Journal Paper

"Improving the Efficiency on Decision Making Process via BDD"

Advances in Intelligent Systems and Computing

Accepted 21 May 2015

Alberto Pliego Marugán
Ingenium Research Group, Universidad de Castilla-La Mancha
Alberto.Pliego@uclm.es

Fausto Pedro García Márquez
Ingenium Research Group, Universidad de Castilla-La Mancha
FaustoPedro.Garcia@uclm.es

Cite as: Marugán, A.P.; Márquez, F.P.G. In Improving the efficiency on decision making process via bdd, Proceedings of the Ninth International Conference on Management Science and Engineering Management, 2015; Springer: pp 1395-1405.

DOI: 10.1007/978-3-662-47241-5_116

Contents

Chapter 1
Improving the Efficiency on Decision Making Process via BDD

Alberto Pliego Marugán and Fausto Pedro García Márquez

Abstract For a qualitatively and quantitatively analysis of a complex Decision Making (DM) process is critical to employ a correct method due to the large number of operations required. This paper presents an approach employing Binary Decision Diagram (BDD) applied to the Logical Decision Tree. LDT allows addressing a Main Problem (MP) by establishing different causes, called Basic Causes (BC) and their interrelations. The cases that have a large number of BCs generate important computational costs because it is a NP-hard type problem.. This paper presents a new approach in order to analyze big LDT. A new approach to reduce the complexity of the problem is hereby presented. It makes use of data derived from simpler problems that requires less computational costs for obtaining a good solution. An exact solution is not provided by this method but the approximations achieved have a low deviation from the exact.

Keywords Decision making · Efficiency · Logical decision tree · Binary decision diagram

1.1 Introduction

DM processes are done continuously by any firm in order to maximize the profits reliability, etc. or minimize costs, risks, etc. There are softwares to facilitate this task, but the main problem is the capability for providing a quantitative solution when the case study has a large number of BCs. The DM problem is considered as a cyclic process in which the decision maker can evaluate the consequences of a previous decision. Fig. 1.1 shows the normal process to solve a DM problem.

A. P. Marugán (✉) · F. P. G. Márquez
Ingenium Research Group. Escuela Técnica Superior de Ingenieros Industriales, Campus Universitario s/n, 13071 Ciudad Real, Spain
e-mail: Alberto.Pliego@uclm.es
e-mail: FaustoPerdo.Garcia@uclm.es

J. Xu et al. (eds.), *International Conference on Management Science and Engineering Management 2015*,

Advances in Intelligent Systems and Computing XXX, DOI: XXXXXXXXXX

Fig. 1.1 Decision Making Process

1.2 Defining the MP via LDT

This paper considers a new approach based on breaking down the problem into different causes that could lead to non-desired situations. This disaggregation leads to determine the number of BCs and also identifies the manner in which all these BCs are logically interrelated. With this purpose, LDT are proposed in this paper as a useful tool being able to addressing the MP. LDT is introduced as an alternative method to draw a DM, considering the interrelation between each BC [11]), that will take into account the logical operators 'AND' and 'OR' [13]. Annex 1 shows a LDT case study composed by:

1. 1 Top event or MP.
2. 63 Logical gates: they determine the logical relation between BCs.
3. 6 Levels: it is related to the depth of each logical gate.
4. 64 BCs: possible causes of the MP.

It is showed only with OR gates because its topology will be changing in the experiments in order to analyse different scenarios

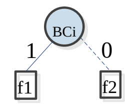
1.3 LDT to BDD Conversion

LDT conversion to BDD provides some advantages in terms of efficiency and accuracy see [1, 5, 10, 15]. BDD helps to show the occurrence of a serious issue in the business in a disjoint form, which indeed provides an advantage from the computational point of view.

