Bull. Nov. Comp. Center, Num. Anal., 17 (2015), 35–42

A parallel method for reliability calculation of diameter constrained networks*

Sergei Nesterov, Denis Migov

Abstract. The problem of network reliability calculation is studied. It is assumed that a network has unreliable communication links and perfectly reliable nodes. The diameter-constrained reliability for such a network is defined as probability that every pair of terminals of network is connected by operational paths with a number of included edges less or equal to a given integer. The problem of computing this characteristic is known to be NP-hard, just like the problem of computing the network connectivity probability. For solving this problem, we propose the parallel method, which is based on the well-known factoring method. The analysis of the numerical experiments allowed us to set some important parameters of the algorithm to speed up calculations.

1. Introduction

In the present paper, we consider the networks where the links are subject to random failures under the assumption that failures are statistically independent. Random graphs are commonly used for the modeling of such networks. As a rule, the network reliability is defined as some connectivity measure. The most common reliability measure of such networks is the probability that all the terminal nodes in a network can keep connected together, given the reliability of each network node and edge. The problem of calculation of the network probabilistic connectivity is known to be NP-hard [1]. Nevertheless, it is possible to conduct the exact calculation of reliability for networks with a dimension of practical interest by taking into consideration some special features of real network structures and based on modern high-speed computers [1–8].

In practice, it is often not enough to have a path between a chosen pair of nodes, but it is necessary to have a path passing via a limited number of communication links [9]. For example, if there is a constraint on the time it takes to transmit the data between two nodes, T, then the number of transit nodes participating in this transmission must not exceed T/t, where t is the time it takes to process the data on each network node. Thus, we arrive at a different reliability measure. The diameter-constrained network reliability (DCNR) is a probability that every two nodes from a given set of terminals are connected with a path of length less or equal to a given

*Supported by the Russian Foundation for Basic Research under Grants 14-07-31069, 13-07-00589.

integer. By the length of a path we understand the number of edges in this path. This reliability measure was introduced in [10] and studied in more detail in [11–13]. The problem of computing this measure in general is known to be NP-hard, just like the problem of computing the probability of network connectivity. By now the complexity of the DCNR calculation has been completely studied for different diameter values and a number of terminals [13]. In our previous studies we developed some methods for speeding-up the DCNR calculations [14–16].

However, despite the improvements achieved on the efficiency of the computational methods for the reliability analysis, such methods still are ineffective, and so their parallel realizations are needed for operations on modern supercomputers. By now in this area we have only a parallel approach to the network reliability estimation by Monte Carlo technique [17, 18] and the parallel implementation of the well-known factoring method, which was proposed in one of our previous work [19].

In this paper, we propose a parallel method for the DCNR calculation. The proposed method is based on the well-known sequential factoring method [20]. We have chosen the fastest modification [10] of the factoring method for the DCNR calculation with the improvements proposed in [16]. For the parallel implementation we have chosen "Master-Slave" parallel programming model, as we done for the calculation of the network probabilistic connectivity [19]. The analysis of the results of the numerical experiments allowed us to optimize some important parameters of the algorithm which further increase its speedup and scalability.

2. The basic definitions and notations

We represent a network with unreliable edges and perfectly reliable nodes by an undirected graph $G = (V, E)$. For each edge e, the presence probability $0 \leq r_e \leq 1$ is given. Further on we refer to this probability as edge reliability. Also, a set of terminals K is given. It is supposed that the network operates well when any pair of terminals can establish a connection via only the operational edges.

An elementary event is a special realization of the graph defined by the existence or absence of each edge. The probability of an elementary event equals the product of probabilities of the existence of operational edges multiplied by the product of probabilities of the absence of faulty edges.

A reliability of G with a diameter constraint d is defined as the sum of elementary events in which every pair of terminals can be connected by a path of length not exceeding d. By a length of path we mean the number of edges which belong to this path. We denote this reliability measure by $R_K^d(G)$.

