length natural number encoding are proposed to generate solutions. However, both methods require removing loops once paths are found, which reduces the available path generating efficiency to some extent.

Another problem with the use of evolutionary algorithms in routing is the uncertainty of output solutions due to the existing of crossover and mutation probabilities. In traditional evolutionary algorithms, the execution of crossover and mutation operations on an antibody depends on whether a random number is within the probability interval. This causes uncertain results, which may be most ideal, less than ideal or even poor, from a single run even if population size and iterations are known. As a result, many scholars resort to using the mean of many runs to prove the superiority of the algorithm [18,20–22]. However, this level of uncertainty is not acceptable in some practical applications such as electric power system communication.

In order to improve the efficiency of generating available path solutions of routing optimization problem, a path generation algorithm based on chaotic search and dynamic adjacency matrix (CDPGA) is presented which can directly calculate an available path without loops. Utilizing the pseudo-randomness and ergodicity of chaotic search and the rapid convergence of artificial immune algorithms, a novel algorithm which integrates dynamic vaccination and free mutation is proposed in the form of the controllable chaotic immune routing algorithm (CCIRA), which is able to increase the convergence speed and guarantee the global optimization capability. Because chaotic search is used in each stage of CCIRA, the ergodicity of the optimization and the determinacy of obtaining optimal solutions is guaranteed, which increases the controllability and practicality of the algorithm.

The rest of this paper is organized as follows. Section 2 introduces the integrated routing risk model. In Section 3, CDPGA and related proof, the dynamic vaccination method, and the overall description of CCIRA are presented. The performance simulation and analysis of CCIRA is given in Section 4. Finally, the conclusions are discussed in Section 5.

2. Integrated routing risk model

2.1. Service layer risk

Service importance can be used to describe the degree of impact on network users due to the interruption of data streams with different service levels in a DiffServ network. For example, a service for a real-time production data stream in power communication networks is more important than a service for a non-real-time office data stream [23]. If the data streams with large service importance values are concentrated on a small number of links in a network, then the network risk is high because a failure in these links will cause a lot of network loss (i.e., make a significant impact on network users). Conversely, if the service importance values of the data streams are distributed uniformly on all links, failures on a link will cause relatively less network loss. In this paper, we use the distribution of service importance value on links to describe the service layer risk.

In the information field, for a ε-ary source, the information entropy is given by

$$H = -\sum_{e=1}^{\varepsilon} p_e \log_2 p_e,\ 0 \le H \le \log_2 \varepsilon \tag{1}$$

where $\sum p_e = 1$, and when $p_e = \frac{1}{\varepsilon}$ the entropy $H = H_{\max} = \log_2 \varepsilon$ [24]. If p_e represents a certain distribution or denotes the proportion of the eth part of an entirety, the more uniform the distribution, the larger the entropy value. This conclusion has been applied to assessing and optimizing portfolio risk [25,26]. In this paper, we use this conclusion to measure the equilibrium degree of the distribution of service importance values on links.

A service layer network model is denoted as $G = (V,E,I)$, where V is the node set, E is the link set, I is the distribution of service importance values on the links in set E, and I_e is the service importance value on link e. The service layer risk is defined as

$$R_X = 1 - \frac{-\sum_{e\in E}(\bar{I}_e \cdot \log_2(\bar{I}_e))}{\log_2(|E|)},\ R_X \in (0,1), \tag{2}$$

where

$$\bar{I}_e = \frac{I_e}{\sum_{e' \in E} I_{e'}} \tag{3}$$

corresponding to the p_e in Eq. 1 is the normalized value of I_e, $|E|$ corresponding to the ε in Eq. 1 is the cardinality of set E. In Eq. 2, the more uniform the distribution of service importance values on links, the larger the value of the entropy $-\sum_{e \in E}(\bar{I}_e \cdot \log_2(\bar{I}_e))$. In order to integrate the service layer risk with other layer risk and convert the monotonicity, the entropy is normalized by Eq. 2 in which the smaller the value of R_X, the lower the network service layer risk.

2.2. Transport layer risk

Within the network transport layer, the index to measure network loss is occupied bandwidth. Similar to the service layer risk model, we use the distribution of occupied bandwidth on links to describe the transport layer risk. A transport layer network model is denoted as $G = (V,E,F)$, where V and E are the same as in 2.1, F is the distribution of occupied bandwidth within link set E, and F_e is the occupied bandwidth of link e. The transport layer risk is defined as

$$R_Y = 1 - \frac{-\sum_{e \in E}(\bar{F}_e \cdot \log_2(\bar{F}_e))}{\log_2(|E|)}, R_Y \in (0,1), \tag{4}$$

where

$$\bar{F}_e = \frac{F_e}{\sum_{e' \in E} F_{e'}} \tag{5}$$

is the normalized value of F_e. The smaller the value of R_Y, the lower the network transport layer risk.

2.3. Physical topology layer risk

In the research on network risk based on network topology, some physical link features, such as distance between the nodes joined by a link, link failure probability, link betweenness and so on, can be transformed into a link weight. If the link weights are additive or approximately additive, the problem of minimizing network risk can be solved by searching the shortest paths (i.e., all paths with the smallest sum of link weights) in a weighted network. Therefore, in a weighted network, the smaller the sum of service path length, the lower the network risk within physical topology layer. For this reason, we use the sum of service path length to describe the physical topology layer risk.

A physical topology layer network model is denoted as $G=(V,E,B)$, where G is a weighted network, V and E are the same as in 2.1, and B is the network service set. The actual path of service b is $p(b)$, the shortest path on the physical topology layer of service b is $p^*(b)$, and the path length of $p(b)$ and $p^*(b)$ are $D_{p(b)}$ and $D_{p^*(b)}$, respectively. The physical topology layer risk is defined as

$$R_Z = 1 - \frac{\sum_{b \in B} D_{p^*(b)}}{\sum_{b \in B} D_{p(b)}}, R_Z \in (0,1). \tag{6}$$

In Eq. 6, the closer the sum of actual service path length is to the sum of shortest service path length, the smaller the value of R_Z, i.e., the lower the physical topology layer risk.

2.4. Integrated routing risk model

In a DiffServ network, service importance is the most direct metric of the impacts on users when there is a service outage. Therefore, if the difference between the importance of different services is rather large, routing should first consider the network's service layer risk in order to minimize the damage from attacks or natural failures. However, if network services have same or similar importance, routing should primarily consider the network's transport layer risk in order to minimize the loss of network traffic from attacks or natural failures. Finally, if network services have a similar degree of importance and bandwidth, routing should consider the physical topology layer risk in order to find the paths with higher availability and reduce the probability of service paths being attacked.

Taking the above into account, an integrated routing risk model with adaptive parameters is defined as

$$R = \alpha R_X + \beta R_Y + \gamma R_Z, R \in (0,1). \tag{7}$$

The values of α, β and γ are

$$\begin{cases} \alpha = 1 - e^{(\frac{\eta_{\min} - \eta_{\max}}{\eta_{\max}})} \\ \beta = (1-\alpha)(1 - e^{(\frac{\theta_{\min} - \theta_{\max}}{\theta_{\max}})}), \\ \gamma = (1-\alpha)e^{(\frac{\theta_{\min} - \theta_{\max}}{\theta_{\max}})} \end{cases} \tag{8}$$

where $\alpha + \beta + \gamma = 1$, $\eta_{\min}$ and $\eta_{\max}$ are respectively the minimum and maximum importance value of services, $\theta_{\min}$ and $\theta_{\max}$ are respectively the minimum and maximum bandwidth of services. In Eq. 8, if $\eta_{\max}$ is far away from $\eta_{\min}$, α will be close to 1, which means the network risk is mainly decided by the service layer risk. On the contrary, if $\eta_{\min}$ and $\eta_{\max}$ are very close, α will be close to 0, which means the services have the same or similar importance, so the network risk is mainly decided by other factors. Similar to the meaning of α, β is the coefficient of transport layer risk. When α is close to 0, if $\theta_{\min}$ and $\theta_{\max}$ are far away from each other, the value of β is large and the network risk is mainly decided by transport layer risk. Otherwise, the network risk is mainly decided by physical topology layer risk. On the whole, the smaller the value of R, the lower the network's integrated routing risk.