BDD is a directed graph representation of a Boolean function, where equivalent Boolean sub-expressions are uniquely represented. A directed acyclic graph is a directed graph with no cycles, i.e. to each vertex v there is no possible directed path that starts and finishes in v. It is composed of some interconnected nodes in a way that each node has two vertices. Each can be either a terminal or non-terminal vertex. BDD is a graph-based data structure whereby the occurrence probability of a certain problem in a DM is possible to be achieved. Each single variable has two branches: 0-branch corresponds to the cases where the variable is 0; 1-branch cases are those where the event is being carried out and corresponds when the variable is 1.

The transformation from DT to BDD is achieved applying some mathematical algorithms. Ite (If-Then-Else) conditional expression is one of the BDD's cornerstones [2], see Fig. 1.2: Fig. 1.2 is defined as: "If BC_i variable occurs, Then f_1, Else

Fig. 1.2 ITE applied to BDD



f_2". The solid line always belongs to the ones as well as the dashed lines to the zeroes. Having into account the Shannon's theorem it can be obtained the following expression:

$$f = BC_i f_1 + \overline{BC_i} f_2. \quad (1.1)$$

It can be also expressed as:

$$f = BC_i f_1 + \overline{BC_i} f_2 = ite(b_i, f_1, f_2). \quad (1.2)$$

In the transformation from DT to BDD is necessary to establish a correct ranking of BCs, further detailed information about the conversion and variable ordering methods can be found in [8]. In this paper only AND and OR gates are used in the LDTs presented to express the interrelation between the BCs. Fig. 1.3 shows the conversion from LDT to its corresponding BDD using the following order for BCs: $BC_1 > BC_2 > BC_3 > BC_4$ Once the conversion from DT to BDD is done, it is possible to obtain an accurate expression of the probability of occurrence of the MP (Q_{MP}) by assigning a probability value to each BC.

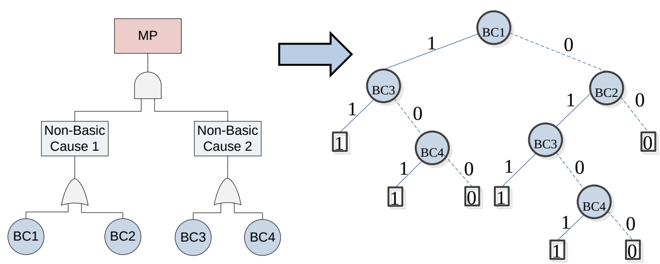


Fig. 1.3 Conversion from LDT to BDD

1.3.1 Cut-sets and The Analytical Expression from BDD

Cut-sets (CS) turn into an important concept when referring to BDDs. They are the paths ”from the top to the ones” that provide significant information due to the fact that the probability of occurrence of the MP could be achieved from them. The following CSs have been obtained from the BDD in Fig. 1.3.

$$
\begin{aligned}
CS_1 &= \{BC_1, BC_3\},\\
CS_2 &= \{BC_1, \overline{BC_3}, BC_4\},\\
CS_3 &= \{\overline{BC_1}, BC_2, BC_3\},\\
CS_3 &= \{\overline{BC_1}, BC_2, \overline{BC_3}, BC_4\}.
\end{aligned}
$$

The MP probability is possible to be achieved due to the fact that the different paths (CSs) are mutually exclusive and may be expressed as the sum of probabilities of all the BDD paths, i.e. an analytic expression consisting of the sum of each analytic expression that forms the CSs. This expression will represent the utility function in the DM process.

1.3.2 The Importance of A Correct Ranking of BCs

It is very important to set a proper ranking for the BCs before carrying out a conversion from LDT to BDD. This ranking determines the size of the resulting BDD and, consequently, the number of CSs. An inefficient variable ranking usually produces a large BDD. If the number of CSs increases, then the computational time required for calculating the MP probability of will rise. The problem of finding the optimal variable ranking is NP-complete and it cannot be solved on a reasonable time, therefore, heuristic methods are widely used in order to find an efficient ranking. These methods do not provide an optimal solution but a good enough one. The main methods are described below:

1. The Top-down-left-right (TDLR) method generates a ranking of the events by ordering them from the original FT structure in a top-down and then left-right manner [3]. The listing of the events is initialized, at each level, in a left to right path adding the BCs found in the ordering list.
2. The Depth First Search (DFS) approach goes from top to down of a root and each sub-tree from left to right [6]. This procedure is a non-recursive implementation and all freshly expanded nodes are added as last-input last-output process.
3. The Breadth-First Search (BFS) algorithm begins ordering all the basic events obtained expanding from the standpoint by the first-input first-output procedure [9]. The LDT is read from left to right and the events are ranked according to the order in which they are found. It must be stated that if a repeated event is found it must be ignored.

