Computing all-terminal reliability of stochastic networks with Binary Decision Diagrams

Gary Hardy[1], Corinne Lucet[1], and Nikolaos Limnios[2]

[1] LaRIA, FRE 2733, 5 rue du Moulin Neuf 80000 AMIENS
email:(corinne.lucet, gary.hardy)@u-picardie.fr

[2] LMAC, UTC, BP 20529 60205 COMPIEGNE Cedex
email:Nikolaos.Limnios@utc.fr

Abstract. In this paper, we propose an algorithm based on Binary Decision Diagram (BDD) for computing all-terminal reliability. It is defined as the probability that the nodes in the network can communicate to each other, taking into account the possible failures of network links. The effectiveness of this approach is demonstrated by performing experiments on several large networks represented by stochastic graphs. [1]

Keywords: Network reliability, Binary Decision Diagram (BDD), Stochastic graph.

1 Introduction

A stochastic network is modeled by an undirected graph $G = (V, E)$ where V is the vertex set and E is the edge set. Sites correspond to vertices and links to edges. The all-terminal reliability $R(G)$ is the probability that G remains connected assuming all edges can fail independently with known probability and nodes are perfect. Provan [Provan, 1986] showed that even for planar graphs this problem is still NP-hard. In literature, two classes of algorithms for computing the network reliability can be distinguished. The first class deals with the enumeration of all the minimum paths. The *inclusion-exclusion* or *sum of disjoint products* methods have to be applied since this enumeration provides non-disjoint events. The algorithms in the second class are factoring algorithms improved by reductions. It consists in reducing the size of the network while preserving its reliability. When no reduction is allowed, the factoring method is used. The idea is to choose a component and decompose the problem into two sub-problems: the first assumes the component has failed, the second assumes it is functioning. Satyanarayana and Chang [Satyanarayana and Chang, 1983] and Wood [Wood, 1985] have shown that the factoring algorithms with reductions are more efficient than the classical path or cut enumeration method for solving this problem. This was confirmed by the experimental works of Theologou and Carlier [Theologou and Carlier, 1991].

[1] Acknowledgment: This research was supported by the Conseil Regional de Picardie.

This paper is organized as follows. First, we give a brief introduction to BDD in Section 2. Then, in Section 3 we proposed a description of our method for computing network reliability. In Section 4, we introduce an other important reliability measure (Birnbaum importance measure) and its fast computation via BDD. Next, we present experimental results in Section 5. Finally, we draw some conclusions and outline the direction of futur works in Section 6.

2 Binary Decision Diagram (BDD)

Akers [Akers, 1978] first introduced BDD for representing boolean function. Bryant popularized the use of BDD by introducing a set of algorithms for efficient construction and manipulation of the BDD structure [Bryant, 1992]. Nowadays, BDD are used in a wide range of area, including hardware synthesis and verification, model checking and protocol validation. Their use in the reliability analysis framework has been introduced by Madre and Coudert [Coudert and Madre, 1992b] [Coudert and Madre, 1992a] and developped by Rauzy [Rauzy, 1993]. Sekine and Imai were the first to use the BDD structure in network reliability [Sekine and Imai, 1998]. A BDD is a directed acyclic graph (DAG) based on Shannon's decomposition. The Shannon's decomposition is defined as follows:

$$f = xf_{x=1} + \bar{x}f_{x=0}$$

where x is one of decision variables and $f_{x=i}$ is the boolean function f evaluated at $x = i$.
The graph has two sink nodes labeled with 0 and 1 representing the two corresponding constant expressions. Each internal node is labeled with a boolean variable x and has two out-edges called 0-edge and 1-edge. The node linked by 1-edge represents the boolean expression when $x = 1$, i.e $f_{x=1}$ while the node linked by 0-edge represents the boolean expression when $x = 0$, i.e $f_{x=0}$.
An ordered binary decision diagram (OBDD) is a BDD where variables are ordered according to a known total ordering and every path visits variables in an ascending order. Afterwards, BDDs will be considered as ordered. Leaves of the BDD give the value of f for the assignment corresponding to a path from the root to the leaf. The size of a BDD structure depends critically on variable ordering. Finding an ordering that minimizes the size of BDD is also a NP-complete problem [Friedman and Supowit, 1990].

