Indian Journal of Fundamental and Applied Life Sciences ISSN: 2231– 6345 (Online)
An Open Access, Online International Journal Available at www.cibtech.org/sp.ed/jls/2015/02/jls.htm
2015 Vol. 5 (S2), pp. 2766-2776/Rad et al.
Review Article

FAULT IDENTIFICATION USING END-TO-END DATA BY IMPERIALIST COMPETITIVE ALGORITHM

Farhad Rad, *Seyed Reza Hosseinpour and Hamid Parvin
Department of Computer Engineering, College of Engineering, Yasooj Branch, Islamic Azad University, Yasooj, Iran
**Author for Correspondence*

ABSTRACT
Faults in computer networks may result in millions of dollars in cost. Faults in a network need to be localized and repaired to keep the health of the network. Fault management systems are used to keep today's complex networks running without significant cost, either by using active techniques or passive techniques. In this paper, we propose a novel approach based on imperialist competitive algorithm using passive techniques to localize faults in computer networks. The presented approach using end-to-end data detect that there are faults on the network, and then we use imperialist competitive algorithm (ICA) to localize faults on the network. The aim of proposed approach is to minimize the cost of localization of faults in the network. According to simulation results, our algorithm is better than other state-of-the-art approaches that localize and repair all faults in a network.

Keywords: *Fault Management System, Imperialist Competitive Algorithm, Normalized Testing Cost, End-to-End Data*

INTRODUCTION
As computer networks continue to grow in size and complexity, effective network management is expected to become even more crucially important and more challenging. Simply stated, the aim of a typical network management system is to monitor the managed system and to ensure that it is running as smoothly as possible (Balaji *et al.*, 2012; Batsakis *et al.*, 2005). In order for the management systems to successfully manage the network a large amount of diagnostic information needs to be obtained and processed. As such, fault management systems can be divided into two groups: (1) active techniques, and (2) passive techniques (Bing *et al.*, 2012; Garshasbi and Jamali, 2014). Active techniques use probing technology so that the managed network can be tested periodically and suspected malfunctioning nodes can be effectively identified and isolated (Lin *et al.*, 2009). However, the diagnosing probes introduce extra management traffic, but it is exactly for localization of faults. Passive techniques use end-to-end data for fault localization in a network. In these techniques, introduces no additional traffic on the network (Bing *et al.*, 2012; Garshasbi and Jamali, 2014). On the other hand, it poses the challenge of fault inference accurate inference from end-to-end data is not always possible because end-to-end measurements can have inherent ambiguity (Balaji *et al.*, 2012; Bing *et al.*, 2012; Garshasbi and Jamali, 2014; Lin *et al.*, 2009; Lu *et al.*, 2013; Małgorzata and Adarshpal, 2004; Salah *et al.*, 2013).
In recent researches, a lot of fault identification algorithms have been proposed, such as Barford *et al.*, (2009) introduced an algorithm based on active techniques. The algorithm sends a number of packets for fault detection on the part of network at any time so that reducing traffic additional overhead in detection step. Zhao *et al.*, (2013) presented an active technique scheduling each network path measurement in multistage; further, the proposed approach checks, and monitors some network links per step. It causes reduced additional traffic as well as reduced testing costs. Bing *et al.*, (2012) proposed an inactive heuristic algorithm based on passive techniques. This approach chooses to test the most-used shared components in different paths; therefore, it prevents transferring additional traffic, but also minimizes the test cost. Lu *et al.*, (2013) introduced an algorithm based on active techniques that divides fault localization process into multiple steps merely runs localization operations on small part of network through using little number of additional packets each step. This fault localization approach also tries to have less destructive effect on traffic. Patrick *et al.*, (2007) offered several heuristic algorithms for

selecting candidate components to be tested that minimizes test cost. Hung and Patrick (2006) proposed a passive-based approach that locates faults in computing networks through following Bayesian principles. Garshasbi and Jamali (2014) proposed an approach for fault localization by genetic algorithm. This approach is a passive technique that localized faulty components in a network with minimum test cost.

Fault localization techniques must be designed and implemented to localize faults in the network with minimum cost test and have not a negative impact on network traffic. So we proposed a passive-base algorithm using imperialist competitive algorithm for fault localization in computer networks. The presented approach using end-to-end data detect that there are faults on the network, and then we use imperialist competitive algorithm to localize faults on the network. The contents of this article are organized as follows. Network modeling is in Section 2. Section 3 shows proposed approach in detail. The results are analyzed in section 4 and section 5 presents the concluding results.