4. The "Level" method creates the ranking of the events according to the level of them [4]. The level of any BC is understood as the number of the gates that has higher up a tree until the MP. In case that two or more events have the same level, the event will have highest priority if it appears early in the tree.
5. The AND criterion establishes that the importance of the BC is based on the "AND" gates that are between the k event and the MP, because these gates imply that there are redundancies in the system [18]. Consequently, a BC under an "AND" gate can be viewed as less important because it is independent to other BCs occurrences. Furthermore:
   BCs with the highest number of "AND" gates will be ranked at the end.
   In case of duplicated basic events, the event with less "and" gates has preference.
   BCs with the same number of "and" gates can be ranked as the TDLR method approach.
6. Other methods, such as Weights method, Method of flows, Method of fathers, Level Method, Heuristic method based on the structural importance are described in references [17], [7], [12], [4] and [20].

There is not a specific heuristic method appropriate for all the LDTs. Some methods are more appropriate than others depending on the logical function. The most appropriate method should be chosen for each case. The heuristic methods described hereby are static. There are also dynamic heuristic methods however, they are not suitable for large or complex LDTs. They present some drawbacks such as they need to store in memory the BDD or a part of it [14].

1.3.3 Evaluation of BDDs

The information that can be gathered from a BDD is:
MCSs and Minimal Path Sets.
Probability of occurrence of the top event.
Sensitivity analysis and importance measures.

Fig. 1.4 shows the different steps required to carry out a complete LDT via BDD analysis. The information that this analysis provides is very useful but not easy to obtain. The computational cost of the evaluation of the BDDs is one of the most important obstacles when a large LDT is required to be analysed. In the following

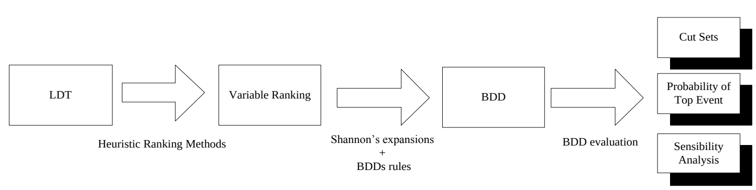


Fig. 1.4 Evaluation of BDDs

section, an estimation of the results of the LDT in Annex 1 is performed to find the MP probability depending on the topology of the tree. It is important to remark that this approach aims to reduce computational time when a large LDT has to be solved but it can be useful only if an exact result is not required.

1.4 New Approach to Reduce Computational Cost

The DM problem described is a NP-hard type problem and, therefore, for a large number of BCs, or a complex topology, it can be not recommended to find a solution. This paper presents a novel approach for finding a good solution minimizing the computational cost. This approach is based on the logical gates, especially the AND gates, the number and the position on the tree (level), and their effects to the solution and the computational cost of the system. The reference solutions, or experimental solutions, are obtained in simple systems, where it can be extrapolated to complex systems via polynomial regression functions. These functions are setting according to the reference solutions, where it will be more precise with more reference solutions.

Fig. 1.5 shows the probabilities and CSs for different LDT cases studies in Annex 1. The probability of occurrence of MP and the CSs are obtained for different amounts of AND gates in each level.

