УДК 004.75

O. V. Hordiichuk, postgraduate

An approximate heuristic reliability network allocation algorithm

Taras Shevchenko National University of Kyiv, 83000, Kyiv, Glushkova st., 4d,

e-mail: oleg.gordiychuck@gmail.com

Гордійчук О. В., аспірант

Приблизний евристичний алгоритм оптимізації надійності мережі

Київський національний університет імені Тараса Шевченка, 83000, м. Київ, пр-т. Глушкова 4д,

e-mail: oleg.gordiychuck@gmail.com

Оптимізація надійності мережі це важлива задача, яка необхідна для побудування стійких однорангових систем та бездротових мереж передачі даних. Головна ціль в цій задачі – побудувати надійну мережу, використовуючи якомога меншу кількість з'єднань поміж вузлами. Ця проблема також відома, як виділення надійності для К-терміналів, яка є NP-складною. В цій статті запропонований новий приблизний алгоритм, який виділяє надійність для мережі, використовуючи метод Монте-Карло для перевірки значення надійності та оптимізаційну евристику, яка базується на властивостях графів розширювачів. Результати, отримані при обчисленні великих мереж до 50000 вузлів та їх різними властивостями, свідчать про те, що цей алгоритм може бути використаний для швидкого приблизного вирішення задачі навіть на комп'ютерах із посередніми обчислювальними можливостями. А для виділення надійності у мережах середніх розмірів (від 100 до 1000 вузлів) цей алгоритм також може застосовуватися у однорангових системах, де присутнє обмеження часу на прийняття рішення.

Ключові слова: оптимізація надійності, однорангові мережі, бездротові мережі.

A reliability allocation is an important task in building robust systems such as peer-to-peer applications and wireless ad-hoc networks. A main goal is to build a reliable network using the smallest number connections between nodes. This problem is also known as a K-terminal redundacy allocation problem, which is NP-hard. This paper presents a new approximate algorithm that can allocate redundancy for network with reliable nodes using Monte-Carlo simulation for reliability checking and an optimization heuristic based on an expander graph properties. The results obtained from calculation large networks with different properties and with up to 50000 nodes show that this algorithm can be used for a fast solving even on a moderate computer capabilities. And for reliability allocation in networks with medium size (from 100 to 1000 nodes) this algorithm also can be used in peer-to-peer networks, where time limitation for execution exists.

Key Words: reliability allocation, peer-to-peer networks, wi-fi ad-hoc networks.

Статтю представив д.т.н., проф. Погорілий С.Д.

Introduction

Today systems without centralized component (decentralized systems) become more popular as they are self-constructive, have huge computation capabilities and good anonymous properties. An example of such systems could be peer-to-peer applications like BitTorrent [1] and wireless ad-hoc networks. All of these systems form a network either software or hardware. However all nodes of such systems are unreliable that's why their topology should be robust as even big networks often fail.

The network could be represented as an undirected graph $G = (V, E)$, where V is a set of nodes (vertices) and E - set of connections (edges) between nodes. Each node has own reliability property that is defined by a function $0 < r(v) \leq 1, v \in V$. Network may have reliable or unreliable nodes and connections, but in this paper it considered only with unreliable nodes and reliable connections. The network is functioning ($\Phi(G) = 1$) if any and only if for any pair $(v, w), v \in V, w \in V$ there exists at least one path $\{v, x_1, x_2, \dots x_n, w\}, x_1, \dots x_n \in V$ that connects two nodes or in case when network is empty ($V = \emptyset$). In all other cases we say that network is defective - $\Phi(G) = 0$.

While functioning network may change its topology as some nodes may fail. As network has $|V|$ nodes there exists a set Ω of all $2^{|V|}$ functioning network states $G' = (V', E'), V' \in V, E' \in E$. This leads to the fact that redundancy could be calculated as a reliability sum of all functioning states of the network:

$$R(G) = \sum_{G' \in \Omega} \Phi(G') \prod_{v\prime \in V\prime} r(v') \prod_{v \in V \setminus V\prime} (1 - r(v)) \quad (1)$$

It's known that computation of network reliability is an NP-hard problem [9]. Another common and not less important task is a network redundancy allocation.