3. Controllable chaotic immune routing algorithm

3.1. Path generation

The traditional binary path generation method encodes a network node with a binary string and then uses random binary combinations to generate random nodes in order to obtain a random path [27,28]. This method is very inefficient because it will generate a large number of invalid path solutions for incomplete graphs. The decimal encoding path generation methods mostly use hop-by-hop searching and are more efficient for incomplete graphs than binary encoding. But in order to avoid loops in paths, the visited nodes are removed from the topology during the search process [29], which may result in no path being found. To ensure the methods can find the destination node, others scholars allows the appearance of loops in search results [16,17], which attaches a removing loop step to the methods and decreases the efficiency. In order to further increase the efficiency of path generation, this paper proposes a path generation algorithm based on CDPGA, which we called the CDPGA algorithm.

Assuming that s is the source node, d is the destination node, v_x is the current node in the path generation process, v_x^- and v_x^+ are the upstream and downstream nodes of v_x respectively, network $G=(V,E)$ is a simple, connected, undirected graph, and the initial network adjacency matrix is

$$A^0 = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ & \ddots & & \\ a_{n1} & a_{n2} & \dots & a_{nn} \end{bmatrix}, \tag{9}$$

where

$$a_{ij} = a_{ji} = \begin{cases} 1, & link(v_i, v_j) \in E \\ 0, & others \end{cases}, \tag{10}$$

Algorithm 1
Path generation algorithm based on chaotic search and dynamic adjacency matrix (CDPGA)

```
1:  Input: G=(V,E), s, d, A^0.
2:  Output: p(s,d).
3:
4:  Let p(s,d)={s}, A=A^0, v_x = s, φ_i = {v_j | a_ij = 1, a_ij ∈ A, ∀j};
5:  while v_x ≠ d do
6:      φ_i = φ_i − {v_x}, ∀i;
7:      A=update(A);
8:      If in A, φ_x ≠ φ then
9:          Use chaotic search to find the downstream node v_x^+;
10:         Add v_x^+ to p(s,d);
11:         Let v_x = v_x^+;
12:     else if v_x = s then
13:         return FAILURE;
14: Else
15:         Remove node v_x from p(s,d);
16:         Let v_x = v_x^- (the upstream node of v_x in p(s,d));
17:         end if
18:         end while
19: return p(s,d);
```

$n = |V|$, the path generation algorithm is as follows:

Initially, let the current adjacency matrix $A = A^0$, the current node $v_x = s$, the current path node set $p(s,d)$={s}, and the neighbor node set of node v_i in the current adjacency matrix A be $\varphi_i = \{v_j | a_{ij} = 1, a_{ij} \in A, \forall j\}$. Then, perform $\varphi_i = \varphi_i - \{v_x\}$, $i = 1, 2, \ldots, n$, i.e., make all elements of the xth column of A be '0'. The new adjacency matrix is denoted by A', and let A=A'. If in A, $\varphi_x \neq \phi$, find the downstream node v_x^+ using chaotic search and add node v_x^+ to the path node set $p(s,d)$. Let $v_x = v_x^+$, and iterate above process excluding the initial part until finding the destination node d. In order to avoid generating loops, all nodes in $p(s,d)$ will be removed from the neighbor node sets of all nodes in the network before searching a new path node. In the iteration, if the number of a node's neighbor is zero, it is indicated that the algorithm cannot find the destination node through this node, so this node will be removed from the network and the algorithm will return to the upstream node to continue to search the path. The pseudo-code of CDPGA is shown in Algorithm 1.

The chaotic search method uses a pseudo-random number δ within an interval (0,1) generated by a chaotic system to calculate

$$m = ceil(\delta \cdot h) \tag{11}$$

in which h is a search object-related parameter and here $h = |\varphi_x|$, the function of $ceil(.)$ is rounding up, and the mth element of φ_x is the downstream node v_x^+.

The effectiveness of the algorithm is proved below:

Proposition 1. *The algorithm is able to find path $p(s,d)$ in connected graph G.*

Proof. Since G is a connected graph, path $p(s,d)$ must exist and so its node set can be denoted as $p(s,d) = \{v_1, v_2, v_3, \ldots, v_{L-1}, v_L\}$ where $v_1 = s$, $v_L = d$ and $d \in \varphi_{L-1} \Rightarrow \varphi_{L-1} \neq \phi$. If Proposition 1 is not true and the algorithm terminates without finding a valid path, according to the termination conditions in step 5 it can be deduced that $\varphi_1 = \phi$. Denoting the neighbor node set of v_1 in A^0 by φ_1^0, then $v_2 \in \varphi_1^0 \Rightarrow \varphi_2 = \phi$, otherwise the algorithm will search for the downstream node of v_2 in $\varphi_2$and will not terminate. Similarly, $\varphi_2 = \phi \Rightarrow \varphi_3 = \phi \Rightarrow \cdots \Rightarrow \varphi_{L-1} = \phi$. $\varphi_{L-1} = \phi$ and $\varphi_{L-1} \neq \phi$ conflict, so the hypothesis cannot be established and Proposition 1 is confirmed.

Proposition 2. *Paths obtained by the algorithm do not contain loops.*

Proof. If Proposition 2 is not true and the algorithm obtains a path with loops, $p(s,d) = \{s, v_2, v_3, \ldots, v_{L'-1}, v_{L'}, \ldots, d\}$can be assumed where $v_{L'} = v_3$. According to steps 4 and 2, after v_3 is added to $p(s,d)$ as the downstream node of v_2, $v_3 \notin \varphi_i (i = 1, 2, ..., n) \Rightarrow v_3 \notin \varphi_{L'-1}$. When the algorithm reaches $v_x = v_{L'-1}$, $v_x^+ = v_{L'} = v_3 \Rightarrow v_3 \in \varphi_{L'-1}$. $v_3 \in \varphi_{L'-1}$ and $v_3 \notin \varphi_{L'-1}$ conflict, so the hypothesis cannot be established and Proposition 2 is confirmed.

3.2. Dynamic vaccination

To ensure that good genes can be preserved during evolution process, artificial immune algorithms use the vaccination strategy which is that some of the antibody genes are assigned values directly based on prior knowledge [30], and do not mutate during the mutation procedure. However, multi-constrained routing problems have no prior knowledge, and the antibody length changes randomly. Therefore, it is impossible to use traditional vaccination methods which select constant genetic positions to assign values. This paper suggests a dynamic vaccination method based on chaotic search, which is able to largely retain the outstanding genes and increase the convergence speed of CCIRA.

Assuming an outstanding antibody as$p(s_0, d_0) = \{s_0, v_2, v_3, \ldots, v_{L-1}, d_0\}$, where $L = |p(s_0, d_0)|$, substitute $h = L$ into Eq. 11, so that

$$y = \begin{cases} L - l\ - 1, & \mathrm{mod}(m, L - l - 1) = 0 \\ \mathrm{mod}(m, L - l - 1), & \mathrm{mod}(m, L - l - 1) \neq 0 \end{cases}, \tag{12}$$

where l is the dynamic vaccination parameter that denotes the number of mutating genes in the antibody; $L - l$ is the length of the vaccine, which actively changes according to the length of the antibody; $mod(.)$ is the remainder function; and the genes preserved by vaccination are denoted by $p'(s_0, d_0) = \{s_0, v_2, \ldots, v_y\} \cup \{v_{y+l+1}, \ldots, d_0\}$. Let $s = v_y, d = v_{y+l+1}$, and search for a new path $p(v_y, v_{y+l+1})$ using the CDPGA algorithm. Then a new antibody $p''(s_0, d_0) = p'(s_0, d_0) \cup p(v_y, v_{y+l+1})$ is generated after vaccination. In order to avoid the appearance of loops during mutation, $\varphi_i = \varphi_i - p'(s_0, d_0)$, $i = 1, 2, ..., n$, needs to be performed before running the CDPGA algorithm.