3. Methods for the DCNR calculation

The definition of the DCNR gives a method for computation of this parameter. However, such a direct approach results in an exhaustive search of all graph realizations; thus, this method is ineffective even for small-scale enumeration networks. That is why other methods are used for the calculation of different reliability measures. The most common method among them is the factoring method [20], which can be applied to any network reliability measure, including the DCNR. The factoring method divides the probability space into two sets, based on the success or the failure of one graph particular unreliable element: a node or an edge. We consider networks with unreliable edges only, so we may choose any unreliable edge. The chosen edge is called pivot. So, we obtain two subgraphs. In one of them the pivot edge is absolutely reliable, and in the second one the pivot edge is absolutely unreliable, that is, absent. The probability of the first event is equal to the reliability of the pivot edge; the probability of the second event is equal to the failure probability of the pivot edge. Thereafter obtained subgraphs are subject to the same procedure. The total probability law gives the expression for the network reliability. For the DCNR, we have the following formula:

$$R_K^d(G) = r_e R_K^d(G/e) + (1 - r_e) R_K^d(G \backslash e), \tag{1}$$

where $G\backslash e$ is the graph G without edge e, G/e is the graph G with the absolutely reliable edge e. Recursions continue until a graph is obtained, in which at least one pair of terminals can not be connected by the path of a limited length (returns 0), or all pairs of terminals are connected by absolutely reliable paths (returns 1). Further on we refer to this method as SFM (simple factoring method).

A modified factoring method for the DCNR calculation was proposed by Cancela and Petingi [10]. Further on we refer to this method as the CPFM (the Cancela & Petingi factoring method). This method is much faster than basic factoring method in the diameter constrained case (1). The main feature of the modified factoring method is operating with a list of paths instead of operating with graphs. In the preliminary step for any pair of terminals s, t, the list $P_{st}(d)$ of all the paths with a limited length between s, t is generated. It automatically removes all edges, which do not belong to any such path, from the consideration. For example, all the so-called "attached trees" without terminals are no longer considered. By P_d the union of $P_{st}(d)$ for all the pairs of terminals is denoted. By $P(e)$, a set of paths from P_d, which include the link e, is denoted. Parameters of the modified factoring procedure are not graphs. Instead, we use 6 parameters, which describe the corresponding graph in terms of P_d. Here we do not outline the algorithm pseudocode, it can be found in [10].

One of the main reasons, which makes the calculation of diameter constrained network reliability much more complicated as compared to other network reliability measures, is the lack of methods for decreasing the recursion quantity. In our previous studies [14–16], we obtained the methods which can make the DCNR calculation faster. These methods are the following: the analogue to the well-known series-parallel transformation for the CPFM, and the pivot edge selection strategy, which can be well applied to the DCNR calculation by the CPFM and by the SFM (1). Also, we have obtained the decomposition methods for calculating the DCNR for the networks with two terminals. The methods obtained allow us to significantly reduce the number of recursive calls in the CPFM and complexity of the DCNR computation.

4. Parallel implementation of the method for DCNR calculation

In this section, we introduce an algorithm with the use of the MPI for the DCNR calculation for supercomputers with distributed memory. The main idea is based on the CPFM method: in the factoring procedure one part of the problem (for example, "contracting" of an edge) remains with the paternal process while the other one is sent to some idle process.

As in our previous research into parallel computing of the network reliability [19] (without diameter constraint), we chose "Master–Slave" parallel programming model (Figure 1). The main idea of such an approach is that one *master* process controls all the other *guided* processes.