The LDT has been calculated for the cases marked in black in Fig. 1.5, and red are estimated results. The calculated results have been obtained using the aforementioned AND criterion to ranking the BCs. The estimations have been obtained through polynomial expressions, where the polynomial degree depends on the number of experimental points obtained. The experimental solutions have been obtained using the algorithms developed by [2]. Fig. 1.6 shows the results of probabilities

Probability of Occurrence of MP						Number of AND Gates	Number of Cut-Sets					
Level 1	Level 2	Level 3	Level 4	Level 5	Level 6		Level 6	Level 5	Level 4	Level 3	Level 2	Level 1
0,0756	0,291	0,3864	0,4313	0,4531	0,4638	1	63	64	72	112	288	1024
	0,0436	0,2836	0,3846	0,4308	0,453	2	63	72	144	672	4608	
		0,1637	0,3341	0,4077	0,4419	3	65	104	536	5840		
		0,027	0,2795	0,3837	0,4306	4	71	208	2528	15616		
			0,2205	0,3587	0,4191	5	85	528	7400			
			0,1578	0,3326	0,4074	6	115	1496	16432			
			0,0911	0,3036	0,3954	7	177	3673	30904			
			0,0204	0,2752	0,3832	8	303	7836	52095			
				0,2458	0,3727	9	527	14952				
				0,2154	0,3604	10	896	26177				
				0,184	0,3479	11	1463	42860				
				0,1516	0,3352	12	2294	66544				
				0,1182	0,3223	13	3462	98961				
				0,0838	0,3092	14	5048	142036				
				0,0484	0,2959	15	7144	197888				
				0,012	0,2824	16	9850	268825				
					0,2687	17	13276					
					0,2548	18	17540					
					0,2407	19	22770					
					0,2264	20	29103					
					0,2119	21	36684					
					0,1972	22	45669					
					0,1823	23	56221					
					0,1672	24	68513					
					0,1519	25	82729					
					0,1364	26	99058					
					0,1207	27	117701					
					0,1048	28	138869					
					0,0887	29	162778					
					0,0724	30	189658					
					0,0559	31	219744					
					0,0392	32	253283					

Fig. 1.5 Experimental results and estimations

found exactly (E) by BDD and the predicted (P) results found by new approach. It is observed that the probability is indirectly proportional to the number of AND gates, and proportional to the level, which is expected. Moreover, the consequences of adding a new AND gate is indirectly proportional to the level. Fig. 1.6 is also plotted (black curve) the absolute deviation expressed as abs$((E-P)/P)$. The deviation is proportional to the number of gates, and with values always inferior to 0,45%. It demonstrates that the accuracy of the solutions founds by the new approach is in every case very good. The deviation has been estimated for different levels and number of AND gates. It has been estimated through quadratic polynomial expression. It is useful in order to know approximately the accuracy of the probability estimated and showed in Fig. 1.7. A similar to the study presented in Fig. 6 has been done taking into account the number of CSs, and showed in Fig. 1.8. The number of CSs is larger in each level when the number of AND gates increase, and the number of CSs is smaller when the level is larger taking into account the same number of AND gates. The error is not as relevant for CSs than for the probabilities, because is the same independently of the number of CSs. It is relevant in order to estimate the computational cost for solving the problem. Exponential expressions have been used to evaluate the size of the CSs.

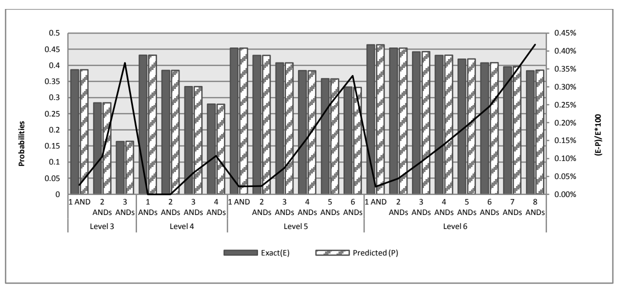


Fig. 1.6 Probability analysis

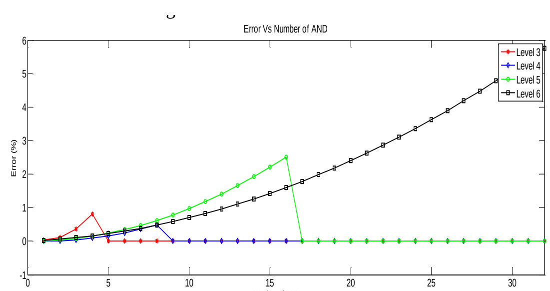


Fig. 1.7 Deviation vs. number of AND

1.5 Conclusions

DM via LDT and the conversion to BDD is presented in this paper. This approach often requires decision maker to obtain a complex analytic expression of the occurrence probability for a MP. The complexity of this expression depends of the number of BCs and the topology of LDT.