In the antibody selection step of CCIRA, the paths generated after dynamic vaccination are compared to the original paths. The outstanding antibodies will be preserved and the others will freely mutate or be discarded. In reality, the dynamic vaccination method uses the ergodicity of chaotic search to compensate for the lack of prior knowledge in order to find genuine outstanding genes and to retain them.

3.3. Overall algorithm realization

In order to minimize the integrated network routing risk, Eq. 7 is selected as the affinity function of CCIRA. The smaller the affinity function value, the more outstanding the antibody. Assuming that service b is assigned to a newly arrived service request, s_0 is the source node, d_0 is the destination node, b_{im} is the importance value, b_f is the needed bandwidth, the steps of the algorithm are as follows:

Step 1: Population initialization

Let N be the population size, G be the maximum iterations, and the current generation number $g = 1$. Using chaotic equation to obtain a matrix $C = [\delta_{i,j}]$ where $\delta_{i,j}$ is in interval (0,1) and is the required chaotic value of the jth search for the downstream node of the ith node, the CDPGA algorithm is used to generate N valid paths to form the initial population $P^0(g) = \{p_1^0(g), p_2^0(g), \ldots, p_N^0(g)\}$.

Step 2: Affinity calculation and antibody selection

Assigning each antibody (path) in P^0(g) to service b, the network integrated routing risk after adding service b is calculated by Eq. 7. Assuming the qth antibody in P^0(g) is $p_q^0(g)$, I_e in Eq. 3 and F_e in Eq. 5 are calculated as follows:

$$I_e = \begin{cases} I_e^0 + b_{im}, & e \in p_q^0(g) \\ I_e^0, & e \notin p_q^0(g) \end{cases}, \tag{13}$$

$$F_e = \begin{cases} F_e^0 + b_f, & e \in p_q^0(g) \\ F_e^0, & e \notin p_q^0(g) \end{cases}, \tag{14}$$

where I_e^0 and F_e^0 denote the service importance value and occupied bandwidth respectively of link e in the network before adding service b. The antibodies are then sorted in ascending order of affinity, and $P^0(g)$ is changed to $P^1(g) = \{p_1^1(g), p_2^1(g), \ldots, p_N^1(g)\}$.

Select the first S antibodies from $P^1(g)$ to form current memory population $M(g)$, where $\mathbf{S} = ceil(\lambda \cdot \mathbf{N})$ and $\lambda \in (0, 1)$ is the $M(g)$ scaling factor.

If $g = G$, the best antibody in $M(g)$ is selected to output and the algorithm terminates. Otherwise, go to step 3.

Step 3: Dynamic vaccination

In order to increase the probability of preserving the excellent genes, the antibodies in $M(g)$ are cloned and the clone number of the qth antibody is

$$c_q = round\left(\mu \cdot \frac{1 - R(q)}{\sum_{q=1}^{S}(1 - R(q))}\right), \tag{15}$$

where $R(q)$ is the integrated routing risk of antibody q, and μ is the clone scaling factor. The cloned antibodies constitute the clone population $P^2(g) = \{p_1^2(g), p_2^2(g), \ldots, p_H^2(g)\}$, where $H = \sum_{q=1}^{S} c_q$. In accordance with the dynamic vaccination method described in Section 3.2, the antibodies in $P^2(g)$ are vaccinated and mutated in order to generate the vaccinated clone mutation population $P^3(g) = \{p_1^3(g), p_2^3(g), \ldots, p_H^3(g)\}$.

Step 4: Free mutation

Free mutation is used to prevent the algorithm from stopping at a local optimum and improve the global search ability. In order to simplify the algorithm, the values of λ and μ can be appropriately assigned so that $S + H < N$. Then the antibodies in the population $P^4(g) = \{p_1^4(g), p_2^4(g), \ldots, p_{N-S-H}^4(g)\} = \{p_{S+1}^1(g), p_{S+2}^1(g), \ldots, p_{N-H}^1(g)\}$ are freely mutated. Assuming the qth antibody of $P^4(g)$ is $p_q^4(g) = \{s_0, v_2, v_3, \ldots, v_{L-1}, d_0\}, L = |p_q^4(g)|$, substituting $h = L$ into Eq. 11, if $m \neq L$, use the CDPGA algorithm to search a new path $p(v_m, d_0)$ for obtaining the freely mutated antibody $p_q^5(g) = \{s_0, v_2, \ldots, v_{m-1}\} \cup p(v_m, d_0)$. If $m = L$, repeat chaotic search and free mutation. The post-free mutation population is $P^5(g) = \{p_1^5(g), p_2^5(g), \ldots, p_{N-H-S}^5(g)\}$.

Step 5: Population updating

The new population generation is $P^0(g+1) = M(g) \cup P^3(g) \cup P^5(g)$. Let g=g+1, and return to step 2.

The pseudo-code of CCIRA is shown in Algorithm 2.

Algorithm 2
Controllable chaotic immune routing algorithm (CCIRA)

Input: $s_0, d_0, b_{im}, b_f, N, G, C, \lambda, \mu$.
Output: the optimal path $p(s_0, d_0)$.

Let g=1;
Generate N paths from s_0 to d_0 using CDPGA to constitute population $P^0(g) = \{p_1^0(g), p_2^0(g), \ldots, p_N^0(g)\}$;
while $g \leq G$ **do**
  for q=1 to N;
    Compute R in Eq. 7 when the path of service b is $p_q^0(g)$;
    Let the affinity of Antibody q be $R(q)$=R;
  end for
  Sort the antibodies in $P^0(g)$ in ascending order according to their affinities, resulting in $P^1(g) = \{p_1^1(g), p_2^1(g), \ldots, p_N^1(g)\}$;
  Let $\mathbf{S} = ceil(\lambda \cdot \mathbf{N})$;
  Select the first S antibodies from $P^1(g)$ to form current memory population $M(g)$;
  if $\mathbf{g} = G$ **then**
    Return the best antibody in $M(g)$;
  end if
  Let $P^2(g) = \phi$;
  for q=1 to S
    Clone the antibody q in $M(g)$ with the scale $c_q = round(\mu \cdot \frac{1-R(q)}{\sum_{q=1}^{S}(1-R(q))})$;
    $P^2(g) = P^2(g) \cup \{$cloned antibodies$\}$;
  end for
  Perform Dynamic Vaccination on $P^2(g)$, resulting in $P^3(g)$;
  Let $P^4(g) = \{p_{S+1}^1(g), p_{S+2}^1(g), \ldots, p_{N-H}^1(g)\}$, where $H = |p_q^3(g)|$ ***s.t.*** $S + H < N$;
  Perform Free Mutation on $P^4(g)$, resulting in $P^5(g)$;
  Let $P^0(g+1) = M(g) \cup P^3(g) \cup P^5(g)$, g=g+1;
end while

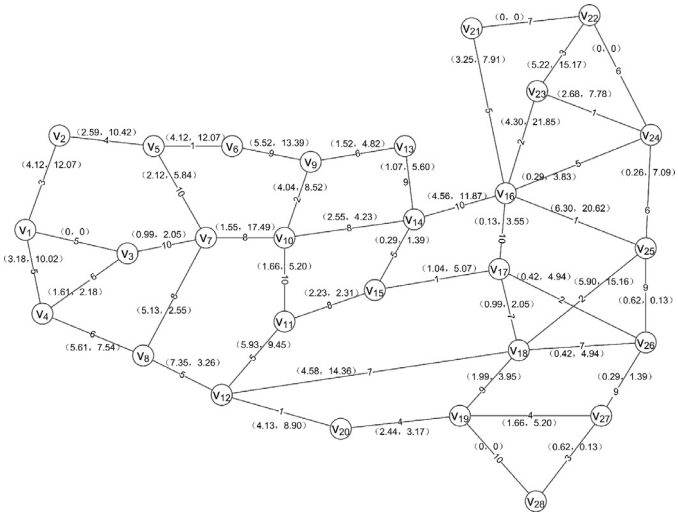


Fig. 1. LATAX network topology, and background service importance value.

