ERJ
Engineering Research Journal
Faculty of Engineering
Minoufiya University

MEASURING THE NETWORK RELIABILITY OF THE DEVICE PLACEMENT LOCATION BETWEEN TWO TERMINALS BY USING CUT-SET METHOD

Farhat M. Alfakhari
Department of Computer Science
Al-Tahaddi University – LIBYA
Email: farhatalfakary@gmail.com

AbduAlsalam Ambarek
Department of Electrical and Electronic Engineering
Garyounis University – LIBYA
Email: abdualsalam_ambarek@yahoo.com

ABSTRACT

The Device Placement Location *(DPL)* is concerned with locating the placement of the corresponding electronic device within certain existing user locations. It considers the cost parameter of the device placement location and routing users. This paper measures the reliability of the DPL and assigning the users to the *DPL* by implementing the *DPLRC & DPLRC* algorithms. In fact, our models are similar in spirit to [26]. However, there is a striking difference between them. In this paper, the users are assigned to the device locations only after measuring the network reliability. The *DPLRC* algorithm is applied to minimize the cost of placing the devices and maximize the reliability of users that are assigned to the device locations. The *DPLRD* algorithm is used to maximize the reliability of placing the devices and the demands of the users that are assigned to device locations. We have used the *Hao-Orlin* algorithm to measure the network reliability. It is implemented to run this algorithm in *O (nm log (n2/m))* time. All the three algorithms mentioned above have been analyzed and designed by the use of *UML*. They are implemented by the use of *C#.NET 2005* programming language. We discuss the relationship between the nodes, edges, and the effectiveness of the cost on the increase in the *DPLs* with the aid of tradeoff curves and tables. We also look at the performance of the algorithms by measuring the *CPU* time taken to find the *DPLs*. The results of this paper demonstrates that the number of the *DPLs* and the users assigned vary according to the total number of nodes, edges and the minimum cost thresholds.

إن طريقة تنسيبِ موقع الجهاز الإلكتروني DPL غايتها تَحديد مكان تنسيبِ الجهاز الإلكتروني ضمن بَعْض مواقع المستخدمين المستعملة حاليا. وهي تأخذ في اعتبارها كلفة موقع تنسيبِ الأداةَ وكلفة مسار المستخدمين الذين سيخصصوا لذلك الجهاز. تقيسُ هذه الورقةِ مدى اعتمادية موقع تنسيبِ الجهاز الإلكتروني وتخصيص مستخدمين لموقع تنسيبِ هذا الجهاز باستخدام خوارزميتين *DPLRC* و*DPLRD* . وفي الحقيقة فإن نماذجنا مماثلة للنموذج المستخدم في [26]، ولكن هناك اختلافا مُميّزا بينها. في هذا البحث، لا يُخصّصُ المستخدمون إلى مواقع الأجهزة إلا بعد قياس مدى اعتمادية الشبكة. إنّ خوارزمية *DPLRC* تُقللُ كلفة وَضْع الأجهزة DPLs وتُزيّدُ ثقة المستخدمين المُخصّصين لمواقع هذه الأجهزةَ. وتُستَعملُ خوارزمية *DPLRD* لتَزْيد اعتمادية عملية وَضْع الأجهزة DPLs وطلباتِ المستخدمين الذين يخصّصون لمواقع هذه الأجهزة.

وقد تم استخدام خوارزمية Hao-Orlin لقيّاس مدى اعتمادية الشبكة. وكلّ الخوارزميات الثلاث التي ذكرتْ سالفاً حُللتْ وصُمّمتْ بإستعمال UML، وبرمجت باستعمال لغة البرمجة C#.NET 2005 . نُناقشُ هنا العلاقة بين العُقَدِ، الحافات، وفعالية الكلفةِ بزيادة DPLs بواسطة عرض بعض المنحنيات . كذلك نناقش أداء الخوارزمياتِ بقياس وقتِ وحدة المعالجة المركزية CPU time لإيجاد DPL. نتيجة البحث المشروح بهذه الورقةِ تُبيّنُ أنَّ عددَ DPLs المستخدمين المخصّصين له بتفاوت العددِ الكليّ للعُقدِ، والحافات وعتبات الكلفةِ الدنيا.

Keywords: The network reliability, The location problem, The facility location and the device placement location.