3 Computing all-terminal reliability

Definitions and notations

A graph G is connected if there exists at least one path between any two vertices. Our network model is an undirected stochastic graph $G = (V, E)$.

x_1	x_2	x_3	f
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1

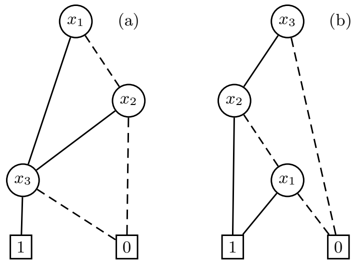


Fig. 1. Function $f(x_1, x_2, x_3) = (x_1 \wedge x_3) \vee (x_2 \wedge x_3)$ represented by its truth table and BDDs with order: (a) $x_1 < x_2 < x_3$ and (b): $x_3 < x_2 < x_1$. A dashed (solid) line represents the value 0 (1).

Each edge e_i of E ($i \in \{1, 2, \ldots, m\}$ where $m = |E|$) can fail independently with known probability q_i ($p_i = 1 - q_i$ is the functioning probability of e_i) and we consider that vertices of G are perfectly reliable. A state $\mathcal{G}$ of the stochastic graph G is denoted by $(x_1, x_2, \ldots, x_m)$ where x_i stands for the state of edge e_i, i.e, $x_i = 0$ when edge e_i fails and $x_i = 1$ when it functions. The associated probability of $\mathcal{G}$ is defined as:

$$Pr(\mathcal{G}) = \prod_{i=1}^{m} (x_i.p_i + (1 - x_i).q_i)$$

At each state $\mathcal{G}$ is associated a partial graph $G(\mathcal{G}) = (V, E')$ such that $e_i \in E'$ if and only if $e_i \in E$ and $x_i = 1$. The all-terminal reliability can be define as follows:

$$R(G) = \sum_{G(\mathcal{G}) \text{ is connected}} Pr(\mathcal{G})$$

We denote by G_{*e} the graph G with contracted edge e and by G_{-e} the graph G with deleted edge e.

Construction of the all-terminal reliability function

Our algorithm follows three steps:

1 The edges are ordered by using a heuristic.
2 The BDD is generated to encode the network reliability.
3 From this BDD, we obtain the all-terminal reliability.

We apply recursively the factoring algorithm in the order of $e_1, e_2, \ldots, e_m$ in a top-down way. The computation process can be represented as a binary tree such that the root corresponds to the original graph G and children correspond to graphs obtained by deletion /contraction of edges. Nodes in the binary tree correspond to subgraphs of G. We use the method introduced by Carlier [Carlier and Lucet, 1996] for represententing graph by partition. It is an efficient way for representing graph and finding isomorphic graphs during the computation process. By sharing the isomorphic subgraphs an expansion tree is modified as a rooted acyclic graph (therefore a BDD).

Sharing isomorphic graphs

Consider that $E_k = \{e_1, e_1, \ldots, e_k\}$ and $\bar{E_k} = \{e_{k+1}, \ldots, e_m\}$. The graphs in the k-th level of the BDD are sub-graphs of G with the edge set $\bar{E_k}$. For each level k, we define the boundary set F_k as a vertex set such that each vertex of F_k is incident to at least one edge in E_k and one edge in $\bar{E_k}$. Then we gather vertices in blocks according the following rule: two vertices s and t of F_k are in the same block if and only if there exists a path made of functioning edges linking s to t. For instance in figure 3(a), in the first level, the boundary set is equal to $\{a, b\}$. G_{*e_1} can be represented by partition [ab] and G_{-e_1} by partition [a][b]. Now, we order partitions in the same level k in order to identify and stock them in an efficient way. We number the partition from 1 to $Bell(|F_k|)$ where $Bell(|F_k|)$ (known as the Bell number) is the theoretical maximum number of partitions in level k. This number grows exponentially with i, consequently the number of classes grows exponentially with the size of the boundary set. From now on, we only manipulate partitions instead of graphs during the all-terminal reliability computation.