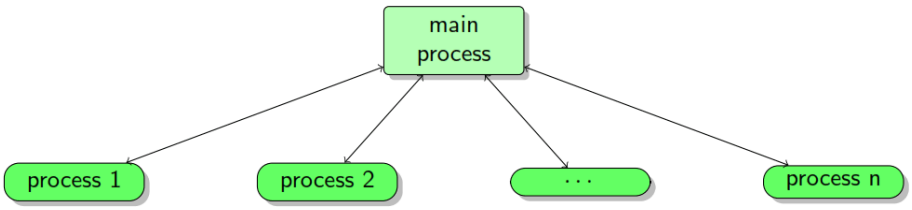

Figure 1. "Master–Slave" model

It was decided to send to a child process all required data for calculation with a probability of the corresponding event. This allows us to store only the part of result reliability with every process; it should be just summarized at the end of the factoring algorithm. Probability of every partial "subnetwork" is a multiplicative factor which is initialized by 1 for the initial network. It is being changed during the factoring procedure: the probability of $G\backslash e$ is the probability of G multiplied by r_e, and for G/e — by $(1 - r_e)$. Below we represent pseudo code of algorithms for master process and for guided processes. The algorithm for guided processes calls the recursive factoring procedure; pseudo code of this procedure is given separately.

```
function MasterProcess
    evaluate(input)                          % make all required data structures
    send(input) to process 1                                  % run 1st process
    while there is any busy process
        receive(message) from sender
        if message = I_AM_FREE
            sender := idle status
        else if message = I_NEED_HELP
            helper := 0 or idle process
            send(helper) to sender
        end if
    end while
end function

function SlaveProcess
    R_K^D := 0                               % reliability part on current process
    recursionCount := 0                                    % on current process
    do
        receive(message) from sender
        if message = MPI_FACTO
            receive(input) from sender
            MPIFactorization(input, 1, R_K^D)
        else if message = CONTRACT_EDGE
            receive(input, e) from sender
            contractEdge(input, e, P_l, R_K^D)
        end if
    while message ≠ STOP
    send(recursionCount, R_K^D) to master process
end function

function MPIFactorization(P_l, R_K^D)
    e := some edge
    send(help message) to master process
    receive(helper) from master process
    if helper = 0
        contractEdge(input, e, P_l * p_e) from sender
                                                % start edge contracting here
    else
        send(input, e, P_l * p_e) to helper          % send this part to a helper
    end if
    deleteEdge(input, e, P_l * (1 − p_e)) to master process
                                                        % another part of job
end function
```

In [19], one important parameter of the algorithm was studied: the lower limit of the dimension of a graph that could be assigned to another process. Here we study the analogue to this parameter with a diameter constraint. In this case it turns to the upper limit of the number of considered edges. After the number of considered edges exceeds this limit, the current process stops sending data to the master process and executes all procedures without any help. We define this parameter as N_{edges}. The best values of this parameter strongly depend on the graph structure and computer architecture and can be selected experimentally. Below we are trying to find the optimal value of N_{edges}, which makes the algorithm faster.

5. Case studies

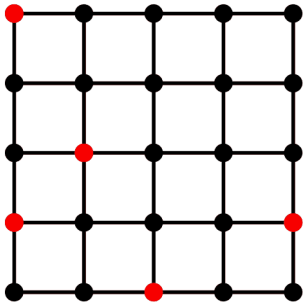

Figure 2. A test graph for the proposed algorithm

In this section, we examine the scalability of the algorithm proposed. We also try to find the optimal value for N_{edges}. For the numerical experiments we choose a 5×5 topology grid, it contains 25 vertices and 40 edges. The number of terminals is equal to 5, diameter being equal to 9 (Figure 2). In spite of its small dimension, this graph is very complicated for computing its DCNR because of non-applicability of various accelerating methods. The experiments were carried out on the computing cluster HKC-30T of the Siberian Supercomputer Center. This cluster consists of double-blade servers HP BL2220 G6 with Intel Xeon 5540 2.53 GHz CPUs.

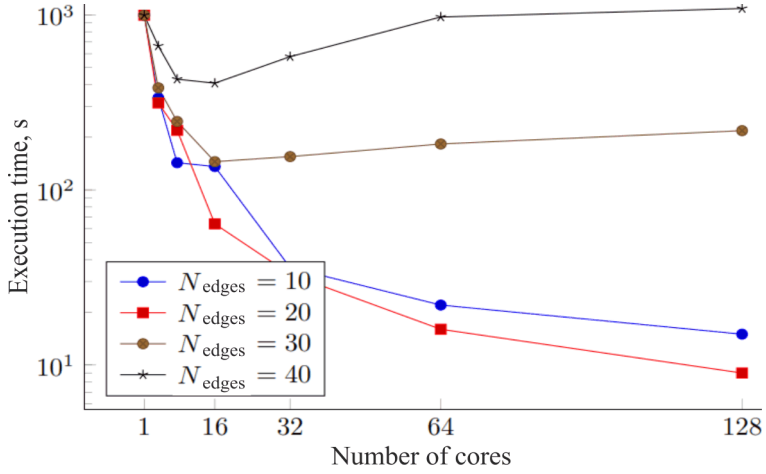

Figure 3. Scalability of the parallel algorithm

Different values of N_{edges} were considered. Figure 3 shows the scalability of the proposed algorithm for different values of N_{edges}. As we can see, the optimal value is between 10 and 20.

The results show that the CPFM works well in parallel implementation for supercomputers with distributed memory. The algorithm shows a linear speedup for the number of cores not exceeding 128.

6. Conclusion

In this paper, we introduced a parallel implementation of the factoring method for the exact calculation of the network reliability in with the diameter constraint. Also, we offer to set one important parameter of the algorithm proposed, which significantly improve its performance. The results of the numerical experiments show that the CPFM works well in parallel implementation for supercomputers with distributed memory. Our primary goal is to further improve the scalability of the proposed algorithm. It seems that there are two ways to do this: using several master processes or not using them at all.