4. Optimization performance of CCIRA

4.1. Simulation environment and parameters

Evolutionary algorithms are most suitable for relatively large networks. Therefore, this paper selected the LATAX network (shown in Fig. 1) with 24 nodes and 46 links and the Italian national network (ITNA network) (shown in Fig. 2) with 33 nodes and 68 links [31] as simulation networks.

Referring to [23], there are 5 types of services in the network and their importance value vector is (0.99,0.94,0.62,0.29,0.13). According to the traffic data in [32] and the classification method proposed in [23], the unitized service bandwidth vector is (2.048,2.048,0.133,1.387,3.547).The numbers of existing services in the two networks are $b_{LATAX} = 50$ and $b_{ITNA} = 200$ respectively. The type and source/destination nodes of existing services are randomly allocated, and the existing service paths are the shortest paths. In Fig. 1, the number on top of each link indicates the unitized link length, and the numbers in brackets next to each link indicate the importance value and the occupied bandwidth of the link respectively. Because there are a large number of nodes and links, only the link lengths are shown in Fig. 2.

The simulation parameters are as follows: the new service b is a type two service; $b_{im} = 0.94$; $b_f = 2.048$; the source and destination nodes in the LATAX and ITNA networks are (v_1, v_{25}) and (v_1, v_{29}) respectively; the population size N=30; the iterations G=50; the $M(g)$ scaling factor $\lambda = 0.2$; the clone scaling factor $\mu = 0.4$; and the dynamic vaccination parameter $l = 2$. According to Eq. 8, the affinity function $R = 0.58R_X + 0.26R_Y + 0.16R_Z$.

Because chaotic search is used in three stages – path generation, dynamic vaccination and free mutation, chaotic sequences with a

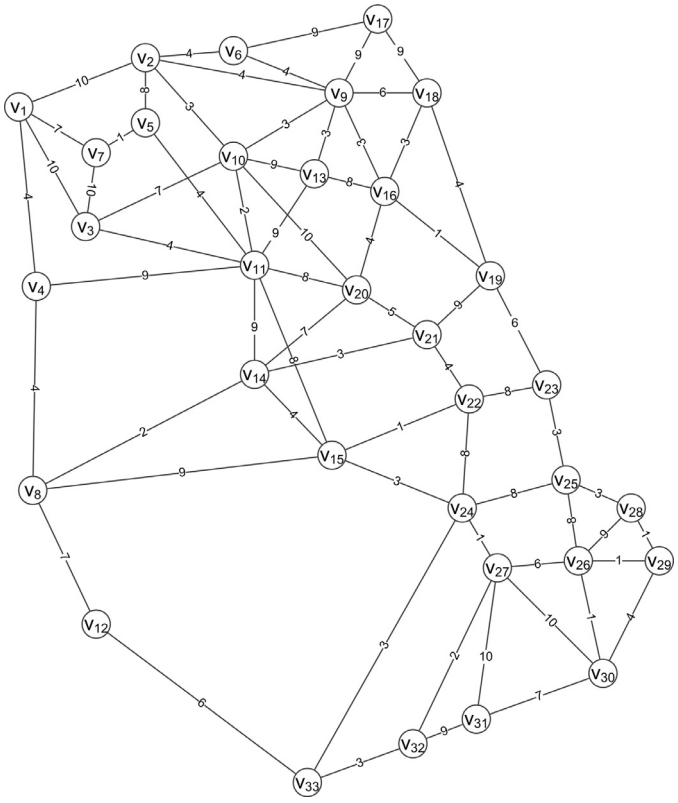


Fig. 2. ITNA network topology and occupied bandwidth distribution.

single state variable (e.g. Logistic chaotic sequences) may correlate between different stages. To avoid the correlations, this paper uses the hyper-chaotic system and mapping method in [33] to generate five chaotic sequences within interval (0,1), in which the sequence of state variable x_1, x_2 and x_3 are used for path generation, dynamic vaccination, and free mutation, respectively.

4.2. Simulation results and analysis

In this section, as our main goal is to prove the outstanding performance of CCIRA in solving routing problems, we select the QoS chaotic GA (QCGA) in [16] and the immune genetic routing algorithm (IGRA) in [17] to compare with CCIRA. During the simulation, the crossover and mutation probabilities in QCGA are p_c=0.7 and p_m=0.1 respectively, while in IGRA p_c=0.5 and p_m=0.1, as shown in [16] and [17]. It is worth noting that our integrated routing risk model is an open model, which means if service importance is measured by service penalty and path length is calculated based on link availability, our integrated routing risk model can contain the risk factors mentioned in [3-6].

The minimum integrated routing risk of the gth generation and the entire optimization process is denoted by $R(g)$ and $R(g)_{\min}$, respectively. When there are no constraints, the optimization processes of all three algorithms on the LATAX network are shown in Fig. 3. The optimal paths and their $R(g)_{\min}$ values obtained by the three algorithms are shown in Table 1.

From Fig. 3, it can be seen that the optimization process of CCIRA is stationary and produces the best output of all three algorithms; QCGA falls into local optimum at $g = 24$; IGRA is good at maintaining population diversity due to the introduction of antibody concentration, but in [17] an elite preservation strategy is not utilized resulting in the emergence of an oscillatory state. However, at $g = 42$ the local optimal solution of IGRA is the same as QCGA. In Table 1, the three

Table 1
Optimal paths and $R(g)_{\min}$ of the three algorithms.

Algorithm	$R(g)_{\min}$	Optimal path $p(v_1,v_{25})$
CCIRA	0.0644	p={1,3,7,10,14,15,11,12,20,19,28,27,26,17,16,21,22,24,25}
QCGA	0.0648	p={1,3,4,8,12,20,19,28,27,26,17,16,21,22,24,25}
IGRA	0.0648	p={1,3,4,8,12,20,19,28,27,26,17,16,21,22,24,25}

algorithms obtained similar optimal paths with the same path segments {1,3} and {12,20,19,28,27,26,17,16,21,22,24,25}. The similarity percentage is above 60%, which confirms the existence of a vaccine during the antibody evolution and that vaccination can be used to accelerate the convergence rate.

The three algorithms attempt to traverse all links with the smallest service importance value and occupied bandwidth in order to minimize the integrated routing risk. This causes long paths that can be seen in Fig. 1, which may not meet the delay requirements of some services in practice because there are no constraints on the algorithms and the difference of service importance values and occupied bandwidth between the different types of services in the network is relatively large. Taking this into consideration, a delay constraint – a maximum number of path nodes denoted by $Hp_{\max}$ (the service delay in this paper is mostly generated by node forwarding) – can be added to the algorithms according to the service requirements. In this paper, $Hp_{\max}$(LATAX)=10, $Hp_{\max}$(ITNA)=15. In order to maintain the efficiency of the algorithms under the constraint, during the optimization, if an antibody is unable to meet the constraint, its R value will be set to '1' instead of being re-generated.