1. INTRODUCTION

Networks involving electronic devices such as *hubs, switches, routers, etc* are important in transportation and telecommunications. In both settings, when there is traffic between several sources and destinations, there are economic benefits if this traffic is concentrated on some nodes and/or edges that are reliable and cover the much number of user's demand of the network. An electronic device is a node of the network or device placement location of the network that concentrates traffic from several sources and distributes it to the final destinations, or send it to a second device placement location, which distributes it to the final destinations. Optimal selection of certain nodes as the device placement locations, leads us to new field in networks that is called the device placement location *(DPL)*. The DPL is known as the network location problem, it is a major part of the location theory [19,23,28]. In the case of network, the location theory is concerned with the notion of deciding where to place a number of device placement locations. The task of solving a location problem is termed to make a location decision. The optimality of a decision depends on the specific problem. It is, therefore, obvious that there is a wide range of criterions used [22].

2. THE NETWORK MODEL

A graph model used in this paper shown in [13,21,33] is represented by the graph theory. The graph theory is used to give a pictorial representation of any kind of relationship between entities. The relations can vary from being a set of roads between towns to communication links between interconnected nodes. The fundamental concept of the graph theory is the graph which is best thought of as a mathematical object rather than a diagram, even though graphs have a very natural graphical representation [11,21].

3. EDGE-WEIGHTED GRAPH PARAMETERS

Graph parameters represent the weight of the edge and/or nodes [22]. We might want parameters such as the fastest or cheapest way to route a data packet between two computers, the best way which covers the maximum demand of computers, and/or the most reliable way to route a data packet even after a failure in some ways of network. Based on the problem being studied, we may use some or all of these parameters. In this paper, they are associated with each pair of placement locations and specified as the weight of the node and edge, which joins the corresponding pair of nodes. Each node has two weights, namely cost and demand. Cost and reliability are the two weights in each edge. These weights might represent several parameters. For example: the cost of transporting components, the cost of setting up a communication link, the time required to communicate or the reliability to communicate between the pair of placement locations, the demand needed on placement locations, and so on.

4. THE ALGORITHM USED TO MEASURE NETWORK RELIABILITY

We have used the Hao-Orlin Algorithm to measure the network reliability in an undirected graph. It is an implementation of the Gomory-Hu algorithm. A key concept of the Hao-Orlin algorithm is that of frozen layers. A frozen layer is a set of nodes or edges in which different layers are distinct. A node is frozen if its edges form a new frozen layer, otherwise it is alive. We denote the set of frozen nodes by F, the set of frozen edges by Q and the set of alive nodes by A. Initially all nodes are alive. When gap relabeling discovers a set of edges disconnected from the source (or sink), these edges form a new frozen layer. This layer is put on a queue Q of layers [4,10]. It is implemented to run this algorithm in O (nm log (n2/m)) time. The main reason behind choosing the Hao-Orlin algorithm is in [4] on experimental study of recent minimum cut algorithms. This algorithm has been chosen to conduct experiments on evaluating the relative performance of minimum cut algorithms. It has been proved that the Hao-Orlin algorithm has the best performance. More details on the Hao-Orlin algorithm can be found in [4,10,16].

5. THE DEVICE PLACEMENT LOCATION

In this section, we show two algorithms to find the device placement location (DPL) and an assignment of users to device locations. It is based on Hao-Orlin algorithm that takes into account some parameters of the locations, such as the reliability, cost, and demand. In other words, we aim to find a set of K-nodes, from the given set of N nodes, which constitute the device placement location. As a result, the reliability parameter of this location is maximized. The demand of each customer is the greatest assigned demand of this location, and combined cost of placing the device placement locations and users to their assigned device locations is as minimum as possible. The algorithms assume that all nodes are fixed and perfectly reliable. Each pair of nodes can be connected by a single edge. The system fails if any node is not connected, through some path to any other node, i.e. the network does not form at least a spanning tree. The number of devices to be placed is not fixed.

5.1 The DPLRC Algorithm

The basic concept of the DPL with Reliability and Cost (DPLRC) algorithm is very similar to the minimum cost r-rooted 2-height spanning tree method [26]. However, there is a little difference between them. The former computes the minimal cost of the device placement location, considering the fact that the reliability of the device placement location is maximized. It means that the reliability measures between placing the device and routing the users to their assigned device locations is possible with the aid of the Hao-Orlin algorithm. In the latter, the path between the place of the device placement location and the farthest node is at most two edges.