Problem Setting and Assumptions

Generally, computer networks are composed of nodes and links (Patrick *et al.,* 2007; Yijiao *et al.,* 2005). Therefore, the network components can be represented as a tree or a graph. Figure (1) shows the network that is composed of several components (links, routers and etc). The failure of any of these components can disrupt the communication between the client and one (or both) of the servers (Patrick *et al.*, 2007). For fault localization in a computer network, we need to make a physical network into logical form. We first map each potentially faulty physical component (topology) to a *logical topology*. For example, we can transform the physical topology in Figure 1 to a logical tree as shown in Figure 2.

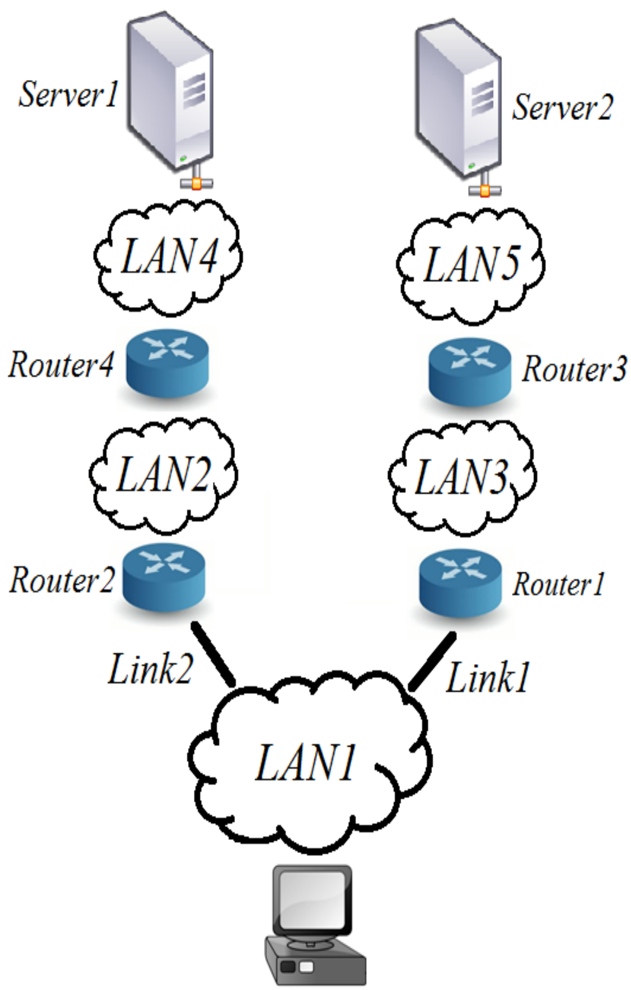

Figure 1: An example of a network consisting of different components

Review Article

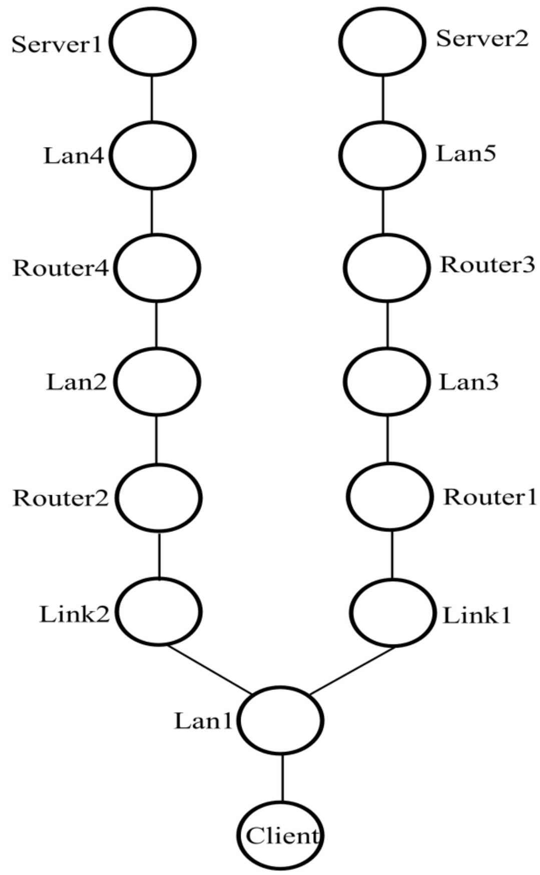

Figure 2: Logical topology corresponding to Figure 1

We assume can recognize the existence faults according to end-to-end data on the network. The amount of end-to-end data can be used to detect faults in the network: insufficient amount of data indicates faults, while sufficient amount of data indicates that the network is operating normally. The status of a component (i.e., whether faulty or not) can be tested to determine status it. So, test of each component in network have costs (Bing *et al.*, 2012). Therefore choice candid component for test is very important to reduce test costs (Garshasbi and Jamali, 2014).

When the known network has abnormal behavior, therefore are faulty components in the network and the network has been disrupted. But the main problem is that we do not know exactly which component (or components) are faulty (Garshasbi and Jamali, 2014). So should be tested network components to find the exact location of faulty components of the network. But cannot test all the network components, because testing of components has costs. So it should be a minimum number of tests to identify the exact location of the faults to reduce costs testing. The main problem in this paper is to minimize the cost of testing. Therefore, we have chosen to test the node that has the highest probability of being faulty. This problem is NP-Complete (Bing *et al.,* 2012; Garshasbi and Jamali, 2014).

In this paper, problem setting and assumptions is as follow:

- We consider logical topology of network.
- We assume can recognize the faults according to end-to-end data on the network.
- The status of a component is faulty or not.
- Test of each component in the networks has cost.