Under the constraint, two simulations were performed on the LATAX and ITNA networks respectively and the results are shown in Figs. 4 and 5.

It is obvious that the optimization processes and results of QCGA and IGRA are stochastic because the crossover and mutation strategies are based on probability. In Fig. 4(a), QCGA finds a local optimal solution at g=32 when $R(g)_{\min}$=R(32)=0.0709; while in Fig. 4(b) it finds a local optimal solution at $g = 41$ when $R(g)_{\min}$=R(41)=0.0727, showing distinct differences both in the optimization process and results. In Fig. 4(a), IGRA finds a local optimal solution at g=44 when $R(g)_{\min}$=R(44)=0.0722; while in Fig. 4(b), a local optimal solution is found at $g = 40$ when $R(g)_{\min}$=R(40)=0.0731 which is inferior to the optimal solution found in Fig. 4(a). The simulation results described above indicate that the uncertainty of the optimization process and solutions is a common problem of evolutionary algorithms based on crossover or mutation probability. In a practical application, the uncertainty will cause problems when it comes to setting parameters such as population size and iterations, and possibly outputting a poor solution after a single run.

On the contrary, the processes and results of CCIRA are entirely consistent during the two simulations as shown in Fig. 4. The optimal solutions of both simulations are similar at $g = 19$ when $R(g)_{\min}$=R(19) = 0.0702 and are superior to the solutions of both QCGA and IGRA. The dynamic vaccination ensures that CCIRA is highly efficient at local optimization, while the free mutation is utilized for breaking out of the local optimum and ensuring global optimization. These two procedures enable CCIRA to quickly converge to a new optimal solution once it breaks out of local optimum. CCIRA takes full advantage of the pseudo-randomness and ergodicity of chaotic search to achieve antibody mutation and dynamic vaccination instead of stochastic strategies based on probability in order to use the determinacy of chaotic trajectory in phase space to overcome the uncertainty of traditional algorithms. From the simulation results it can be seen that the novel mutation and dynamic vaccination strategies not only ensure the determinacy but also improve the performance of evolutionary algorithms. In CCIRA, chaotic search is used in each of the three important stages: path generation, dynamic

Please cite this article as: B. Fan et al., A controllable chaotic immune algorithm for risk-aware routing in DiffServ networks, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.11.003

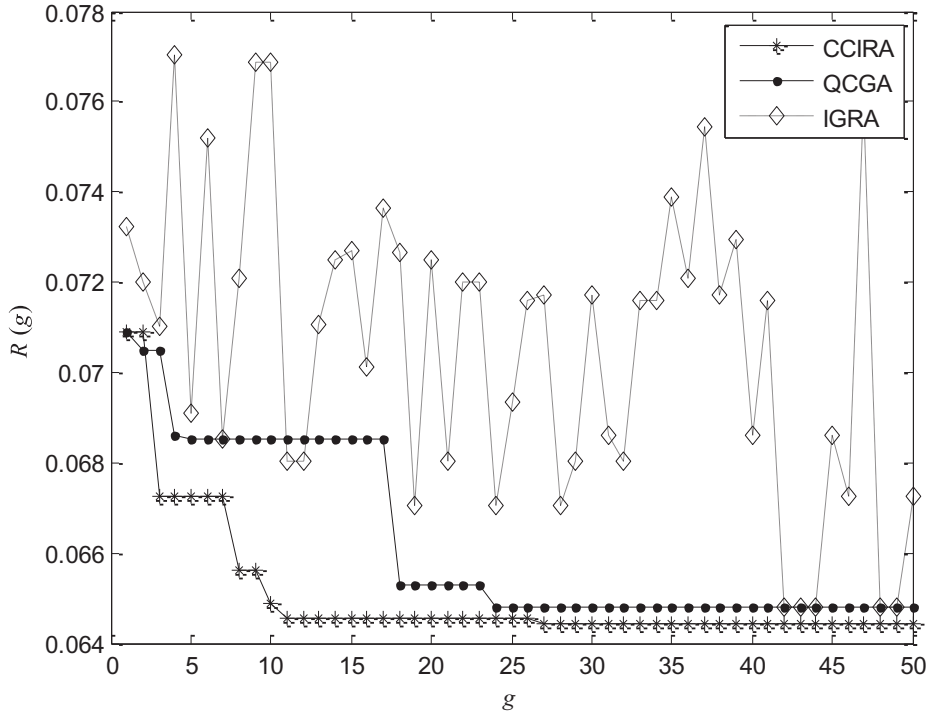


Fig. 3. Optimization processes of the three algorithms with no constraints.

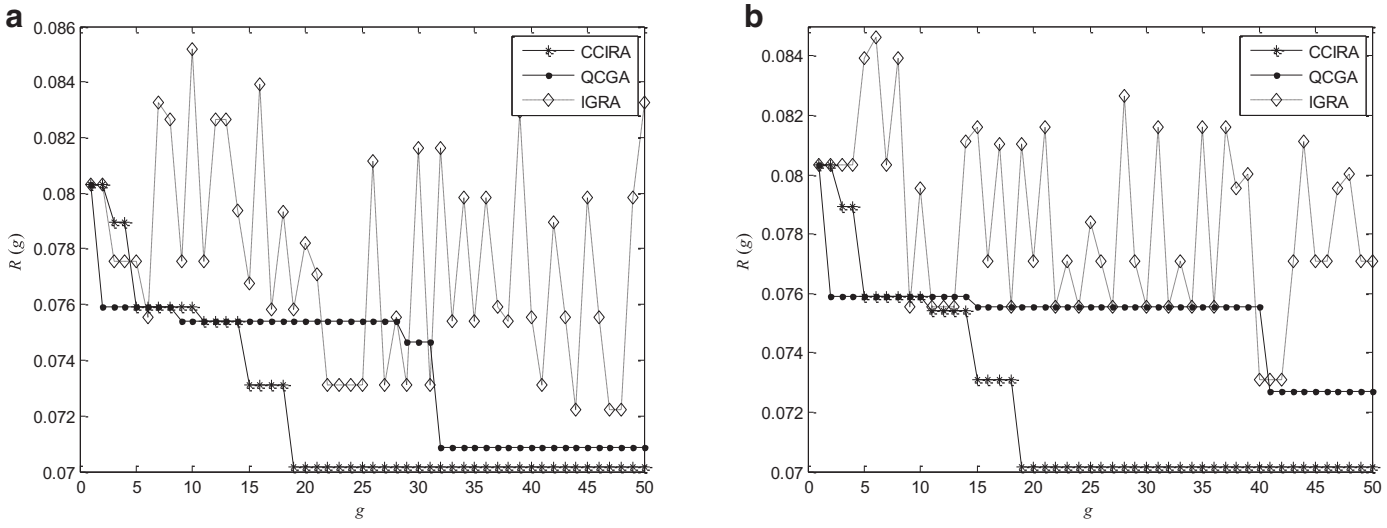


Fig. 4. Results of two random simulations on the LATAX network.

vaccination, and free mutation, giving the algorithm a degree of controllability compared to other evolutionary algorithms. For example, we can change the chaotic trajectory by setting the initial value or parameters of the chaotic system to improve the performance of CCIRA.

Similar results are shown in Figs. 4 and 5, but the performance of IGRA declined compared to both QCGA and CCIRA with the optimal solution being noticeably inferior to both algorithms. Since the ITNA network has a larger number of nodes and links compared to the LATAX network, the solution space becomes larger, which causes CCIRA to be only able to find the optimal solution at $g = 40$. Because IGRA introduces antibody concentration and the differences between antibodies will increase in a larger solution space, the effect of antibody concentration rather than affinity on the optimization orientation increases, reducing the performance of IGRA.