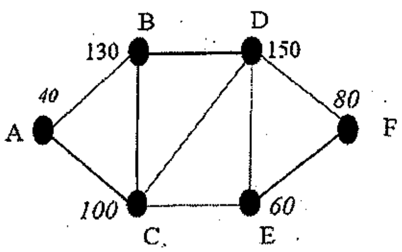

Fig. 1: The 6- Node Graph

5.1.1 THE ALGORITHM DESCRIPTION

Let us assume that a given graph G has a specified node 'r' and it is joined to all the other nodes, setting cru= fu, Fig. 2. The algorithm divides the nodes into two collections: a total assigned collection and a partial assigned collection. An r-rooted n-height spanning tree of G is a spanning tree T that clearly corresponds to a solution for the device placement as follows:

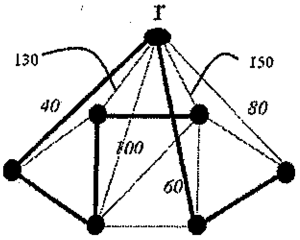

Fig. 2: Solutions for the device placement location for example in Fig. 1

The nodes at height 1 correspond to the nodes where the devices are placed. The node 'u' that is at height n from the root in the tree corresponds as follows:

Let c be the minimum cost of linking u to an existing node, that has a device location, and fu is the cost to place a device at node u. If c < fu, then it is possible to measure the reliability. If (R> maximum reliability threshold Ro), the node u is assigned to an existing node that has a device location as total assigned and put in the total assigned collection; otherwise, it is assigned to an existing node that has a device location as partial assigned and put in the partial assigned collection. When the algorithm detects a node placed at a new location, the partial Collection is tested.

Figure 3 depicts the example of the process of the algorithm suggested in Figs 1 and 2. It also shows the solutions for a graph that consists of six nodes and eight edges. In Fig. 3, a node with a circle contains a device, and the bold edges are used in the solution to service nodes. For the spanning tree model, the bold edges are the edges in the solution. In this figure, nodes A and E serve node B, it is also partially assigned to nodes A and E with maximum reliabilities of 0.50 and 0.40 respectively. Other nodes are totally assigned to nodes A and/or E, with their reliability greater than the maximum reliability threshold (in this example the maximum reliability threshold is 0.5).

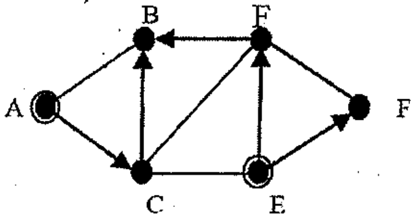

Fig. 3 The example of the process of the algorithm suggested in Fig. 3.2

5.2 THE DPLRD ALGORITHM

To ensure that demand at different users is satisfied to a given placement locations service, network design models usually include cost, demand, distance and/or reliability as service requirements [24,28]. The service requirement is often represented as the minimum cost between the user and demand location. It may be represented also as the maximum reliability with demand, which is defined as "the probability that a required demand can be supplied from source to sink through the multi-state links" (Jose E. Marquez, David W. Coit, and Michael Tortorella 2004). Generally, the algorithm used to determine the DPL using reliability and demand may be formulated as follows:

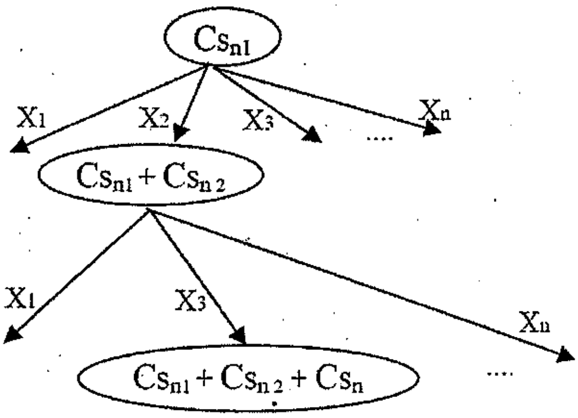

Fig. 4 The formulating of the DPLRDP algorithm.

In Fig. 4, x1...xn denote the nodes of graph G, Sn1 ... Sn3 denoted to the nodes where the devices are placed (the device placement location), and Csn1... Csn13 denote the cost of the device placement location.

5.2.1 THE ALGORITHM DESCRIPTION

This algorithm is based on a given graph containing a set of N nodes and a set of E edges. It also includes some parameters such as i) a set of Di specified demand patterns at the node i, where i=1...n, ii) a set of specified reliability values at each edge rij, where i and j=1...n and iii) a set of Ci specified cost patterns at the node i, where i=1...n. The algorithm chooses one node as a starting node (device placement location) Sn, which can contain the maximum number of demand users Di. The user u that satisfies minimal demand Du with Sn is added to the next starting node set. The assignment of node u to the starting node Sn is with the maximum value denoted as Du×Rxs-u. The algorithm processing will be finished when the total cost of the starting node set Csn exceeds the minimum cost threshold C0. The nodes in starting node set correspond to the nodes where the devices are placed (the device placement location).