This problem has different formulation, but in this paper considered optimization of network reliability with using the least number of connections between nodes:

$$\begin{gathered} \min|E| \\ R(G) > \varepsilon \end{gathered} \quad (2)$$

The redundancy allocation problem is also not an easy task as computation time of it using classical optimization approaches like Lagrangian multipliers [2] exponentially grows with a network size. That's why in big networks or real time applications approximation methods are used. One of this method is a tabu-search [3], where a meta-heustric of finding optimal solution is used. The main idea of this method is to choose each new solution candidate x' in neighborhood to a current step at x. Other approaches use quazi-optimal methods like ant-colony algorithm [4] and genetic algorithm [5]. Also some researches proposed optimization redundancy solutions using neural networks [6]. Anyway all of these methods are not well suitable for really big-sized networks neither in computing time nor in accuracy.

In this paper a new approximate redundancy network allocation algorithm is proposed which is suitable for a fast reliability allocations. This solution is based on a expander graphs properties, estimation of the network reliability is done by a Monte-Carlo method and an optimization - by binary search approach. The resulting algorithm is easy to implement and it may be used for redundancy allocation even for big-sized network (> 1000 nodes).

Alternative network reliability concepts

Besides exact and approximate methods of reliability evaluation it also can be expressed in meaning of network connectivity properties. One of them is a minimal cut - set of nodes that divides network into two disconnected subnetworks. It's known that cardinality of a minimal cut is equal to a minimum count of paths between any two nodes in the network (max-flow min-cut theorem [7]). This fact could be used to obtain the minimal cut and there are efficient heuristic algorithms [8] for finding it. If all minimal network cuts is known as well as their reliability values $p_1, p_2, \dots, p_n$ it's also possible in this case use an inclusion-exclusion formula (Poincare theorem)[9] for exact network reliability calculation:

$$\begin{aligned} R(G) = \sum_{1 \le i \le} p_i - \sum_{1 \le i_1 \le i_2 \le n} p_{i_1} p_{i_2} + \cdots \\ + (-1)^{m+1} p_1 p_2 \dots p_n \end{aligned} \quad (3)$$

However such approach is used rarely as finding the all minimal cut-set as well as calculation sums of probabilities are a costly operations. Anyway this solution is useful for estimating when most of the right part of the sums is not used.

Another important property in network reliability is a Cheeger constant [11]. Let's define a function $e(A) = \{(x, y) \in E \mid x \in A, y \in V \setminus A\}$ which describes a set of edges from some subset A to a remaining subset $V \setminus A$. Then the Cheeger constant $h(G)$ is defined as following:

$$h(G) = \min\left\{\frac{|e(A)|}{|A|}\right\}, A \subseteq V, 0 < |A| \le \frac{|V|}{2} \quad (4)$$

This function describes how any subset $A \subseteq V$ is connected to the remaining subset $V \setminus A$. The bigger value this constant has the bigger the minimal cutset of the network is. A graph with good connectivity properties is called an expander. The network is a (d, ϵ)-expander if it is d-regular and $h(G) > \epsilon$. There are several approaches for building expander graphs: algebraic [12], zig-zag product [13] and random-walks [14]. An algorithm that is proposed in this paper use random-walks approach as building of big random graphs is faster than building structured and it's known [14] that a d-regular random graph is a $(d, \frac{d}{4})$ expander with a high probability. This means that d-regular random graphs have good reliability and this fact is used for allocation in algorithm proposed in paper.

Monte Carlo method for the network reliability

While exact methods for calculation network reliability have exponential complexity, it's possible to use approximate approaches to achieve result with some accuracy. As mentioned before the Poincare theorem may be used for it, but it requires finding all min-cut sets. On the other hand a Monte Carlo method may be used because it only needs to run some amount of tests for achieving provided accuracy. For this an approximate network reliability algorithm should simulate nodes failings according to the provided reliability and after that test resulting network for functioning (function 1). The complexity of Algorithm 1 is equal to $O\,(2|V| + |E|)$ if network network is represented as list of nodes where each of it contains list of links to other nodes. If to run this operation n times (function 2) and calculate ratio between successful tests with overall number of tests then an approximate value of network reliability will be achieved.

SimulateAndTest
$\boldsymbol{input}$: $network\ topology\ G = (V, E)$
$\boldsymbol{output}$: $true\ if\ generated\ network\ is\ functioning\ and\ false - otherwise$
$\boldsymbol{begin}$
$V' \leftarrow \emptyset$;
$E' \leftarrow \emptyset$;
$\boldsymbol{for}\ v \in V\ \boldsymbol{do}$
  $p \leftarrow rand(\,)$;
  $\boldsymbol{if}\ p > r(v)\ \boldsymbol{then}$
    $V' \leftarrow V' \cup v$;
    $E' \leftarrow E' \cup c(v, V')$;
  $\boldsymbol{end}$
$\boldsymbol{end}$
$K' \leftarrow BFS(V', E')$;
$\boldsymbol{return}\ |K'| = |V'|$;
$\boldsymbol{end}$

Function 1. Simulates possible random network structure and evaluates its reliability using BFS (a breadth-first search function).