To demonstrate that the superior overall performance of CCIRA compared to QCGA and IGRA is not limited to only two random simulations, we compared the means of 50 simulation results for both QCGA and IGRA with the results of CCIRA on the LATAX and ITNA networks. As shown in Fig. 6(a), on the LATAX network, QCGA and IGRA converged slowly and CCIRA is distinctly superior to QCGA and IGRA both in terms of convergence rate and output solution performance. In Fig. 6(b), on the ITNA network, QCGA and IGRA converge fast, but when $g > 10$, they both fall into local optimum and converge very slowly. When $g \leq 20$, QCGA is superior to CCIRA both in terms of convergence rate and solution performance. However, when $g > 20$, its performance is inferior to CCIRA because of prematurity. Similar to the results of a single simulation, the performance gap between IGRA and the other two algorithms is more evident in Fig. 6(b) due to the larger solution space.

Please cite this article as: B. Fan et al., A controllable chaotic immune algorithm for risk-aware routing in DiffServ networks, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.11.003

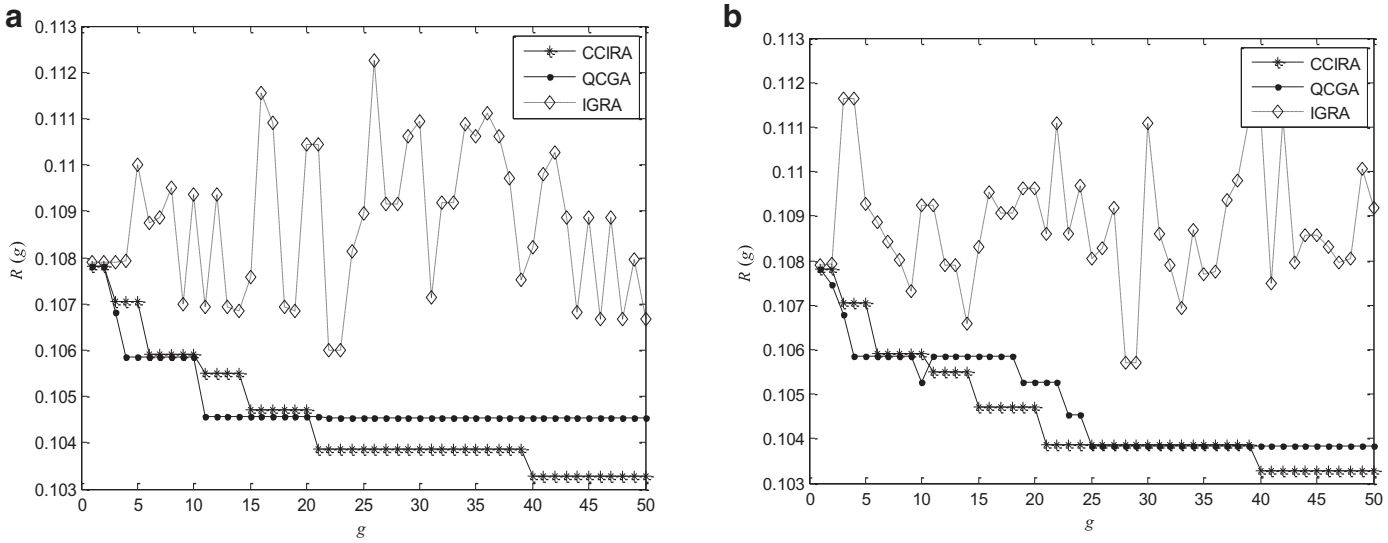
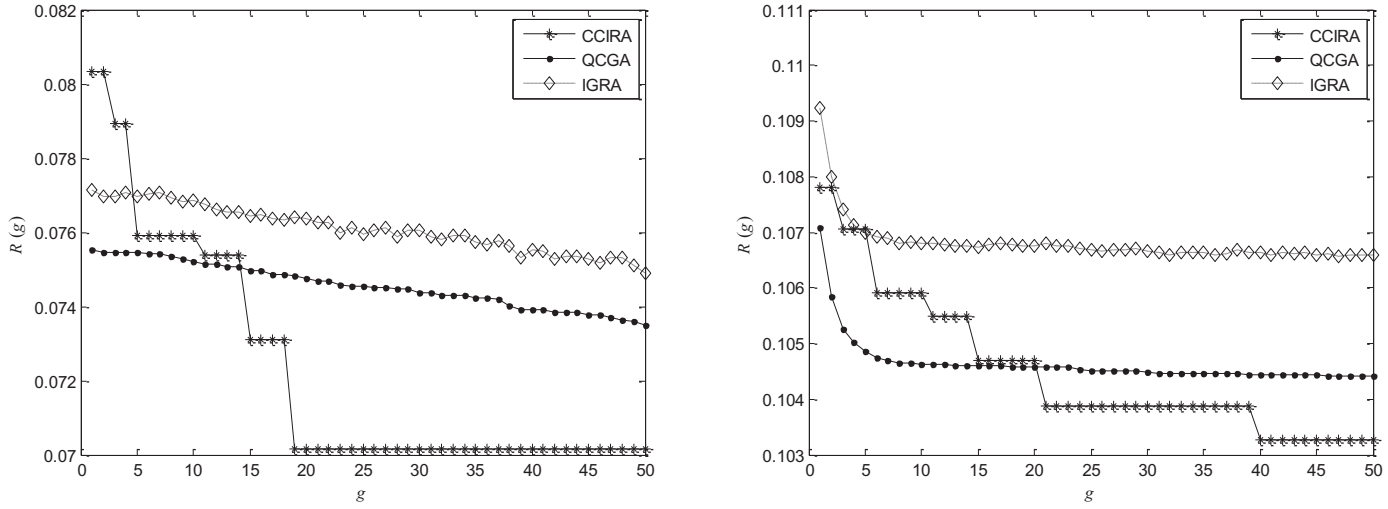


Fig. 5. Results of two random simulations on the ITNA network.



(a) LATAX network (b) ITNA network

Fig. 6. Overall performance comparison of the three algorithms on the two networks.