6. DESIGN OF OUR SYSTEM

The device placement location consists of three modules, namely Hao-Orlin algorithm, the DPLRC algorithm and the DPLRD algorithm. Hao-Orlin algorithm can be executed, either separately or jointly with the DPLRC algorithm and the DPLRD algorithm. We have adopted a graphical notation known as UML [9]. It is describing the system more precisely than English is, but less detailed than programming code. Our system is divided into two groups of UML diagrams. The Use Case diagram, the class diagram, and sequence diagram. Fig. 5 depicts The Use Case diagram of our system.

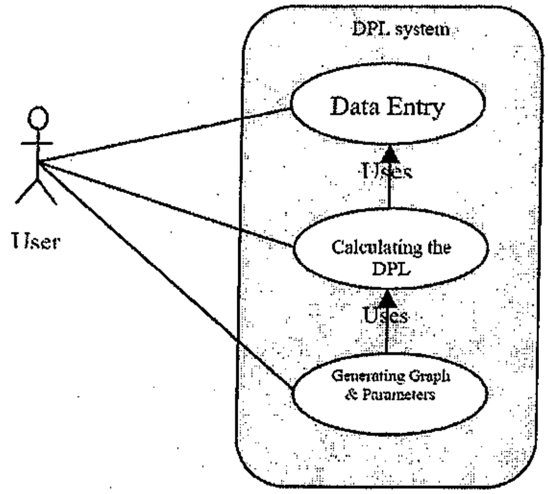

Fig. 5: The Use Case diagram for the DPL system.

7. TESTING THE SYSTEM

In recent years, the unit testing process has become more popular and more structured than before. In computer programming, unit testing is a test (often automated) that ascertains whether or not the individual units of source code are working properly. A unit is the smallest testable part of an application. In object-oriented programming, the smallest unit is a method, which may belong to a base/super class, abstract class or a derived/child class [1]. We have used the Unit Testing Support of Visual Studio 2008 to test the classes of our system. With the aid of the Unit Testing support, it is very easy for developers to create, execute and repeat unit test cases.

8. RESULTS

We will test the DPLRC and DPLRD algorithms using the three network examples. The first two networks are the 13-node set which can be found in [14,26] and the 30-node set which can be found in [15]. The third network is an example of the 125-node set, which is used to experiment with the synthetic data.

In this paper, we show only the results obtained for the DPLRC & DPLRD algorithms with the synthetic data (Tables 2 & 3). In order to test our algorithms, we randomly generated 26 instances on 125-node network for each algorithm. We start the synthetic data experiment with a network whose size is 50 nodes. Later, the same is repeated with networks bearing 75, 100, and 125 nodes. The transportation cost Cij weight is randomly generated in the interval [20, 100], the reliability of edges is distributed in [0, 1], and the demand Di and fixed cost Ci weights are randomly generated in the interval [50,200]. Tables 1 and 2 summarize the final results obtained for the DPLRC and the DPLRD algorithms respectively.

Tables 1 and 2 summarize the final results obtained for the DPLRC and the DPLRD algorithms respectively. Both these tables share some common characteristics: i) the #Nodes column gives the total number of nodes in the network ii) the column marked # Edges gives the total number of edges in the network. iii) the column marked #DPL gives the total number of the device placement locations. iv) the column marked CPU Time gives the total number of CPU seconds required.

In Table 1, the column marked #TC gives the total cost of the DPL. The numbers of the total and partial assigned users are shown by the columns marked #TA Users and #PA Users.

In Table 2, the column marked #DPL with the cost threshold gives the total number of the DPLs whose minimum cost thresholds are set to 500, 800, 1500, and 2000 in columns C-thr1, C-thr2, C-thr3, and C-thr4 respectively.

Table 1: The final result of the DPLRC Algorithm for synthetic data.

#Nodes	#Edges	#DPL	#TA Users	#PA Users	#TC	CPU time
50	40,30,...	>=24	12	47	1685	05.00
	50	21	20	35	1290	04.27
	70	17	18	38	960	07.84
	90	15	13	66	906	13.88
	100	10	18	57	432	11.64
	120	8	12	83	489	13.94
	150	7	22	54	55	12.80
	200	5	14	57	110	25.13
75	50,40,...	>=29	17	73	1683	29.73
	70	24	26	69	1424	24.81
	90	23	23	72	1083	24.64
	100	22	24	82	1036	29.63
	120	19	30	83	708	32.39
	150	18	30	91	1159	50.34
	200	15	32	89	917	49.98
100	70,50,...	>=39	33	90	2430	33.02
	90	39	30	91	1982	36.42
	100	36	31	90	2042	37.56
	120	35	27	95	1835	48.19
	150	32	38	109	1683	60.25
	200	26	36	122	1473	89.77
125	70,50,...	>=45	42	106	2960	67.67
	90	40	47	119	2134	66.64
	100	43	54	95	2357	62.97
	120	39	36	126	1496	71.06
	150	34	54	124	985	93.39
	200	28	58	140	1678	140.88

Table 2: The final result of the DPLRC Algorithm for synthetic data.