- We consider persistent faults in the network.
- The routing path from a source (client) to the server can be static or dynamic.

- We only know probabilistic path information

Imperialist Competitive Algorithm

In computer science, Imperialist Competitive Algorithm (ICA) is a computational method that is used to solve optimization problems of different types. Like most of the methods in the area of evolutionary computation, ICA does not need the gradient of the function in its optimization process. From a specific point of view, ICA can be thought of as the social counterpart of genetic algorithms (GAs). ICA is the mathematical model and the computer simulation of human social evolution, while GAs are based on the biological evolution of species (Atashpaz-Gargari, 2007).

Figure 1 shows the flowchart of the Imperialist Competitive Algorithm. This algorithm starts by generating a set of candidate random solutions in the search space of the optimization problem. The generated random points are called the initial Countries. Countries in this algorithm are the counterpart of Chromosomes in GAs and Particles in Particle Swarm Optimization (PSO) and it is an array of values of a candidate solution of optimization problem. The cost function of the optimization problem determines the power of each country. Based on their power, some of the best initial countries (the countries with the least cost function value), become Imperialists and start taking control of other countries (called colonies) and form the initial Empires (Atashpaz-Gargari, 2007).

Two main operators of this algorithm are Assimilation and Revolution. Assimilation makes the colonies of each empire get closer to the imperialist state in the space of socio-political characteristics (optimization search space). Revolution brings about sudden random changes in the position of some of the countries in the search space. During assimilation and revolution a colony might reach a better position and has the chance to take the control of the entire empire and replace the current imperialist state of the empire. Imperialistic Competition is another part of this algorithm. All the empires try to win this game and take possession of colonies of other empires. In each step of the algorithm, based on their power, all the empires have a chance to take control of one or more of the colonies of the weakest empire.

Algorithm continues with the mentioned steps (Assimilation, Revolution, Competition) until a stop condition is satisfied (Atashpaz-Gargari, 2007).

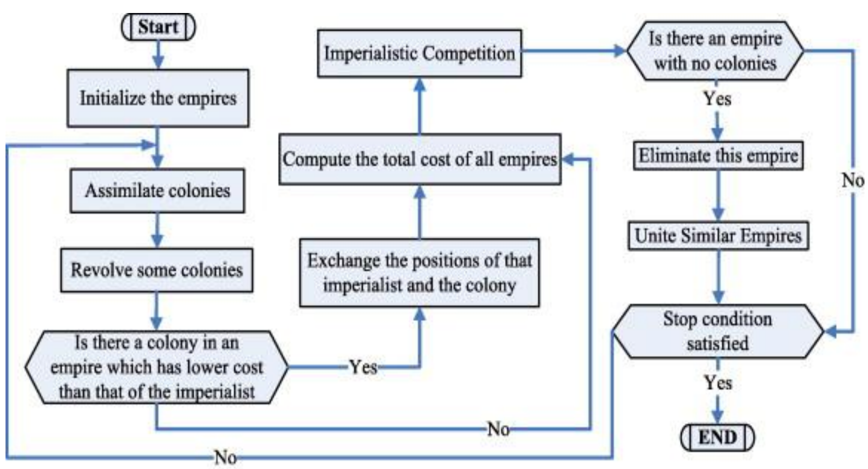

Figure 3: Flowchart of ICA (Atashpaz-Gargari, 2007)