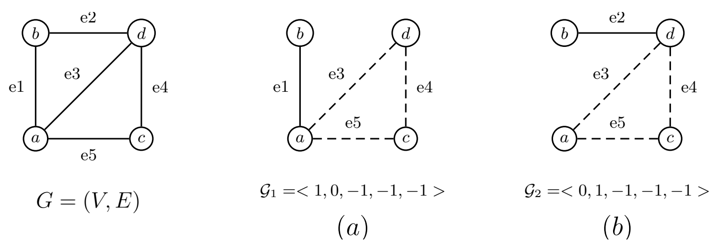


Fig. 2. G($\mathcal{G}_1$) and G($\mathcal{G}_2$) represent sub-graphs in level 2 in the computation process illustred in figure 3(a). G($\mathcal{G}_1$) and G($\mathcal{G}_2$) has the same partition: $[a][d]$ during the computation. $e_i = -1$ means the state of e_i is not yet fixed.

All-terminal reliability computation

In the previous section, BDD of the all-terminal reliability function was constructed. The BDD can be recognized as a graph-based set of disjoint products. Based on the disjoint property of this structure, we can now easily compute the all-terminal reliability of G. Given the non-failure probabilty p_k

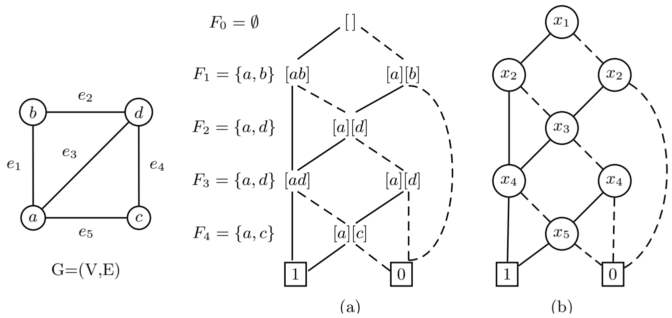


Fig. 3. Graph G and its BDD (b). A dashed (solid) line represents the value 0 (1). (a) illustrates the computation process of the BDD.

($k \in \{1, 2, \ldots, m\}$) of edge e_k, the all-terminal reliability of a BDD-based function f can be recursively obtain by:

$$R(G) = Pr(f = 1) = Pr(x_k.f_{x_k=1} = 1) + Pr(\bar{x_k}.f_{x_k=0} = 1) \ (\textit{disjoint property})$$
$$R(G) = Pr(f = 1) = p_k.Pr(f_{x_k=1} = 1) + q_k.Pr(f_{x_k=0} = 1) \ (\textit{independent property})$$

The reliability is evaluated by traversing the BDD from the root to the leaves.

4 Importance measure

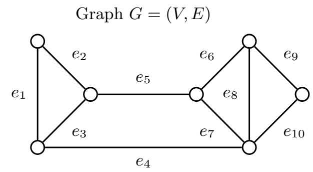
Finding the critical components is also an important issue for reliability analysis and the optimization design of network topology. The aim is to obtain information concerning a component's contribution to the system reliability. The three most used importance measures are: Birnbaum, Critically and Fussell-Vesely. We briefly explain here the Birnbaum importance measure. The Birnbaum importance measure of a component e_k is the probability that a system is in a critical state with respect to e_k and that the failure of component e_k will then cause the system to fail. Here, the Birnbaum importance measure of edge e_k, noted I_k^B, is defined as:

$$I_k^B = Pr(f_{x_k=1} = 1) - Pr(f_{x_k=0} = 1)$$

The figure 4 shows the importance measures for the reliability graph G.

5 Experimental results

Computations are done by using Pentium 4 with 512 MB memory. Our program is written in C language. The experimental results are shown in Tables 1 and 2. The unit of time is in second. The running time includes the BDD generation and the all-terminal reliability computation. The heuristic



ordering: e_5,e_1,e_2,e_9,e_{10},e_4,e_6 ,e_3,e_8,e_7

e_k	q_k	I_k^B
e_1	*0.3*	*0.2849*
e_2	*0.3*	*0.2849*
e_3	*0.3*	*0.1611*
e_4	*0.4*	*0.2285*
e_5	*0.05*	*0.3534*
e_6	*0.4*	*0.1872*
e_7	*0.4*	*0.1497*
e_8	*0.2*	*0.1543*
e_9	*0.3*	*0.2625*
e_{10}	*0.3*	*0.2625*

Fig. 4. Sensibility analysis of graph G. According to the Birnbaum importance measure, e_5 has the highest degree of contribution to the graph reliability.

used for ordering edges (and so variables in BDD) in the experiments is known as a breadth-first-search (BFS) ordering. We give two characteristics of the generated BDD: its *size* (number of nodes) and its *width* (if $|W_i|$ is the number of nodes in the ith level then the bdd width is: $\max_i |W_i|$). $|F_{max}|$ corresponds to the maximal size of the boundary set during the computation process. The computation speed heavily depends on $|F_{max}|$ and so the edge ordering.