$\boldsymbol{CalculateReliability}$
$\boldsymbol{input}$: $network\ topology$ G = (V, E), number of tests n
output: network reliability (probability that it functions)
begin
t ← 0;
$\boldsymbol{for}\ i\ \leftarrow 0\ to\ n\ \boldsymbol{do}$
  $\boldsymbol{if}\ SimulateAndTest(G)\ \boldsymbol{then}$
    $t \leftarrow t + 1$;
  $\boldsymbol{end}$
$\boldsymbol{end}$
$\boldsymbol{return}\ \frac{t}{n}$;
$\boldsymbol{end}$

Function 2. An approximate network reliability evaluation using Monte Carlo method.

Binary search of optimal network reliability

Let's investigate properties of reliability function. $R(G) = 0$ is a lower bound of function and it's possible then and only then when all values of $r(v) = 0$, no matter how many connections network has. If the network is a complete graph its reliability is 100% ($R(G) = 1$), because its min-cut set is equal to an empty set. It is a monotonically increasing function to number of edges in the network $R(G = (V, E)) < R(G' = (V, E \cup e))$. Such function behaviour makes possible to increase or decrease reliability by adding or removing connections. In any network there are up to $\frac{|V|(|V|-1)}{2}$ edges and some combination of it provides optimal reliability. The task of finding exact allocation is NP-hard and that is why it's reasonable to use heuristic based on described in previous section properties of expander graph. The main idea of proposed algorithm is to build d-degree random network that will have expander properties by changing a cardinality of connections set E at each step. Adding one edge per turn will significantly affect the algorithm speed in a bad manner. For avoiding this problem a possible solution is to use binary search algorithm. In this case a count of steps that should be done for finding quazi-optimal solution lies in bounds $\left[0, \frac{|V||V|-1}{2}\right]$. If at current step a value of reliability is still far a way from searching value accuracy then the next step should be increased by two times. It continues until searching value is not found. In case when calculated reliability is more than provided searching value R^* and their absolute difference is more than accuracy value, next steps are done using binary search algorithm (function 3).

$\boldsymbol{AllocateReliability}$
$\boldsymbol{input}$: $set\ of\ nodes\ V, their\ reliability\ values\ of\ r(v), v\ \in V, desirable\ network\ reliability\ R^*, accuracy\ \varepsilon, number\ of\ tests\ n$
$\boldsymbol{output}$: $network\ topology\ G = (V, E)$
$\boldsymbol{begin}$

```
E ← ∅;
R ← 0;
G ← (V, E);
step ← 1;
fastsearch ← true;
up ← true;
e ← nil;
do
begin
   if up then
      for i to step do
      begin
        e ← GenerateRandomEdge(E);
        E ← E ∪ e;
      end
   else
      for i to step do
      begin
        e ← GetRandomEdge(E);
        E ← E\e;
      end
    end
R ← CalculateReliability(G, n);
if (|R − R*| > ε) then
  if up then
     if R > R* then
        if fastsearch then
           step ← step * 2;
        else
           step ← step/2;
        end
     else
      fastsearch ← false;
      up ← false;
      step ← step/2;
     end
  else
     step ← step/2;
     if R > R* then
        up ← true;
     end
end
end
while |R − R*| > ε;
end
```

Function 3. The network reliability allocation algorithm.

If there is a need to get some accuracy ε with high probability it's possible to use Laplassian function:

$$P(|R^{*}-R|<\varepsilon)=2L\left(\frac{\varepsilon\sqrt{n}}{\sqrt{R(1-R)}}\right) \qquad (5)$$

, where R^{*} is observing reliability and R - real reliability and L is a Laplassian function.

Evaluation of method

Methods described in previous sections are implemented in C++. The program has 242 lines of code. It has run with different input data for achieving detailed statistic. All tests have executed 100 times and their average execution time collected. The results presented in a next table:

$\|V\|$	R	ε	n	A	Average execution time
100	0.1	0.05	100	0.9	78
100	0.99	0.05	100	1	96
100	0.9	0.05	100	0.9	163
1000	0.9	0.01	10000	0.99	137267
1000	0.99	0.01	10000	1	130557
1000	0.9	0.5	10000	1	53842
1000	0.5	0.05	1000	0.98	9752
1000	0.9	0.05	1000	1	7280
1000	0.99	0.05	1000	1	7034
1000	0.9	0.05	100	0.9	909
1000	0.5	0.05	100	0.78	1525
10000	0.9	0.05	1000	1	377109
10000	0.99	0.05	1000	1	91814
10000	0.9	0.05	100	0.9	24297
10000	0.99	0.05	100	1	12701
20000	0.99	0.05	100	1	23212
50000	0.99	0.05	100	1	96123
10000	0.85	0.05	100	0.8	17532
20000	0.9	0.05	100	0.9	46542
50000	0.9	0.05	100	0.9	243747

Table 1. Evaluation of the method. Average exection time presented in milliseconds.

