

# Optimal Reliability in Cloud Computing Networks Under Dynamic Node Failure

Debashreet Das<sup>1</sup>

<sup>1</sup>School of Computer Science and Engineering, Vellore Institute of Technology, Katpadi, Vellore, 632014, Tamil nadu, India. debashreetdas12@gmail.com, m: +91 -7205671494, ORCID-ID: 0000-0002-2070-9477.

Pradyumna Kumar Tripathy<sup>2\*</sup>

<sup>2</sup>Department of Computer Science and Engineering, Silicon Institute of Technology, Patia, Bhubaneswar, 751024, Odisha, India. pradyumnatripathy@gmail.com, ORCID-ID: 0000-0002-9788-1013.

**\*Corresponding Author:** Pradyumna Kumar Tripathy

<sup>\*</sup>Department of Computer Science and Engineering, Silicon Institute of Technology, Patia, Bhubaneswar, 751024, Odisha, India. pradyumnatripathy@gmail.com, ORCID-ID: 0000-0002-9788-1013.

## Abstract

With the rise in the adoption of the cloud across organizations, it is vital to deliver consumers on-demand services that meet their needs. As a result, services like reliability, security, and resource management become critical. Among these, the reliability aspect is crucial in defining the performance of a cloud computing network. However, as user expectations increase, it becomes essential to make cloud networks and services more dependable. Hence, A novel enumerative approach is proposed in this work in order to optimize the reliability of Cloud Computing Networks. Our primary objective is to optimize the cloud network to maximize network reliability during a dynamic node failure. Since reliability optimization under the dynamic node failure is a critical task, it leads to an NP-Hard problem. In addition, the results of our experiment for a few samples of cloud networks are presented and compared with the technique of fine-grained check pointing for providing reliability in cloud networks. The suggested methodology and is also applicable to large-scale cloud networks and found to be more efficient

**Keywords:** Cloud Computing, Enumerative Approach, Node failure, Optimization, Reliability

## 1 Introduction

The main objective of the research work is to optimize the reliability by taking into account the parameters like degree and diameter since these two parameters contribute in determining the reliability of a CCN. If the degree is more, then the CCN becomes more stable. Similarly, if the diameter is less, the data transfer in a CCN becomes faster, which leads to a highly reliable CCN. Hence, the motive of the proposed approach is to maximize the network reliability considering the dynamic node failure. Alongside the cloud computing systems, the distributed systems can take the form of grid computing, edge computing, peer-to-peer computing, etc.[1] for allowing resource sharing and the facilities for transferring the data. These systems provide opportunities to service providers of the cloud to establish new data centres as well as for the users of the cloud to use the cost-efficient cloud. The system of CCNs enables the users to either scale in or scale out the virtual resources dynamically as per the computing requirements without human interaction [2]. Moreover, depending on the service or location, the CCNs can be categorized as either private, public, hybrid, or community-based. The public clouds can be accessed by anyone and, thus, vulnerable to several attacks. However, a private cloud is available to only

a specific organization or user. However, the public cloud is found to be more cost-effective than the private cloud. A hybrid cloud is a mix of private and public clouds used by organizations sharing a common objective or interest.

The paper is organized into the following Sections: Section 2 presents the Literature Survey till date. The notations, assumptions, and the proposed approach followed by an Illustration is described in Section 3. Section 4 discusses the result and the paper is concluded in Section 5.

## 2 Related Work

Although various studies have been conducted on analyzing reliability in the context of Cloud Computing Networks, none of the approaches has considered the diameter and degree parameters for computing reliability. Regardless, F.Zhang [3] proposed a method based on intelligent cloud computing for analyzing reliability. The author considered the reliability analysis matrix for computer networks as well as the failure rate of the networks with large analysis error rates. In addition, M. R. Mesbahi et.al. [4] conducted an in-depth study of high availability and reliability in CCNs, considering the failure rate of the entire cloud system. The authors in [5] not only considered the

"when" and "how" techniques for improving availability and reliability but also the characteristics of the cloud computing system. Besides, the authors in [6] focused on providing a reliable approach to resource allocation for cloud computing while minimizing the cost.

However, from the perspective of resilience and reliability, given that network reliability evaluation problems and the traditional assignment have been shown to be NP-hard [7], the addressed problem is also NP-hard. One solution to this problem is to use a soft computing algorithm. According to related research, the genetic algorithm (GA) [8] is found to be one of the popular solutions to the assignment of resources or optimization problems pertaining to network reliability due to high diversification in the search capability.