Proposed Algorithm

Fault detection is the first step of network fault management (not fault localization). In other words, first step of network fault management is fault detection, and then second step is fault localization. In this paper, we used end-to-end data for fault detection in first step. In fact, if the data is sent from the source to server do not fit, then we can recognize that the network is functioning abnormal. In second step the exact location of faults in the network must diagnosed and repaired them. Therefore, we introduced an algorithm based on the imperialist competitive algorithm for finding the exact location of faulty components.

Fault Localization by Imperialist Competitive Algorithm

Countries Encoding

We use binary coding for represent countries. In fact, each country represents a path, and each element of the country refers to a node. Each element can have two values, 0 or 1.If the element value of country is 1, the node corresponding to the element of the path do not uses for data transfer or routing. If the element value of country is 0, the node corresponding to the element uses in routing. Length of each country is equal to the total number of nodes in the network. Table 1 shows an example of the paths. According to table 1 the total number of nodes are three, so clients these paths for transmit data to the server. According to table 1 can be produced three countries. Figure 4shows binary representation of table 1.

Table 1: Paths from client to server

Path 1	**Node 1**	**Node 2**	**Node 3**
Path 1	Node 2	-	-
Path 1	Node 1	Node 2	-

Figure 4: Binary representation of table 1

	Node 1	**Node 2**	**Node 3**
Country 1	0	0	0
Country 2	1	0	1
Country 3	0	0	1

In the example of Figure 4, the *path 1* uses of the three nodes for data transmission. The *path 2* uses of one node and *path 3* uses of two nodes for data transfer to server (or servers). We consider bad or faulty paths as an initial countries.

Evaluation Fitness of Countries

In this paper, we use two parameters to evaluate fitness of countries:

- The number of nodes sharing the different paths (degree of node)
- The number of nodes used in a path

First, we calculate the weight of each node using equation (1). N is number of total countries. W_i is weight of $node_i$. n_{ik} shows amount of each $element_i$ in $country_i$. We use equation (2) for calculate the number of nodes used in a path. n_i is the $element_i$ in the $country_i$. N is number of total countries.

(1)

(2)

According to the above description, fitness function to evaluate the fitness of each country is based on equation (3). α is a value between 0 and 1 (α = [0,1]). If α = 1, then the evaluation is based on the number of nodes used in a path. If α = 0, then the evaluation is based on degree of node. α can have a value between 1 and 0, this value determines importance of W_i, and importance of A. More value of the f_i corresponds to a better fitness value for the country.

(3)

There is an exception in evaluating fitness of countries that if the all elements if country have value of 1, therefore fitness country will be equal to zero (*fi = 0*).

Empires Formation

To start the optimisation algorithm, initial countries of size $N_{Country}$is produced ($N_{Country}$ = number of bad paths). We select N_{imp} of the most powerful countries (countries with high fitness) to form the empires.

Review Article

The remaining N_{col} of the initial countries will be the colonies each of which belongs to an empire. To form the initial empires, the colonies are divided among imperialists based on their fitness.The colonies are randomly chosen and given to the n^{th} imperialist. For calculate power of empires uses equation (4). Where P_n is the total cost of empire and α is a positive number that is between 0 and 1, and f shows fitness.

(4)

Movement of Colonies toward the Imperialist

To define a policy of assimilation, we use the same operator with crossover operator in genetic algorithm. Each country is subjected to crossover with probability P_c. One country is selected from the population, and a random number (RN = [0,1]) is generated for it. If $RN<P_c$, these country is subjected to the crossover operation with empire using single point crossover. Otherwise, these countries are not changed. The pseudo code of the crossover function is as follows.

1. *Select one country*
2. *Let RN a random real number between 0 and 1*
3. *If RN < 0:5 /* operators probability*
4. *Crossover (country, empire)*

Revolution

Revolution is a fundamental change in power that takes place in a relatively short period of time. In the terminology of ICA, revolution causes a country to suddenly change its socio-political characteristics. That is, instead of being assimilated by an imperialist, the colony randomly changes its position in the socio-political axis (Atashpaz-Gargari, 2007).