It's possible to make next conclusions based on obtained results:

1. Execution time inversely depends on reliability that is allocated;
2. Fast and accurate results could be achieved by using small (100) number of tests n if allocated reliability tends to 1.

The first fact comes up from the behavior of function 2. On the one hand it means that for real tasks such as network reliability allocation for networks smaller than 1000 nodes, where big values of reliability is preferred, it's possible to use this method for redundancy allocation in real time systems such as peer-to-peer applications and wi-fi ad-hoc networks. And on the other hand it's possible to use this method for reliability allocation even for big networks (10000 of nodes).

Conclusion

In this paper proposed the new hybrid approximate algorithm for reliability allocation. While there are a lot of existing approximate and exact solutions, it has shown that there is still no algorithm that is suitable for real time systems such as peer-to-peer applications and wireless ad-hoc networks. The network redundancy allocation is done in 2 steps: calculating graph reliability and next step guessing. The proposed approach uses idea that comes from the expander graph properties and that's why by its nature the algorithm produces random graphs, where all vertices have nearly a same value of edges. However it should be additional investigated possibility of building reliable networks with fixed number of vertex edges.

The algorithm described in this paper evaluates network functioning at each step that is done by using so known breadth-first search algorithm. Also the Monte Carlo method, which is often used for solving tasks by collecting statistically information, is used for calculating network reliability that gives ability to do it fast. And finally the upgraded version of the binary search algorithm is used for finding semi-optimal solution.

The proposed solution has been implemented and evaluated in C++ language with different input data. It has been noticed that an execution speed of this algorithm depends not only on the actual network size and number of Monte Carlo tests, but also on the achieving reliability. As well as it's possible to achieve accurate and fast results with small number of tests (100) in case when allocated reliability tends to the 1. So this method could be used for obtaining very accurate result for even networks with 10000 nodes running only 6 minutes in average on moderate computer capabilities and at the same time it could be used for a real-time calculations (less than 1s) in medium-size networks with number of nodes equal to 1000.

Список використаних джерел

1. *BitTorrent.” http://www.bittorrent.com”.*
2. *Kuo W.* Reliability optimization with the Lagrange-multiplier and branch-and-bound technique / *Kuo W.* Reliability, IEEE Transactions on. – 1987. – Vol. 36. – №. 5. – P. 624-630.
3. *Kulturel-Konak S.* Efficiently solving the redundancy allocation problem using tabu search / *Kulturel-Konak S., Smith A. E., Coit D. W.* IIE transactions. – 2003. – Vol. 35. – №. 6. – P. 515-526.
4. *Liang Y. C.* An ant colony optimization algorithm for the redundancy allocation problem (RAP) / *Liang Y. C., Smith A. E.* Reliability, IEEE Transactions on. – 2004. – Vol. 53. – №. 3. – P. 417-423.
5. *Kanagaraj G.* A hybrid cuckoo search and genetic algorithm for reliability–redundancy allocation problems / *Kanagaraj G., Ponnambalam S. G., Jawahar N.* Computers & Industrial Engineering. – 2013. – Vol. 66. – №. 4. – P. 1115-1124.
6. *Coit D. W.* Solving the redundancy allocation problem using a combined neural network/genetic algorithm approach / *Coit D. W., Smith A. E.* Computers & operations research. – 1996. – Vol. 23. – №. 6. – P. 515-526.
7. *Christos H.* 6.1 The Max-Flow, Min-Cut Theorem. Combinatorial Optimization: Algorithms and Complexity. / *Christos H., Papadimitriou, Kenneth Steiglitz.* Dover., 1998. – P. 120-128.
8. *Ugurlu O. et al.* New Algorithm for Finding Minimum Vertex Cut Set / *Ugurlu O. et al.*, Problems of Cybernetics and Informatics (PCI), 2012 IV International Conference. – IEEE, 2012. – P. 1-4.
9. *Lucet C.* Exact methods to compute network reliability / *Lucet C., Manouvrier J. F.* Statistical and Probabilistic Models in Reliability. – Birkhäuser Boston, 1999. – P. 279-294.
10. *Srivaree-ratana C.* Estimation of all-terminal network reliability using an artificial neural network / *Srivaree-ratana C., Konak A., Smith A. E.* Computers & Operations Research. – 2002. – Vol. 29. – №. 7. – P. 849-868.