After a thorough literature study, it was found that although many of the existing approaches have addressed the reliability issues in CCN. However, none of the techniques considered the two essential factors, such as diameter and degree, for optimizing the network reliability. Hence, this article proposes an algorithm OptiCCN for optimizing the network reliability in the CCN considering degree and diameter under dynamic node failures.

### 3 Proposed Approach

This section proposes a new and efficient algorithm for the reliability optimization of cloud computing networks considering the dynamic node failure.

#### 3.1 Notations

The notations used in the proposed approach are presented here:

- $G$  : Given graph in the form of adjacency matrix
- $f_n$  : Faulty Node
- $G'$  : Graph after removing the defective node and its associated links.
- $A_l$  : Available links
- $K$  : Incrementer
- $DS$  :Diameter Set for storing all the possible combinations

- $e_p$  :Edge probability
- $ST_v$  : $v^{\text{th}}$  Spanning Tree of the graph  $G$

#### 3.2 Problem Statement

In a distributed environment like the cloud, ensuring the delivery of quality of services becomes a challenge. There may be several factors responsible for QoS delivery such as load balancing, reliability, security, etc. out of which reliability is considered as one of the crucial factors in a cloud environment. In such a case, there is always a chance of node failure. In addition, there can be single node failure or multi-node failures. However, the links connecting the faulty nodes will be reconnected with the non-faulty nodes in order to make the network operational. So, the objective is to design an optimum reliable Cloud Computing Network (CCN) considering the dynamic node failure.

##### 3.2.1 Mathematical Model for Reliability Computation of CCN

In this subsection, a mathematical formulation has been done for evaluating the reliability of the CCN. Here, we have considered only the link failures as the probability by generating the spanning trees from the given graph and hence, measuring their probability. The reliability of Cloud Computing Network can be expressed as :

$$R = \begin{cases} \left( 1 - \left( \prod_{e \in ST_0} (1 - e_p) \right) \right), v = 0 \\ \sum_{ST_v=1,2,3} \left\{ \left( 1 - \left( \prod_{e \in ST_0} (1 - e_p) \right) \right) \prod_{e \in (ST_{v-1} \sim ST_v)} (1 - e_p) \right\}, v > 0 \end{cases} \quad (1)$$

Where  $ST_0$  is the initial spanning trees. All generated spanning trees of CCN are enumerated and then disjointed to get the reliability value. The proposed approach for optimization of CCN uses equation 1 for the reliability evaluation of CCN [8].

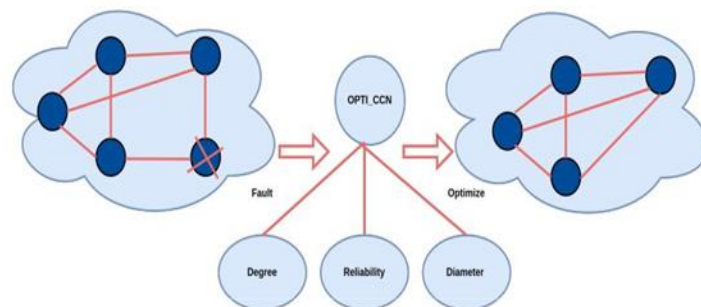


Fig. 1: The CCN model considered for our problem

However, we consider the above CCN model in Figure 1 for our approach.

### 3.3 Proposed Algorithm

The cloud computing network (CCN) can be considered as a probabilistic graph (G) where the nodes represent the processing elements, and the edges between the nodes represent the links. For

simplicity, the reliability of each link is considered the same as 0.9. However, the proposed algorithm works fine for any real probabilistic value between 0 and 1.

The algorithm 1 Opti CCN finds the CCN with optimum reliability considering the faulty node-set  $f_n$ .