#Nodes	#Edges	#DPL with Cost threshold				Average of CPU time
		C-thr1	C-thr2	C-thr3	C-thr4	
50	50	4	8	14	17	1.64
	70	6	9	14	19	1.75
	90	5	8	13	17	1.88
	100	4	7	12	16	1.92
	120	4	6	12	15	2.05
	150	5	7	13	18	2.29
	200	5	7	12	16	2.51
75	50	3	7	13	17	8.61
	70	4	6	11	14	8.48
	90	5	7	12	16	8.25
	100	4	6	11	15	8.30
	120	4	7	12	16	8.42
	150	5	7	11	15	8.75
	200	5	8	13	18	9.43
100	50	4	6	10	14	21.67
	70	3	5	12	17	21.64
	90	4	7	11	15	21.72
	100	5	6	12	16	21.74
	120	3	5	10	14	22.42
	150	5	7	12	16	22.92
	200	4	6	13	18	23.53
125	50	4	7	12	16	49.11
	70	5	7	13	17	49.03
	90	4	7	12	15	49.17
	100	4	6	10	14	49.04
	120	4	7	12	15	49.12
	150	4	6	11	15	50.06
	200	3	5	9	13	52.07

9. DISCUSSIONS OF RESULTS

In this section, we discuss the relationship between the nodes, edges, and the effectiveness of the cost on the increase in the DPls with the aid of the tradeoff curves and tables. However, from Fig. 6, we can see that if the total number of nodes increases, then the number of DPLs. From Fig. 7, an increase in the total number of edges leads to a decrease in the number of DPLs.

From Fig. 8, the minimum cost thresholds influence the DPLRD algorithm.

Figures 9, 10 and 11 given below illustrate the performance of the DPLRC and DPLRD algorithms. We measure the CPU time taken to find the DPLs when we increase the total number of nodes and edges. From Figs. 5.6, 5.7 and 5.8 we learn that the CPU time increases if the total number of nodes and edges increases in both algorithms.

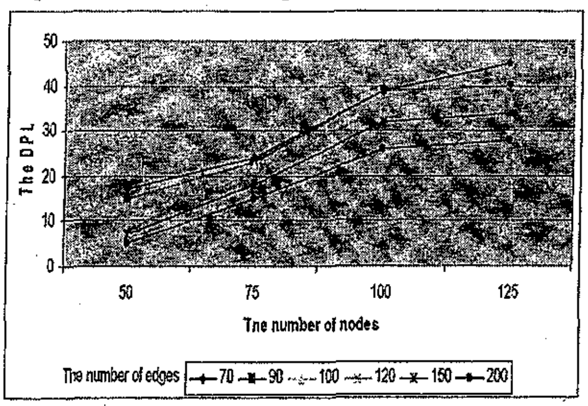

Fig. 6 The DPLRC algorithm for the synthetic data experiment (Number of nodes vs. DPL)

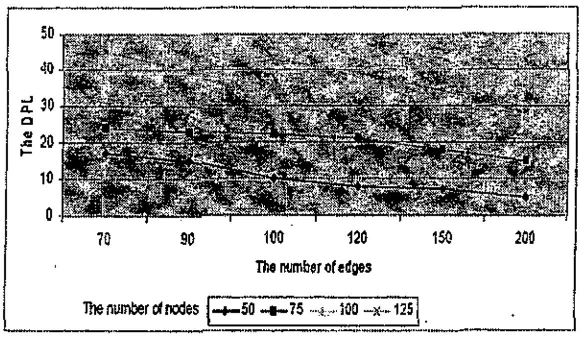

Fig. 7 The DPLRC algorithm for the synthetic data experiment (Number of edges vs. DPL)

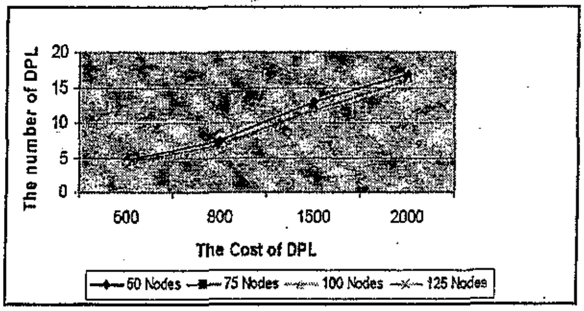

Fig. 8 The minimum cost thresholds influence on the DPLRD algorithm.