References

[1] Colbourn Ch. J. The Combinatorics of Network Reliability. — New York: Oxford University Press, 1987.

[2] Wood R.K. Triconnected decomposition for computing k-terminal network reliability // Networks. — 1989. — Vol. 19. — P. 203–220.

[3] Shooman A.M., Kershenbaum A. Methods for communication-network reliability analysis: probabilistic graph reduction // Proc. Reliability and Maintainability Symposium. — 1992. — P. 441–448.

[4] Migov D.A., Rodionova O.K., Rodionov A.S., Choo H. Network probabilistic connectivity: using node cuts // Lect. Notes Comp. Sci. — Springer, 2006. — Iss. 4097. — P. 702–709.

[5] Won J.-M., Karray F. Cumulative update of all-terminal reliability for faster feasibility decision // IEEE Trans. on Reliability. — Sep. 2010. — Vol. 59, No. 3. — P. 551–562.

[6] Rodionov A.S., Migov D.A., Rodionova O.K. Improvements in the efficiency of cumulative updating of all-terminal network reliability // IEEE Trans. on Reliability. — June 2012. — Vol. 61, No. 2. — P. 460–465.

[7] Xiao Y. An efficient network reliability computation method based on isomorphism judgment // J. Networks. — 2014. — Vol. 9, No. 8. — P. 2037–2044.

[8] Forghani-elahabad M., Mahdavi-Amiri N. A new efficient approach to search for all multi-state minimal cuts // IEEE Trans. on Reliability. — 2014. — Vol. 63, No. 1. — P. 154–166.

[9] Pandurangan G., Raghavan P., Upfal E. Building low-diameter peer-to-peer networks // IEEE J. on Selected Areas in Communications. — 2003 — Vol. 21, No. 6. — P. 995–1002.

[10] Cancela H., Petingi L. Diameter constrained network reliability: exact evaluation by factorization and bounds // Proc. Int. Conf. on Industrial Logistics. Okinawa, Japan. — 2001. — P. 359–356.

[11] Cancela H., Robledo F., Rubino G., Sartor P. Monte Carlo estimation of diameter constrained network reliability conditioned by pathsets and cutsets // Computer Communications. — 2013. — Vol. 36, No. 6. — P. 611–620.

[12] Jin R., Liu L., Ding B., Wang H. Distance-constraint reachability computation in uncertain graphs // VLDB Endowment. — 2011. — Vol. 4. — P. 551–562.

[13] Canale E., Cancela H., Robledo F., et al. On computing the 2-diameter-constrained k-reliability of networks // Int. Transactions in Operational Research. — 2013. — Vol. 20, No. 1. — P. 49–58.

[14] Migov D.A. Computing diameter constrained reliability of a network with junction points // Automation and Remote Control. — 2011. — Vol. 72, No. 7. — P. 1415–1419.

[15] Migov D.A., Rodionova O.K. Calculating two-terminal reliability in a diameter constrained network with cutenodes // Proc. Int. Conference on Ubiquitous Information Management and Communication (ACM ICUIMC 2012, Kuala Lumpur, Malaysia). — New York: ACM, 2012. — Article No. 130.

[16] Migov D.A., Nesterov S.N., Rodionov A.S. Acceleration methods for diameter constrained network reliability calculation // Vestnik SIBSUTIS. — 2014. — Vol. 1. — P. 49–56 (In Russian).

[17] Khadiri M.E., Marie R., Rubino G.: Parallel estimation of 2-terminal network reliability by a crude Monte Carlo technique // Proc. 6th Int. Symposium on Computer and Information Sci. — Elsevier, 1991. — P. 559–570.

[18] Martinez S.P., Calvino B.O., Rocco S.C.M. All-terminal reliability evaluation through a Monte Carlo simulation based on an MPI implementation. Proc. European Safety and Reliability Conference: Advances in Safety, Reliability and Risk Management (PSAM 2011/ESREL 2012). — Helsinki, 2012. — P. 1–6.

[19] Migov D.A., Rodionov A.S. Parallel implementation of the factoring method for network reliability calculation // Computational Science and Its Applications – ICCSA 2014. Proc. 14th International Conference (Guimarães, Portugal, June 30 – July 3, 2014): Part VI. — Springer, 2014. — P. 654–664. — (LNCS; 8584).

[20] Page L.B., Perry J.E. A practical implementation of the factoring theorem for network reliability // IEEE Trans. on Reliability. — Aug. 1998. — Vol. 37, No. 3. — P. 259–267.