#### Algorithm 1 Opti\_CCN

Input: G,  $f_n$

Output: Optimized Network  $G'$

```
1: Opti CCN(G,  $f_n$ ) //  $G'$  is the graph generated by removing the faulty node(s) and their corresponding links from G.
2:  $G' = G - f_n$ 
3:  $K = 0$ 
4:  $A_i = \text{Adj}(f_n)$ 
5: For all  $N_i \in G'$  // For  $i = 1, 2, 3, \dots, N$ 
6: if  $\text{degree}(N_i) \leq 1$  then
7:  $K = K + 1$ 
8:  $\text{Arr}[i] = N_i$ 
9: end if
10: if  $K == 2$  then
11: Connect  $\text{Arr}[1]$  with  $\text{Arr}[2]$ 
12:  $A_i = A_i - 1$ 
13: end if
14: if  $K \neq 2$  then
15: Find all combinations of the elements of Arr with respect to the number of available links
16: Find the diameter of each combination of elements in K
17: if  $A_i > 0$  then
18: Connect the links with the nodes that produce the highest diameter values
19: end if
20: end if
21: Produce the diameter matrix for each node to every other node in  $G'$ 
22:  $DS = \text{maximum diameter}(N_i, N_j) \forall i, j \in G'$  // Compute the maximum diameter from node i to node j
23: if  $DS > 1$  then
24: Find the reliability of each combination of elements in Join the link with the nodes which produces the highest network reliability
25: else
26: Join the links associated with the nodes in the DS
27: end if
28: Calculate the reliability of  $G'$ 
```

#### Explanation of the Algorithm:

The input to our algorithm is a network considered as a fully connected graph with 'V' no. of vertices and 'E' no. of edges. In this example, the authors considered only a single node failure. However, the proposed algorithm can also work well for multiple node failures. After identifying the faulty node, it is removed, and a new graph  $G''$  is generated. An incrementor keeps track of the number of nodes having degrees 0 or 1. The initial value of the incrementor is set to 0. The links associated with the faulty node are removed along with the faulty node and stored in the array " $A_i$ ". The diameter of the rest of the other nodes that are in the new graph  $G''$  are computed. If the degree or diameter of any

node is found to be less than 1, then the value of incrementor is increased by 1 and all those nodes are stored in the array  $\text{Arr}[i]$ . If there exist only two nodes having degrees less than equal to 1 (for  $K=2$ ), they may be joined through the available link. For the case when  $K > 2$ , all possible combinations of the node-set in Arr is generated, and the corresponding diameters are computed and stored. If some available links exist, then connect the links with the nodes with the highest diameter. A diameter matrix is formed by computing the diameter from every node to every other node in  $G''$ . The nodes with the maximum diameter (there may be one node pair producing maximum diameter) are stored in the diameter set DS. Suppose more than one node pair

exists in the DS. In that case, each node pair's reliability is computed using equation 1, and the node pair producing the highest net-work reliability are connected through the available link. Otherwise, join the links associated with the nodes in the DS. Finally, the optimized reliability of the CCN is computed. The obtained CCN after the node(s) failure is now optimized considering the degree, diameter, and reliability.

#### 4 Results and Discussions

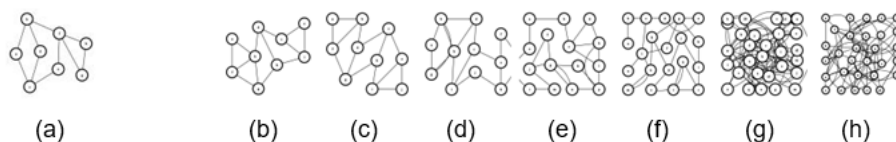
The following section describes the results after implementing the proposed algorithm on some of the sample cloud networks, and the results are recorded in Table 2. The algorithm is implemented

in MATLAB 2016b in Windows 10, a 64-bit operating system. The parameters considered for our experiment are mentioned in Table 1.

It is evident from Table 3 that as we increase the network's connectivity by increasing the number of links and nodes, the reliability also increases. The reliability column of Table 3 represents the initial reliability value of the net-work before node failure and prior to the optimization. It can be observed that the optimized output network always yields a better reliability value as compared to the input network. The links associated with the faulty nodes are re-configured so as to get a better degree, diameter, and reliability from the link arrangement of the remaining nodes.

**Table 1:** Parameters Setting for our Experiments

Parameters	Numbers
Minimum No. of Nodes	4
Maximum No. of Nodes	30
Degree	3
Diameter	2-4
Minimum No. of Links	5
Maximum No. of Links	20
Reliability	0.82-0.99