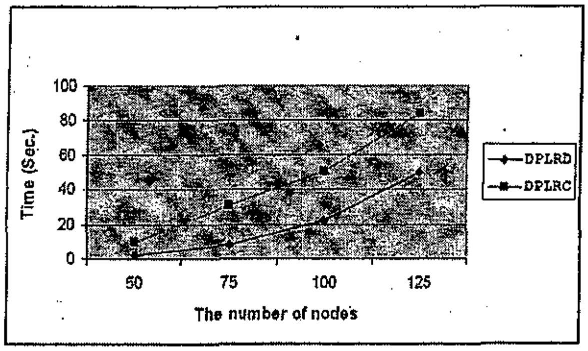

Fig. 9 The synthetic data experiment (number of nodes vs. time)

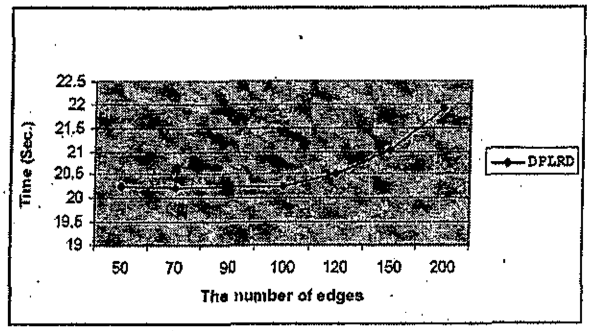

Fig. 10 The synthetic data experiment (number of nodes vs. time)

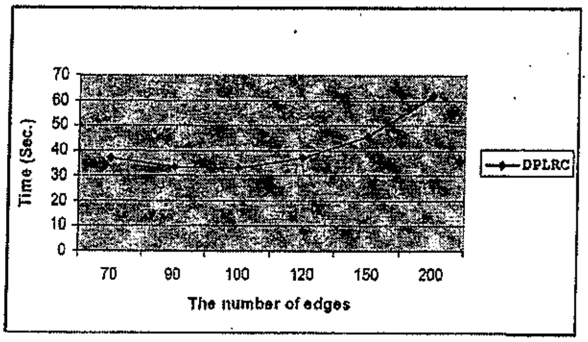

Fig. 11 The synthetic data experiment (number of edges vs. time)

10. CONCLUSION

We have accomplished the objectives of this research by implementing the DPLRC Algorithm with the reliability and cost parameters and the DPLRD Algorithm with the reliability and demand parameters. The Hao-Orlin Algorithm has been used to measure the network reliability in an undirected graph. All the three algorithms mentioned above have been analyzed and designed by the use of UML. They are implemented by the use of C#.NET 2005 programming language. Moreover, they have been tested with the three network examples (two networks as real data and one network as synthetic data). The synthetic data randomly generated 26 instances on a 125-node network for each algorithm. The results obtained from the experiments are summarized as follows:

- The DPLRC algorithm varies according to the total number of nodes and edges.
- The minimum cost thresholds influence the DPLRD algorithm.
- The DPLRC & DPLRD algorithms depend on the Hao-Orlin Algorithm that is considered as the best exact algorithm used to compute the network reliability.
- In all the experiments that we conducted, the DPLRD algorithm has produced fewer numbers of DPLs than the DPLRC algorithm.

11. REFERENCES

[1] A. Hunt, D. Thomas and M. Hargett, (2007) Pragmatic Unit Testing in C# with NUnit. Publisher: Pragmatic Bookshelf Pages: 239 ISBN 10: 0-9776166-7-3.

[2] A. O. Balan, (2003) An Enhanced Approach To Network Reliability Using Boolean Algebra. An Honors Thesis presented to the Departments of Computer Science and Mathematics of Lafayette College.

[3] C. Lucet and J. F. Manouvrier, (1997) Exact Methods To Compute Network Reliability. Lehigh University of Technologie of copie'gne, UMR CNRS 6599 Heudiasyc.

[4] C. S. Chekuri, A. V. Goldberg, D. R. Karger, M. S. Levine and C. Stein, (1998) Experimental Study of Minimum Cut Algorithms.

[5] D. B. Johnson and L. Raab, (1994) Complexity of Network Reliability and Optimal Resource Placement Problems. Vol. 23, No. 3, Pp.

[6] D. L. Deeter and A. E. Smith, (1998) Economic Design of Reliable Networks. Department of Industrial Engineering, University of Pittsburgh, Accepted to IIE Transactions, Special Issue on Economics of Reliability Engineering.