11. *Donetti L.* Optimal network topologies: Expanders, Cages, Ramanujan graphs, Entangled networks and all that / *Donetti L., Neri F., Munoz M. A.* Journal of Statistical Mechanics: Theory and Experiment. – 2006. – Vol. 2006. – №. 08. – P. 807.
12. *Michael Holroyd*, Synchronization and Connectivity of Discrete Complex Systems / *Michael Holroyd* International Conference on Complex Systems, 2006.
13. *Reingold O.* Entropy waves, the zig-zag graph product, and new constant-degree expanders and extractors / *Reingold O., Vadhan S., Wigderson A.* Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on. – IEEE, 2000. – P. 3-13.
14. *Goldreich O.* Basic facts about expander graphs / *Goldreich O.* Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation. – Springer Berlin Heidelberg, 2011. – P. 451-464.
15. *Kuo W.* Recent advances in optimal reliability allocation / *Kuo W., Wan R.* Computational Intelligence in Reliability Engineering. – Springer Berlin Heidelberg, 2007. – P. 1-36.

References

1. BIT TORRENT. A peer-to-peer file distribution client application. ”http://www.bittorrent.com”.
2. KUO W. (1987) Reliability optimization with the Lagrange-multiplier and branch-and-bound technique. *In Reliability, IEEE Transactions on 36 (5)*. – p. 624-630.
3. KULTUREL-KONAK S., SMITH A.E., COIT D.W. (2003) Efficiently solving the redundancy allocation problem using tabu search. *In IIE transactions 35 (6).* p. 515-526.
4. LIANG Y. C., SMITH A. E. (2004) An ant colony optimization algorithm for the redundancy allocation problem (RAP). *In Reliability, IEEE Transactions on 53 (3).* p. 417-423.
5. KANAGARAJ G., PONNAMBALAM S.G., JAWAHAR N. (2013) A hybrid cuckoo search and genetic algorithm for reliability–redundancy allocation problems. *In Computers & Industrial Engineering 66 (4).* p. 1115-1124.
6. COIT D. W., SMITH A. E. (1996) Solving the redundancy allocation problem using a combined neural network/genetic algorithm approach. *In Computers & operations research 23 (6)*. p. 515-526.
7. CHRISTOS H., PAPADIMITRIOU, KENNETH S.. (1998) 6.1 The Max-Flow, Min-Cut Theorem. Combinatorial Optimization: Algorithms and Complexity. *Dover*. p. 120-128.
8. UGURLU O.. (2012) New Algorithm for Finding Minimum Vertex Cut Set. *In Problems of Cybernetics and Informatics (PCI), IV International Conference. – IEEE.* p. 1-4.
9. LUCET C. MANOUVRIER (1999) J. F. Exact methods to compute network reliability. Statistical and Probabilistic Models in Reliability. *Birkhäuser Boston*. p. 279-294.
10. SRIVAREE-RATANA C., KONAK A., SMITH A. E. (2002) Estimation of all-terminal network reliability using an artificial neural network. *Computers & Operations Research 29 (7).* p. 849-868.
11. DONETTI L., NERI F., MUNOZ M. A. (2006) Optimal network topologies: Expanders, Cages, Ramanujan graphs, Entangled networks and all that. *Journal of Statistical Mechanics: Theory and Experiment 2006 (08).* p. 807.
12. MICHAEL H. (2006). Synchronization and Connectivity of Discrete Complex Systems. *International Conference on Complex Systems 2006*.
13. REINGOLD O., VADHAN S., WIDGERSON A. (2000) Entropy waves, the zig-zag graph product, and new constant-degree expanders and extractors. *Foundations of Computer Science, 2000. Proceedings. 41st Annual Symposium on*. p. 3-13.
14. GOLDREICH O. (2011) Basic facts about expander graphs. *Studies in Complexity and Cryptography. Miscellanea on the Interplay between Randomness and Computation. – Springer Berlin Heidelberg*. – p. 451-464.
15. KUO W., WAN R. Recent advances in optimal reliability allocation. *Computational Intelligence in Reliability Engineering. – Springer Berlin Heidelberg*. p. 1-36.

Надійшла до редколегії 20.06.14