**Fig. 2: Input Networks**

**Table 2:** Implementation of the proposed algorithm on some sample cloud networks

#	Input Network (N,L)	Reliability	Failure Node	Output Optimized Network (N,L)	Optimized Reliability
1	Fig. 2a(8N,11L)	0.987	c	Fig. 3a(7N,11L)	0.9921
2	Fig. 2b(9N,13L)	0.9909	c	Fig. 3b(8N,13L)	0.9982
3	Fig. 2c(10N,14L)	0.9900	c	Fig. 3c(9N,14L)	0.9980
4	Fig. 2d(11N,16L)	0.9820	c	Fig. 3d(10N,16L)	0.9930
5	Fig. 2e(15N,20L)	0.9830	c	Fig. 3e(14N,20L)	0.9940
6	Fig. 2f(20N,24L)	0.9845	c	Fig. 3f(19N,16L)	0.9950
7	Fig. 2g(25N,39L)	0.9870	c	Fig. 3g(24N,39L)	0.9970
8	Fig. 2h(30N,60L)	0.9880	c	Fig. 3h(29N,60L)	0.9980

In Table 3, the parameters like the degree and diameter of the network are considered. It can be concluded that the degree remains almost the same before and after the optimization. Although the degree of the optimized network in row 1 of Table 3 gives a lower value as compared to its input value, its diameter and reliability are found to be better than the input network. As the proposed approach avoids parallel edges and self-loops, therefore, the network in row 1 with degree 3 cannot be further optimized. Since we have considered only a single node failure, the links connecting to the faulty node are adjusted, and we get the optimized network with a reduced diameter. However, the proposed algorithm can be applied for multi-node failures. It can be visible from Table IV that the optimized network always produces a lower diameter as compared to input CCN

#	Input Network (N,L)	Input Degree	Output Optimized Network (N,L)	Output Degree	Output Diameter
1	Fig. 2a(8N,11L)	3	Fig. 3a(7N,11L)	3	1
2	Fig. 2b(9N,13L)	3	Fig. 3b(8N,13L)	3	2
3	Fig. 2c(10N,14L)	3	Fig. 3c(9N,14L)	3	3
4	Fig. 2d(11N,16L)	4	Fig. 3d(10N,16L)	4	2
5	Fig. 2e(15N,20L)	4	Fig. 3e(14N,20L)	5	2
6	Fig. 2f(20N,24L)	4	Fig. 3f(19N,24L)	5	2
7	Fig. 2g(25N,39L)	4	Fig. 3g(24N,39L)	5	2
8	Fig. 2h(30N,60L)	3	Fig. 3h(29N,60L)	6	1

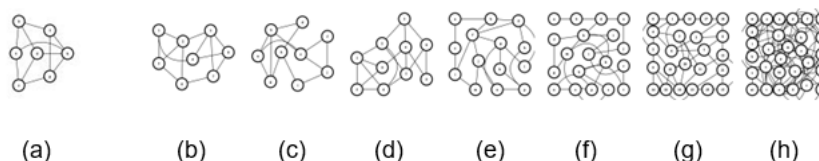


Fig. 3: Optimized Output Networks

A graph has been plotted in Fig. 4 showing the given reliability of the original network before the node's failure and the network's optimized reliability after the node's failure. It can clearly be seen from the graph that the proposed algorithm always produces a better reliability value after optimization under node(s) failure. The available links are re-used among the remaining nodes so as to produce a better degree, diameter, and maximum reliability of the CCN.

It can be concluded from Fig. 5 that with the increased number of nodes, the diameter of the

nodes remains uniform, leading to increased reliability. The graph compares the diameter of the nodes before and after optimization considering the failure of a single node. Since the parallel edges and loops are not taken into account, the diameter remains constant from node no. 4 to 8, after which it increases to its peak value by increasing slowly (from nodes 9 to 10) and again decreases after optimization (for node no. 11); but, before optimization, the diameter for nodes 9 to 11 remains constant.

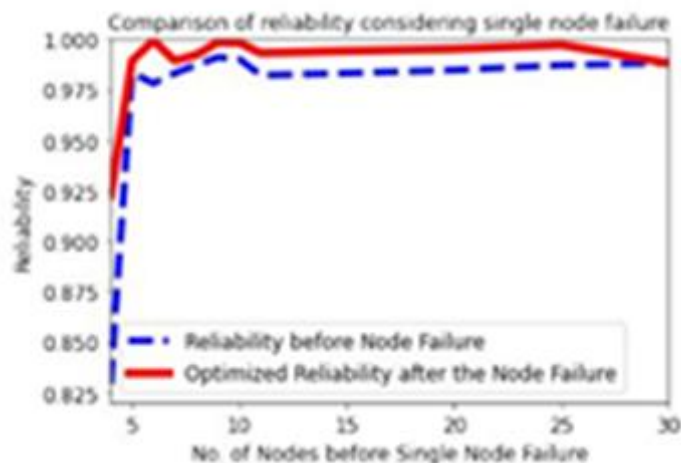


Fig. 4: Comparison of Reliability before and after optimization considering single node failure