[7] D. Medhi, (1999) Network Reliability and Fault Tolerance. Department of Computer Networking University of Missouri–Kansas City, Wiley Encyclopedia of Electrical and Electronics Engineering, Vol. 14, [Ed. J. G. Webster], pp. 213-218, 1999.

[8] D. P. Bertsekas and J. N. Tsitsiklis, (2002) Introduction to Probability, Athena Scientific, Belmont, Massachusetts, ISBN 1-886529-40-X.

[9] F. A. lotou, M. Schoellmann, Slater J. and Wu C., (2003) Professional UML with Visual Studio.Net: Unmasking Visio for Enterprise Architeets. Wrox Press Ltd, Birmingham.

[10] H. Jianxiuasup and O. James, (1994) A Faster Algorithm for Finding The Minimum Cut In A Directed and An Undirected Graph. Pages: 424 – 446, ISSN: 0196-6774.

[11] J. L. Gross and J. Yellen, (2006) Graph Theory and Its Applications. CRC Press ISBN 158488505X, 9781584885054779. Edition: 2nd ed. Material: 779 p.

[12] K. Marcus,(1997) A new approach to the simple plant location problem. The Pennsylvania State University CiteSeer Archives, United States (CiteSeerPSU):142788.

[13] K. P. Hui, (2005) PhD Thesis in title Network Reliability Estimation, University of Adelaide, Faculty of Engineering, Computer and Mathematical Sciences, School of Mathematical Sciences.

[14] L. E. Miller, (2004) Evaluation Of Network Reliability Calculation Methods, Report JC-2097-FF under contract DAAL02-92-C-0045.

[15] L. V. Snyder and M. S. Daskin (2004) Reliability Models for Facility Location, Department of Industrial & Systems Engineering, Lehigh University, and Technical Report #04T-016.

[16] M. Jünger, G. Rinaldi and S. Thienel, (1997) Practical Performance of Efficient Minimum Cut Algorithms, report No. 97.271, University ZU KÖLN.

[17] M. L. Shooman, (2002) Reliability of Computer Systems and Networks, Polytechnic University, ISBN 0-471-22460-X.

[18] M. S. Lin, (1994) The Reliability Analysis on Distributed Computing Systems, Institute of Computer Science and Information Engineering, National Chiao Tung University Hsinchu, Taiwan

[19] N. Stefan, and P. Justo, (2005) Location Theory A Unified Approach. XXII, 437 p. 116 illus., Hardcover. ISBN: 978-3-540-24321-2.

[20] Q. Wang, K. Xu, T. Glen and H. Hossam, (2007) Device Placement for Heterogeneous Wireless Sensor Networks: Minimum Cost with Lifetime Constraints. IEEE Transactions On Wireless Communications. ISSN 1536-1276, Vol. 6, No7, Pp. 2444-2453 [10 Page(S) (Article)] (22 Ref.), Publisher Institute of Electrical and Electronics Engineers.

[21] R. Diestel, (2000) book Graph Theory II. Faculty of Mathematics and Computer Science. 2nd Edition, Academic Press, Handbook of Combinatorics. ISBN 0-387-98976-5.

[22] R. Martin, (2008) Complexity of Algorithms (COMP 202). Lecture Notes in Department of Computer Science, University of Liverpool.

[23] R. S. Chen, D. J. Chen and Y. S. Yeh, (1994) Reliability Optimization in the Design of Distributed Computing Systems. Institute of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu, Taiwan, R.O.C.

[24] S. Aksel, (2005) Network Performance. Department of Computer Science University College London.

[25] S. Alumura, and B. Y. Kara, (2006) Network Hub Location Problems, Department of Industrial Engineering, Bilkent University, Ankara, Turkey.

[26] S. Boyd and K. Marcus. (1997). The Economic Addition of Functionality to A Network. The departments of Computer Science Pennsylvania State University, CiteSeer Archives. HPCN Europe 1997: 688-697.

[27] S. Boyd, K. Marcus, (1997) A New Approach For A Restricted Concentrator Location Problem. The departments of Computer Science Pennsylvania State University, CiteSeer Archives.

[28] S. O. Krumke, (1996) On the Approximability of Location and Network Design Problems. Faculty of Mathematic and Informatics, university of Wurzburg.

[29] S. Robinson, C. Nagel, J. Glynn, M. Skinner, K. Watson and B. Evjen, (2004) Professional C#. ISBN: 0-7645-5759-9, Published by Wiley Publishing, Inc.

[30] S. Salcedo-Sanz and X. Y. Fellow, (2004) A Hybrid Hopfield Network-Genetic Algorithm Approach for the Terminal Assignment Problem. IEEE Transactions on Systems, Man, and Cybernetics—Part B: Vol. 34, No. 6.