While moving toward the imperialist, a colony might reach to a position with lower fitness than the imperialist. In this case, the imperialist and the colony change their positions. Then the algorithm will continue by the imperialist in the new position and the colonies will be assimilated by the imperialist in its new position.

Imperialistic Competition

All empires try to take the possession of colonies of other empires and control them. The imperialistic competition gradually brings about a decrease in the power of weaker empires and an increase in the power of more powerful ones. The imperialistic competition is modelled by just picking some (usually one) of the weakest colonies of the weakest empire and making a competition among all empires to possess these (this) colonies (Atashpaz-Gargari, 2007).

To start the competition, first a colony of the weakest empire is chosen and then the possession probability of each empire is found. The possession probability P_P is proportionate to the total power of the empire. The process of selecting an empire is similar to the roulette wheel process which is used in selecting parents in GA.

The Final Condition and Select Appropriate Node for Testing

If only one element value of each country is 0, and the rest of the elements values are 1, therefore the algorithm should be terminate. Element whose value is 0, node corresponding to the element tested. After the test node, if the node is faulty, the node is repaired else the algorithm is repeated. Figure 5 shows the flowchart of the proposed method. Proposed algorithm will be repeated until all faulty nodes are localized.

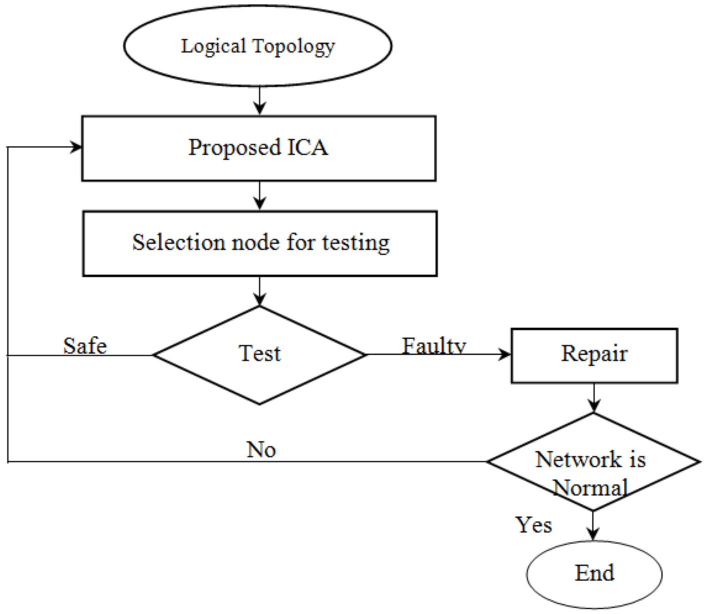

Figure 5: Flowchart of proposed algorithm

Evaluation of Simulation Results

We evaluate the performance of our algorithm through extensive simulation (using a MATLAB) in a general network. The proposed method compared with the methods in (Balaji *et al.,* 2012; Hung and Patrick, 2006; Yijiao *et al.*, 2005). Several scenarios have been considered for simulations. In general, three different scenarios have been considered for simulation that shows in table 2. The performance metrics we use are the normalized testing cost, false positive and total test cost.

Table 2: Scenarios for simulations

Scenarios	**Number of Nodes**	**Number of Clients**
Scenario 1	60	40
Scenario 2	60	80
Scenario 3	150	80

Figures 6-8show the results of total test cost for algorithms in three scenario. The results of proposed algorithm, Greedy, Ordering and NFDM are plotted in the figures 6-8. We observe that proposed algorithm has better total test cost compared to other algorithms.