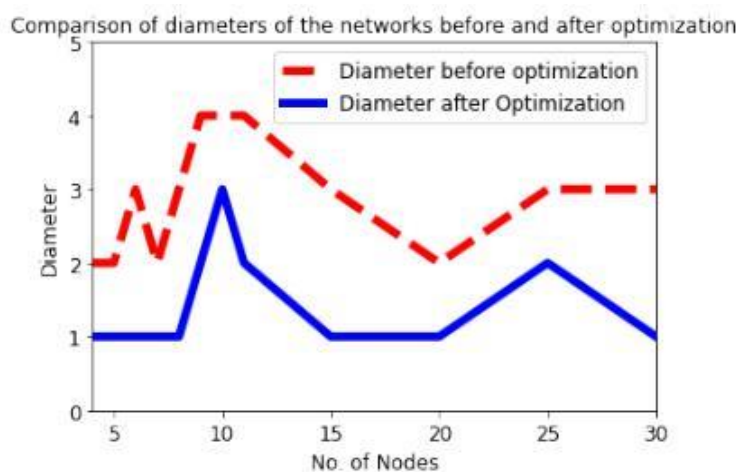


Fig. 5: Comparison of diameters of the networks before and after optimization

From Fig. 6, it can be observed that with the increase in the number of links, there is also an increase in reliability. This can be attributed to the

fact that increasing the links increases the network's connectivity. After optimization, the number of links gets adjusted and achieves optimized reliability.

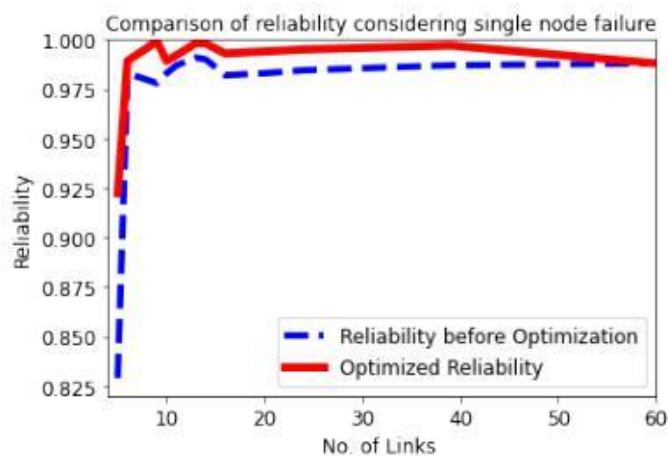


Fig. 6: Comparison of the number of links with reliability before and after optimization

It can be concluded from Fig. 7 that with the increased number of nodes, the degree of the nodes increases, which leads to increased reliability. The graph compares the degree of the nodes before and

after optimization considering the failure of a single node. The proposed approach restricts the inclusion of parallel edges and self-loops during optimization.

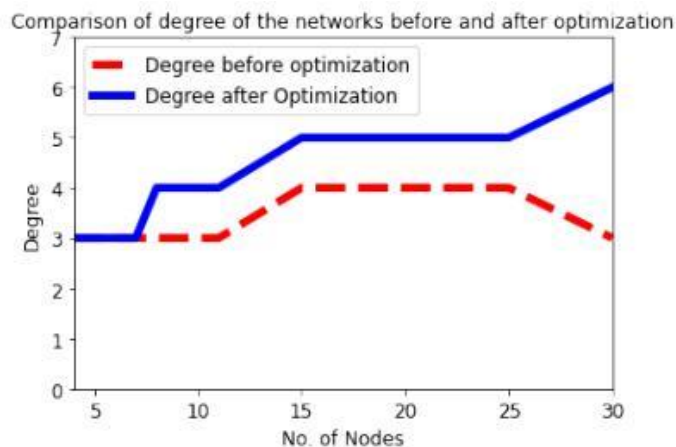


Fig. 7: Comparison of the degree of networks with number of nodes before and after optimization