[31] T. B. Brecht, (1985) Lower Bounds for Two-Terminal Network Reliability. The Departments of Mathematics in Computer Science University of Waterloo.

[32] T. Dasic and B. Djordjevic, (2002) PhD Thesis in Method for water distribution systems reliability evaluation. Faculty of Civil Engineering, Belgrade, Yugoslavia.

[33] T. Wolle, (2005) Computational Aspects of Treewidth Lower Bounds and Network Reliability. Universiteit Utrecht. ISBN 90-393-3972-4 .158 pag.

[34] Y. K. Tung and Cullinane, Jr, (1989) Book Reliability Analysis of Systems. Chapter 9, WWRC- 8 9 – 21.

[35] Y. Shpungin, (2006) Combinatorial Approach to Reliability Evaluation of Network with Unreliable Nodes And Unreliable Edges. International Journal of Computer Science Volume 1, number 3, ISSN: 1306-4428.

[17] M. L. Shooman, (2002) Reliability of Computer Systems and Networks, Polytechnic University, ISBN 0-471-22460-X.

[18] M. S. Lin, (1994) The Reliability Analysis on Distributed Computing Systems, Institute of Computer Science and Information Engineering, National Chiao Tung University Hsinchu, Taiwan

[19] N. Stefan, and P. Justo, (2005) Location Theory A Unified Approach. XXII, 437 p. 116 illus., Hardcover. ISBN: 978-3-540-24321-2.

[20] Q. Wang, K. Xu, T. Glen and H. Hossam, (2007) Device Placement for Heterogeneous Wireless Sensor Networks: Minimum Cost with Lifetime Constraints. IEEE Transactions On Wireless Communications. ISSN 1536-1276, Vol. 6, No7, Pp. 2444-2453 [10 Page(S) (Article)] (22 Ref.), Publisher Institute of Electrical and Electronics Engineers.

[21] R. Diestel, (2000) book Graph Theory II. Faculty of Mathematics and Computer Science. 2nd Edition, Academic Press, Handbook of Combinatorics. ISBN 0-387-98976-5.

[22] R. Martin, (2008) Complexity of Algorithms (COMP 202). Lecture Notes in Department of Computer Science, University of Liverpool.

[23] R. S. Chen, D. J. Chen and Y. S. Yeh, (1994) Reliability Optimization in the Design of Distributed Computing Systems. Institute of Computer Science and Information Engineering, National Chiao Tung University, Hsinchu, Taiwan, R.O.C.

[24] S. Aksel, (2005) Network Performance. Department of Computer Science University College London.

[25] S. Alumura, and B. Y. Kara, (2006) Network Hub Location Problems, Department of Industrial Engineering, Bilkent University, Ankara, Turkey.

[26] S. Boyd and K. Marcus. (1997). The Economic Addition of Functionality to A Network. The departments of Computer Science Pennsylvania State University, CiteSeer Archives. HPCN Europe 1997: 688-697.

[27] S. Boyd, K. Marcus, (1997) A New Approach For A Restricted Concentrator Location Problem. The departments of Computer Science Pennsylvania State University, CiteSeer Archives.

[28] S. O. Krumke, (1996) On the Approximability of Location and Network Design Problems. Faculty of Mathematic and Informatics, university of Wurzburg.

[29] S. Robinson, C. Nagel, J. Glynn, M. Skinner, K. Watson and B. Evjen, (2004) Professional C#. ISBN: 0-7645-5759-9, Published by Wiley Publishing, Inc.

[30] S. Salcedo-Sanz and X. Y. Fellow, (2004) A Hybrid Hopfield Network-Genetic Algorithm Approach for the Terminal Assignment Problem. IEEE Transactions on Systems, Man, and Cybernetics—Part B: Vol. 34, No. 6.

[31] T. B. Brecht, (1985) Lower Bounds for Two-Terminal Network Reliability. The Departments of Mathematics in Computer Science University of Waterloo.

[32] T. Dasic and B. Djordjevic, (2002) PhD Thesis in Method for water distribution systems reliability evaluation. Faculty of Civil Engineering, Belgrade, Yugoslavia.

[33] T. Wolle, (2005) Computational Aspects of Treewidth Lower Bounds and Network Reliability. Universiteit Utrecht. ISBN 90-393-3972-4 .158 pag.

[34] Y. K. Tung and Cullinane, Jr, (1989) Book Reliability Analysis of Systems. Chapter 9, WWRC- 8 9 – 21.

[35] Y. Shpungin, (2006) Combinatorial Approach to Reliability Evaluation of Network with Unreliable Nodes And Unreliable Edges. International Journal of Computer Science Volume 1, number 3, ISSN: 1306-4428.