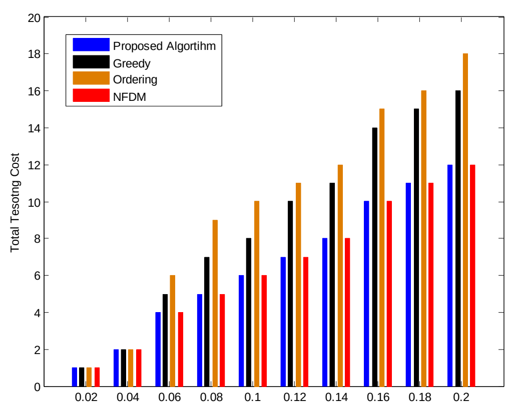

Figure 6: Total test cost in scenario 1

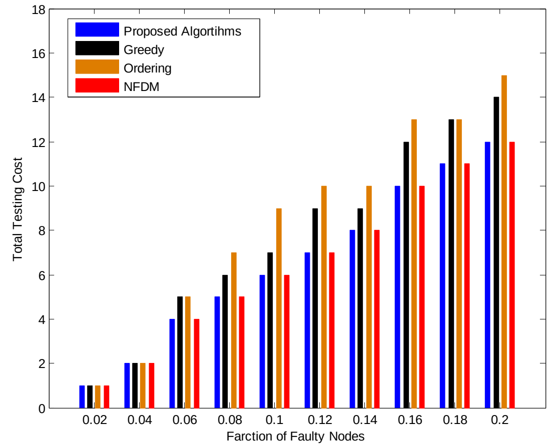

Figure 7: Total test cost in scenario 2

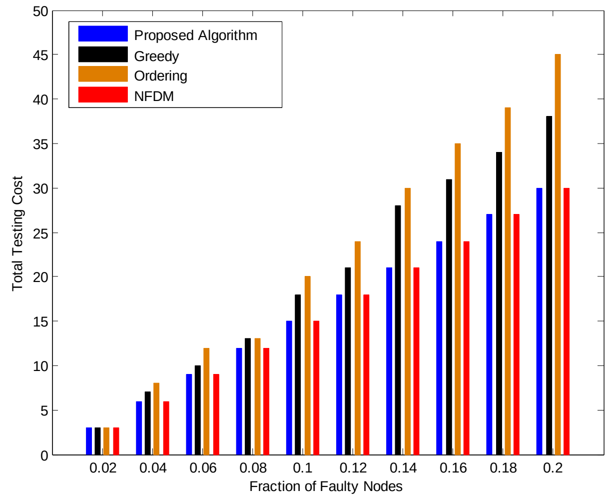

Figure 8: Total test cost in scenario 3

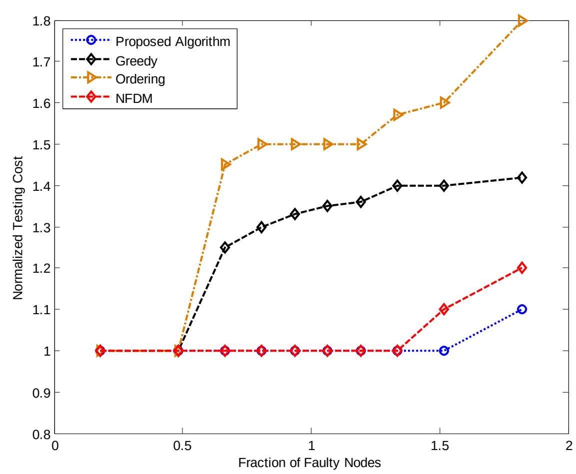

Figure 9: Normalized testing cost in scenario 1

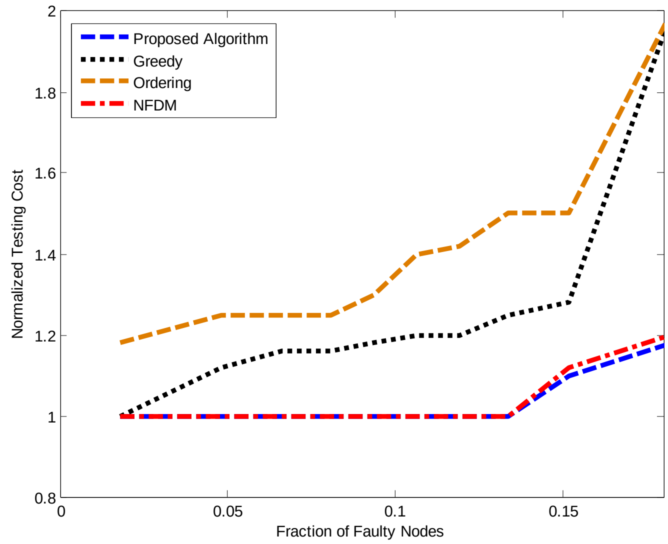

Figure 10: Normalized testing cost in scenario 2

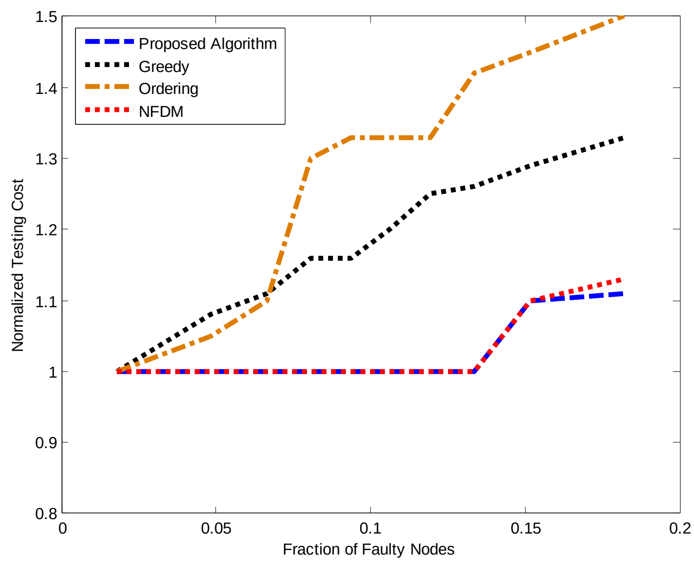

Figure 11: Normalized testing cost in scenario 3

Figures 9-11 show the results of normalized testing cost for algorithms in three scenario. According to results show in these figures, we observe that proposed algorithm has better normalized testing cost compared to other algorithms, and it has minimum normalized testing cost compared to other three algorithms.

Figures 12-14 show the total test cost by applying proposed algorithm, Greedy, Ordering and NFDM for fault localization in scenarios 1-3 respectively. According to these figures total test cost of our algorithm is minimum compared to other algorithms. According to simulation results, our approach in addition to reducing the cost of fault localization, it provides good false positive compared to similar methods. Obtained results in large and small scales (different scenarios) indicate that our proposed algorithm can provide similar results in different scales and scenarios, and proves the robustness of the proposed method in different scales.