5. Risk control performance analysis

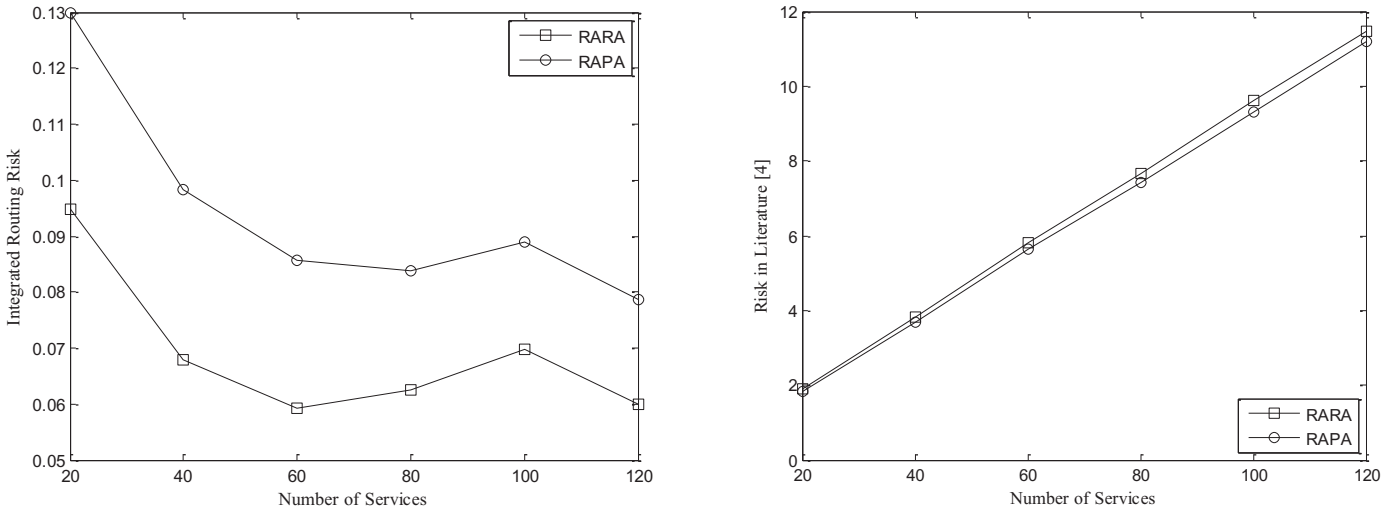
5.1. Simulation parameters

In this section, the risk control performance of our risk-aware routing algorithm (RARA), which is the combination of the integrated routing risk model and CCIRA, is simulated and analyzed. The risk-aware provisioning algorithm (RAPA) in [4] is selected as the comparison algorithm. Literature [4] takes into account two risk factors which are SLA violation penalty per unit time and link failure probability under different types of disasters, and defines network risk as the sum of all connection (path) risk, which is the product of the penalty and the path failure probability, under all disasters. RAPA weights a link with the penalty and the link failure probability if the link has enough free capacity and finds the shortest path as the routing of a service.

In order to compare RARA with RAPA, the service importance in our integrated routing risk model is measured by service penalty, i.e., the service importance value vector (0.99, 0.94, 0.62, 0.29, 0.13) in Section 4.1 is taken as service penalty vector in this section, and the path length is the path failure probability which is approximately equal to the sum of the failure probability of all links on this path [6], so the link weight is converted into link failure probability through multiplying the original link length by a basic probability p_{base}= 0.01. Under the premise of not affecting the performance comparison between RARA and RAPA, we assume that one link can only be affected by one type of disaster, and different link failure probabilities correspond to different types of disasters. Because the service bandwidth in [4] is equal to each other, the service bandwidth vector in this section is (1,1,1,1,1). The LATAX network (shown in Fig. 1) is selected as the simulation network and the delay constraint $Hp_{\max}$(LATAX)=10 of CCIRA is similar to Section 4.2.

5.2. Simulation and analysis

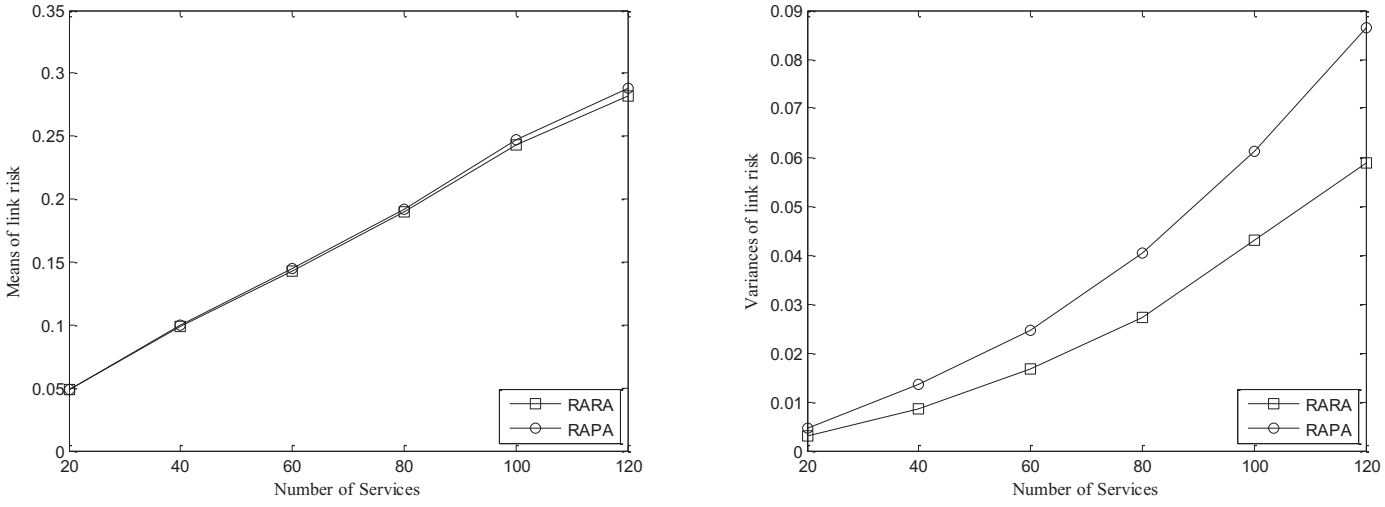
In the simulation, different numbers (20, 40, 60, 80, 100 and 120) of services are routed in descending order with regard to their penalty values, as with RAPA in [4], and the source-destination node pairs of services are randomly assigned in every run. The average network risk of 50 runs is shown in Fig. 7 where Fig. 7(a) shows the network risk defined in our paper by Eq. 7 and Fig. 7(b) shows the network risk defined in [4]. As the network risk defined in this paper emphasizes



(a) Network risk defined in this paper

(b) Network risk defined in literature [4]

Fig. 7. Risk control performance comparison between RARA and RAPA on LATAX network.



(a) Means of link risk

(b) Variances of link risk

Fig. 8. Distribution of link risk on LATAX network under RARA and RAPA.

the equilibrium of the load on different layers, the risk values reflect the relative risk level of the network with current load, and the risk curves are not monotonic with the number of services, as shown in Fig. 7(a). On the other side, the risk curves in Fig. 7(b) are monotonically increasing with the number of services because the network risk defined in [4] reflects the absolute risk level. From Fig. 7, it can be seen that the relative risk control performance of RARA is evidently superior to RAPA and the absolute risk control performance of RARA is marginally inferior to RAPA, which demonstrates that the overall risk control performance of RARA is superior to RAPA.

Because the service bandwidths are equal to each other in the simulation, according to the basic definition of risk, that risk is the effect of uncertainty on objectives [34], the link risk can be defined as $R_{link} = I_e \times P_e$, where I_e is the penalty sum of all services on this link and P_e is the link failure probability. In order to compare the risk control performance between RARA and RAPA fairly, the distribution of the link risk is observed by the average of 50 runs of the two algorithms and the results are shown in Fig. 8, where Fig. 8(a) shows the means of link risk and Fig. 8(b) shows the variances of link risk. In Fig. 8, the link risk mean of RARA is marginally superior to RAPA but the link risk variance of RARA is evidently superior to RAPA. The curves in Fig. 8 imply that under the two algorithms the average effects of single-link failure on the network are similar but the distribution of link risk under RARA is more balanced than the distribution under RAPA. In Fig. 8(b), the gap between the two curves increases with the increase of service number, which implies that the advantage of RARA will become more evident in the case of heavy network load.

6. Conclusions

In this paper, taking into consideration the effect of service routing on network service layer, transport layer, and physical topology layer risk, we created an integrated routing risk model, after which we proposed a controllable chaotic immune routing algorithm (CCIRA) in order to reduce the routing risk. Due to the inefficiency of traditional path generation methods, we proposed a path generation method based on chaotic search and dynamic adjacency matrix. This method can efficiently generate feasible solutions, improving the efficiency of routing optimization algorithms. In CCIRA, the use of a method that combines dynamic vaccination and free mutation ensures the convergence rate and global optimization capability. The use of chaotic

Please cite this article as: B. Fan et al., A controllable chaotic immune algorithm for risk-aware routing in DiffServ networks, Computer Communications (2015), http://dx.doi.org/10.1016/j.comcom.2015.11.003

search strategy instead of probability-based strategy during the path generation, vaccination, and free mutation stages improves the controllability and practicability of the algorithm because of the pseudo-randomness, ergodicity and determinacy of chaotic sequences. On the LATAX and ITNA networks, the outstanding optimization performance of CCIRA is proved by the simulation results. The risk control performance of the combination of the integrated routing risk model and CCIRA is also proved to be superior by the comparison results with the other risk-aware routing algorithm. This paper conducted an exploratory study on the feasibility of using chaotic search strategy instead of probability-based strategy in evolutionary algorithms for routing problems, and can be used as a reference for future research on improving the controllability of evolutionary algorithms for routing problems.