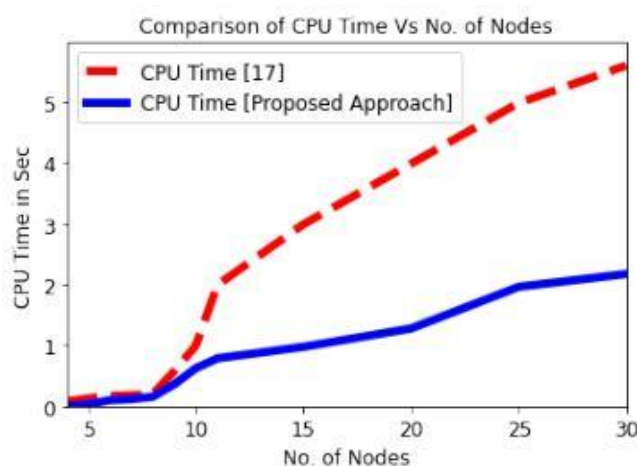


Fig. 8: Number of nodes vs CPU Time

It can be concluded from Fig. 8 that the CPU time or execution time for our proposed approach increases linearly compared to the execution time of the existing approach proposed in [9]. This can be attributed to the fact that as the number of nodes increases, our approach Opt CNN recursively computes the degree and diameter, thereby adjusting the links among the remaining nodes. The diameter and degree of the final optimum network get reduced, thereby giving the optimum reliability value.

## 5 Conclusions and Future Work

In this research work, a new algorithm Opti CCN is proposed for optimizing the Cloud Computing network by rearranging the links of the network in case of dynamic node failures. The Cloud Computing network is topologically optimized to give good connectivity, less diameter, and high reliability. The reliability of each intermediate network is computed using a mathematical formula until the final optimum network is obtained. The proposed approach can also be applicable for multi-node

failure scenarios. It is efficient enough to optimize the network in a reasonable amount of time. Although the experiments are carried out for CCNs with homogeneous link capacity. Practically, the proposed approach works well for CCNs with non-homogeneous link capacity as well.

## Declarations

Author Contributions Both the authors Dr. Debashreet Das and Dr. Pradyumna Kumar Tripathy have equally contributed to this research paper. Dr. Debashreet has prepared the manuscript as well as prepared the graphs. Dr. Pradyumna Kumar Tripathy has made the description of the graphs in the Results and Discussion part and has finally made the review of the manuscript.

## Funding No funding is available to carry out this research work

Conflict of Interest: The authors declare that there is no competing interests (neither financial nor non-financial) as defined by Springer, or any other

interests that might be perceived to influence the results and/or discussion reported in this paper.

## References

1. Chander Diwaker, P.T.: Identification of factors and techniques to design and develop component based reliability model. *International Journal of Scientific Research in Computer Science and Engineering* 5, 107–114 (2017)
2. Li, R., Li, Q., Huang, N., Kang, R.: A simplified reliability analysis method for cloud computing systems considering common-cause failures. *Proceedings of the Institution of Mechanical Engineers, Part O: Journal of Risk and Reliability* 231(3), 324–333 (2017)
3. Zhang, F.: Research on reliability analysis of computer network based on intelligent cloud computing method. *International Journal of Computers and Applications* 41(4), 283–288 (2019)
4. Mesbahi, M.R., Rahmani, A.M., Hosseinzadeh, M.: Reliability and high availability in cloud computing environments: a reference roadmap. *Human-centric Computing and Information Sciences* 8(1), 1–31 (2018)
5. Mesbahi, M.R., Rahmani, A.M., Hosseinzadeh, M.: Reliability and high availability in cloud computing environments: a reference roadmap. *Human-centric Computing and Information Sciences* 8(1), 1–31 (2018)
6. Alam, A.B., Zulkernine, M., Haque, A.: A reliability-based resource allocation approach for cloud computing. In: *2017 IEEE 7th International Symposium on Cloud and Service Computing (SC2)*, pp. 249–252 (2017). IEEE
7. Migov, D.A.: Parallel methods for network reliability calculation and cumulative updating of network reliability bounds. In: *2018 3rd Russian-Pacific Conference on Computer Technology and Applications (RPC)*, pp. 1–5 (2018). IEEE
8. Das, D., Tripathy, C.R., Tripathy, P.K., Kabat, M.R.: A genetic algorithm based approach for designing multi-state computational grid with cost and bandwidth constraints. *Journal of King Saud University-Computer and Information Sciences* (2018)
9. Dhingra, M., Gupta, N.: Algorithms to enhance the reliability of virtual nodes using adaptive fault tolerance techniques. *International Journal of Innovative Technology and Exploring Engineering* 8(11), 515–519 (2019)