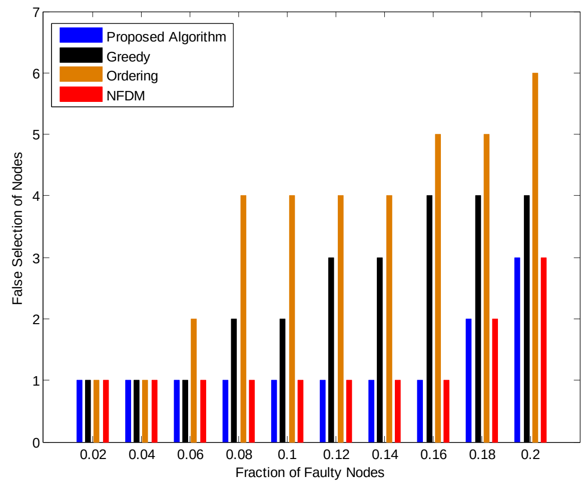

Figure 12: Total test cost in scenario 1

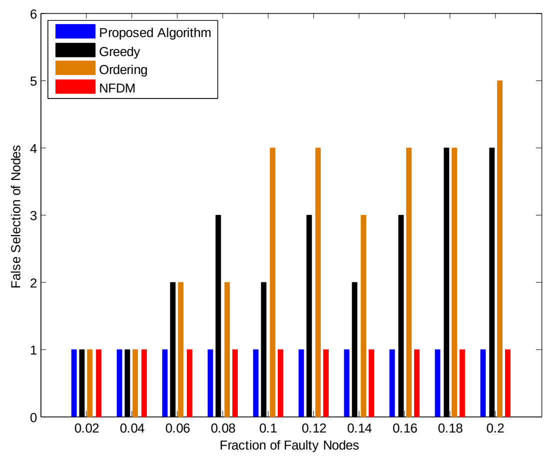

Figure 13: Total test cost in scenario 2

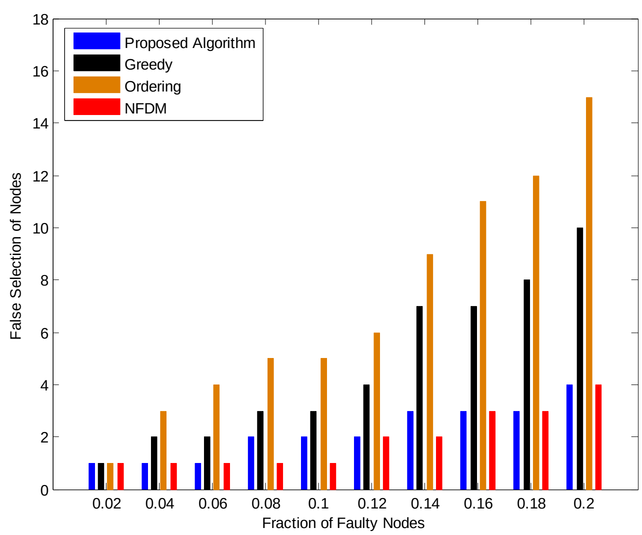

Figure 14: Total test cost in scenario 3

CONCLUSION

In this paper a new algorithm based on imperialist competitive algorithm was implemented for fault localization in computer networks. The presented approach used end-to-end data detected that there are faults on the network, and then used imperialist competitive algorithm (ICA) to localized faults on the network. Extensive simulation demonstrated that our algorithm is outperforms the other algorithm for fault localization in computer networks.
As future work, we are pursuing in the following two directions: 1) evaluating the performance of our approach under other scenarios, for instance, when the location of faulty components follows a more clustered distribution instead of uniform random distribution and 2) developing proposed algorithm for temporary fault localization.

REFERENCES

Atashpaz-Gargari (2007). Imperialist competitive algorithm: An algorithm for optimization inspired by imperialistic competition. *IEEE Congress on Evolutionary Computation*, CEC 2007, *25-28 Sept*.
Balaji P, Shakti K and Vinay KP (2012). Fault management of computer networks based on probe station and probe set selection Algorithms. *International Conference on Advances in Electronics. Qinhuangdao, China, June 25-27* 504-508.
Barford P, Duffield N, Amos R and Joel S (2009). Network performance anomaly detection and localization. *28th IEEE International Conference on Computer Communications, Rio de Janeiro, April 19-25* 1377-1385.
Batsakis A, Tanu M and Andreas T (2005). Practical passive lossy link inference. *Passive and Active Network Measurement* **431** 362-367.
Bing W, Wei W, Hieu D, Wei Z and Krishna R (2012). Fault localization using passive end-to-end measurements and sequential testing for wireless sensor networks. *IEEE Transactions on Mobile Computing* **11** 439-452.
Coates M and Nowak R (2000). Network loss inference using unicast end-to-end measurement. *Modeling and Management* **23** 19-28.
Duffield N (2003). Simple network performance tomography. *Proceedings of the 3rd ACM SIGCOMM Conference on Internet Measurement, New York, NY, USA* 210-215.
Garshasbi MS and Jamali Sh (2014). A New Fault Detection Method Using End-to-End Data and Sequential Testing for Computer Networks. *International Journal Information Technology and Computer Science* **01** 93-100.
Hung N and Patrick T (2006). Using End-to-End Data to Infer Lossy Links in Sensor Networks. *25th IEEE International Conference on Computer Communications Proceedings*.
Lin Y, Liang B and Li B (2009). Passive loss inference in wireless sensor networks based on network Coding. *IEEE Conference on INFOCOM, Rio de Janeiro, Apr 19-25* 1809-1817.
Lu L, Zhengguo X, Wenhai W and Youxian S (2013). A new fault detection method for computer networks. *Reliability Engineering and System Safety* **114** 45 –51.
Małgorzata S and Adarshpal S (2004). A survey of fault localization techniques in computer networks. *Science of Computer Programming* **53** 165–194.
Patrick L, Vishal M and Rubenstein D (2007). Toward optimal network fault correction via end-to-end inference. *26th IEEE International Conference on Computer Communications, New York, Oct 25-26* 165-172.
Salah S, Fernandez G and Jesus E (2013). A model-based survey of alert correlation techniques. *Computer Networks* **94** 116-143.
Yijiao Y, Qin L and Liansheng T (2005). A graph-based proactive fault identification approach in computer networks. *Computer Communications* **28** 366-378.
Zhao Y, Xiao F and Shengwei W (2013). An intelligent chiller fault detection and diagnosis methodology using Bayesian belief network. *Energy and Buildings* **57** 278–288.