Acknowledgment

This work was supported by the National High Technology Research and Development Program of China (Grant no. 2014AA01A701) and the Fundamental Research Funds for the Central Universities (Grant no. 2014XS01).

References

[1] P. ChołDa, B. Helvik, Editorial: reliable network-based services, Comput. Commun 36 (6) (2013) 607–610.
[2] S. Blake, D. Black, M. Carlson, E. Davies, Z. Wang, W. Weiss, Architecture for differentiated services, RFC 2475 (Dec. 1998).
[3] P. Chołda, Risk-aware design and management of resilient networks, in: Proceedings of the 9th International Conference on Availability, Reliability and Security (ARES), Fribourg, Switzerland, 2014, pp. 468–475.
[4] F. Dikbiyik, M. Tornatore, B. Mukherjee, Minimizing the risk from disaster failures in optical backbone networks, J. Lightwave Technol. 32 (18) (2014) 3175–3183.
[5] S. Yuan, B. Wang, Highly available path routing in mesh networks under multiple link failures, IEEE Trans. Reliab. 60 (4) (2011) 823–832.
[6] M. Xia, M. Tornatore, C. Martel, B. Mukherjee, Risk-aware provisioning for optical WDM mesh networks, IEEE/ACM Trans. Netw. 19 (3) (2011) 921–931.
[7] B. Vidalenc, L. Noirie, L. Ciavaglia, E. Renault, Dynamic risk-aware routing for OSPF networks, in: Proceedings of the 2013 IFIP/IEEE International Symposium on Integrated Network Management, Ghent, Belgium, 2013, pp. 226–234.
[8] S.W. Jeon, K. Jung, H. Chang, Fully distributed algorithms for minimum delay routing under heavy traffic, IEEE Trans. Mob. Comput 13 (5) (2014) 1048–1060.
[9] W. Cai, H. Yang, F. Xiong, J. Li, J. Liu, Z. Zhao, K. Liu, An optimized service routing allocation method for electric power communication network considering reliability, Power Syst. Technol. 37 (12) (2013) 3541–3545 (in Chinese).
[10] Q. Zeng, X. Qiu, S. Guo, F. Qi, L. Meng, Risk balancing based routing mechanism for power communication service, J. Electron. Inf. Technol. 35 (6) (2013) 1318–1324 (in Chinese).
[11] K. Deb, A. Pratap, S. Agarwal, T. Meyarivan, A fast and elitist multiobjective genetic algorithm: NSGA-II, IEEE Trans. Evolut. Comput. 6 (2) (2002) 182–197.
[12] H. Dai, H. Qu, J. Zhao, QoS routing algorithm with multi-dimensions for overlay networks, China Commun 10 (10) (2013) 167–176.
[13] M. Ghatee, QoS-based cooperative algorithm for integral multi-commodity flow problem, Comput. Commun 34 (7) (2011) 835–846.
[14] X. Liu, F. Li, B. Zheng, Multi-constrained QoS routing algorithm based on quantum genetic algorithm, J. Nanjing Univ. Posts Telecommun. (Nat. Sci.) 31 (2) (2011) 31–35 (in Chinese).
[15] H. Xing, Xi. Liu, X. Jin, L. Bai, Y. Ji, A multi-granularity evolution based quantum genetic algorithm for QoS multicast routing problem in WDM networks, Comput. Commun 32 (2) (2009) 386–393.
[16] S. Fang, E. Zou, J. Xin, J. Lin, L. Lin, New chaos genetic algorithm applied in multi-constrained QoS routing, Appl. Res. Comput. 29 (8) (2012) 3078–3080 (in Chinese).
[17] L. Zhu, Z. Li, J. Xiang, Y. Cheng, An immune genetic routing algorithm for mesh network with QoS constraints, in: Proceedings of Third IEEE International Conference on Wireless Communications, Networking and Mobile Computing, Shanghai, China, 2007, pp. 1701–1704.
[18] H. Yetgin, K.T.K. Cheung, L. Hanzo, Multi-objective routing optimization using evolutionary algorithms, in: Proceedings of IEEE Wireless Communications and Networking Conference 2012, Paris, France, 2012, pp. 3030–3034.
[19] M. Cameo, C. Omaña, H. Castro, QoS routing algorithm based on multi-objective optimization for wireless mesh networks, in: Proceedings of IEEE Latin-American Conference on Communications 2010, Bogota, Colombia, 2010, pp. 1–6.
[20] A.H. Abdullah, R. Enayatifar, M. Lee, A hybrid genetic algorithm and chaotic function model for image encryption, AEU-Int, J. Electron. Commun. 66 (10) (2012) 806–816.
[21] Z. Chai, L. Chen, S. Zhu, Parameter optimization of cognitive engine based on chaos multi-objective immune algorithm, Acta Phys. Sin 61 (5) (2012) 058801 (in Chinese).
[22] Z. Chai, L. Zheng, S. Zhu, Chaotic immune optimization based resource allocation in cognitive radio network, Acta Phys. Sin. 61 (11) (2012) 118801 (in Chinese).
[23] B. Fan, L. Tang, Vulnerability analysis of power communication network, Proc. CSEE 34 (7) (2014) 1191–1197 (in Chinese).
[24] L. Hanzo, R. Maunder, J. Wang, L. Yang, Near-Capacity Variable-Length Coding: Regular and EXIT-Chart-Aided Irregular Designs, Wiley-IEEE Press, 2011, pp. 36–37.
[25] J. Ou, Theory of portfolio and risk based on incremental entropy, J Risk Financ. 6 (1) (2005) 31–39.
[26] Y. Jiang, S. He, X. Li, A maximum entropy model for large-scale portfolio optimization, in: Proceedings of International Conference on Risk Management & Engineering Management 2008, Beijing, China, 2008, pp. 610–615.
[27] F. Xing, L. Junzhou, W. Jieyi, G. Guanqun, QoS routing based on genetic algorithm, Comput. Commun 22 (15-16) (1999) 1392–1399.
[28] Z. Wang, Z. Chen, Z. Yuan, QoS routing optimization strategy using genetic algorithm in optical fiber communication networks, J. Comput. Sci. Technol 19 (2) (2004) 213–217.
[29] C. Ahn, R. Ramakrishna, A genetic algorithm for shortest path routing problem and the sizing of populations, IEEE Trans. Evolut. Comput 6 (6) (2002) 566–579.
[30] L. Jiao, L. Wang, A novel genetic algorithm based on immunity, IEEE Trans. Syst. Man Cybern. A 30 (5) (2000) 552–561.
[31] J. Li, C. Yang, J. Chen, Star-block design in two-level survivable optical networks, IEEE/ACM Trans. Netw 19 (2) (2011) 526–539.
[32] The power secondary system part of the 12th five-year plan of Guangdong power grid corporation, China Guangdong Power Grid Corp. (2010).
[33] B. Fan, L. Tang, A new five-dimensional hyperchaotic system and its application in DS-CDMA, in: Proceedings of 9th International Conference on Fuzzy Systems and Knowledge Discovery, Sichuan, China, 2012, pp. 2069–2073.
[34] ISO 31000 Risk Management-Principles and Guidelines, (Nov. 2009), Available from: <http://www.iso.org/iso/iso31000> (accessed 15.12.2014).