6 Conclusion

A method for evaluating the all-terminal reliability via BDD has been proposed in this paper. Based on this approach, our futur works will focus on computing other kinds of reliability and reusing the BDD structure in order to optimize design of network topology.

References

[Akers, 1978]B. Akers. Binary decision diagrams. *IEEE Trans. On Computers*, vol. C-27:509–516, 1978.

[Bryant, 1992]R. E. Bryant. Symbolic Boolean manipulation with ordered binary-decision diagrams. *ACM Computing Surveys*, 24(3):293–318, 1992.

[Carlier and Lucet, 1996]J. Carlier and C. Lucet. A decomposition algorithm for network reliability evaluation. In *Discrete Applied Mathematics*, volume 65, pages 141–156, 1996.

[Coudert and Madre, 1992a]O. Coudert and J. C. Madre. Implicit and incremental computation of primes and essential primes of boolean functions. In *Proceedings of the 29th ACM/IEEE Design Automation Conference (DAC'92)*, pages 36–39. IEEE Computer Society Press, June 1992.

[Coudert and Madre, 1992b]O. Coudert and J. C. Madre. A new method to compute prime and essential prime implicants of boolean functions. In *Advanced Research in VLSI and Parallel Systems*, pages 113–128, March 1992.

type	n	m	*time*	*size*	*width*	$\|F_{max}\|$
K08	8	28	0.03	2745	405	7
K09	9	36	0.06	10265	1265	8
K10	10	45	0.13	39856	3925	9
K11	11	55	0.52	160793	15105	10
K12	12	66	2.14	673934	652	11
K13	13	78	9.97	2932248	279981	12
K14	14	91	50.00	13227624	1191235	13
K15	15	105	490	61780095	5021561	14

Table 1. Benchmark on complete graphs

type	n	m	*time*	*size*	*width*	$\|F_{max}\|$
6x6	49	84	0.15	39523	858	8
7x7	64	112	0.7	179410	2860	9
8x8	81	144	3.16	797916	9724	10
9x9	100	180	14.75	3495491	33592	11
10x10	121	220	67.94	15137188	117572	12
15x10	176	325	101.2	33360848	117572	12
20x10	231	430	101.2	24249018	117572	12
11x11	144	264	321.3	64959137	416024	13

Table 2. Benchmark on lattice graphs

[Friedman and Supowit, 1990]S. J. Friedman and K. J. Supowit. Finding an optimal variable ordering for binary decision diagrams. *IEEE Trans. On Computers*, vol. C-39:710–713, 1990.

[Provan, 1986]J.S. Provan. The complexity of reliability computations on planar and acyclic graphs. *SIAM J. Computing*, 15(3):694–702, 1986.

[Rauzy, 1993]A. Rauzy. New algorithms for fault tolerant trees analysis. *Reliability Engineering and System Safety*, pages 203–211, 1993.

[Satyanarayana and Chang, 1983]A. Satyanarayana and M.K. Chang. Network reliability and the factoring theorem. *Networks*, 13:107–120, 1983.

[Sekine and Imai, 1998]K. Sekine and H. Imai. Computation of the network reliability (extended abstract). Technical report, Department of Information Science, University of Tokyo, 1998.

[Theologou and Carlier, 1991]O. Theologou and J. Carlier. Factoring and reductions for networks with imperfect vertices. In *IEEE Trans. on Reliability*, volume 40, pages 210–217, 1991.

[Wood, 1985]R.K. Wood. A factoring algorithm using polygon-to-chain reductions for computing k-terminal network reliability. In *Networks*, volume 15, pages 173–190, 1985.