An analysis of different scenarios regarding to the AND gates and levels is done in this paper. It has been demonstrated that the number of CSs, and therefore the computational cost, can increase significantly and do not viable to find a solution in a reasonable time. One of the most influent factors in the computation cost is the ranking of the BCs. In this paper, different heuristic methods are presented in order to establish a proper order for them.

This paper also presents a novel approach for not complex topologies that allows, employing simplex regression techniques, to estimate the solution of different scenarios for a LTD problem. Polynomial and exponential expression have been used with this purpose. It leads solutions with very good accuracy associated to scenarios associated to a large number of CSs. It leads therefore to reduce the computational cost for solving the problem.

Appendix

Acknowledgements The work reported herewith has been financially supported by the Spanish Ministerio de Economła y Competitividad, under Research Grant DPI2012-31579.

References

1. Akers S (1978) Binary decision diagrams. IEEE Transactions on Computers 27(6):509–516

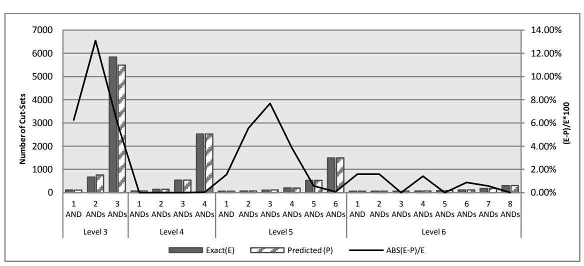


Fig. 1.8 CS analysis

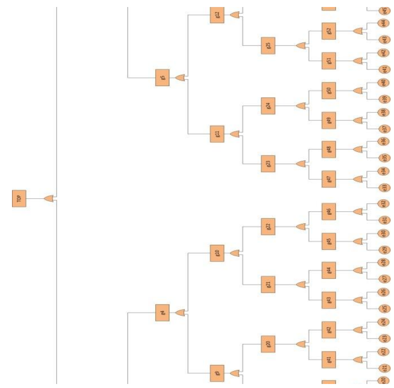


Fig. 1.9 Annex 1

2. Artigao E (2009) Anlisis de rboles de fallos mediante diagramas de decisin binarios. PFC Universidad de Castilla La Mancha
3. Bartlett L (2003) Progression of the binary decision diagram conversion methods. proceedings of the 21st international system safety conference. Ottawa, Westin Hotel 48:116–25
4. Bartlett L, Andrews J (2001) An ordering heuristic to develop the binary decision diagram based on structural importance. Reliability Engineering and System Safety 72:31–38
5. Bryant R (1986) Graph-based algorithms for boolean functions using a graphical representation. IEEE Transactions on Computing 35:677–691
6. Cormen T, Leiserson CE, Rivest RLea (2001) Introduction to algorithms. MIT Press and McGraw-Hill pp 540–9
7. Ericson C (1999) Fault tree analysis: A history. Proceedings of the 17th International Safety Systems Conference
8. Garcła Mrquez F, H M (2012) Introducci*ó*n an*á*lisis de *Á* rboles de fallos: Empleo de bdds. Editorial Acadmica Espa*ñ*ola
9. Jensen R, Veloso M (2000) Obdd-based universal planning for synchronized agents in non-deterministic domains. Journal of Artificial Intelligence Research pp 189–226
10. Lee CY (1959) Representation of switching circuits by binary decision diagrams. Bell System Technology 38:985–999
11. Lopez D, Van Slyke WJ (1977) Logic tree analysis for decision making. omega. The Int Journal of Management Science 5
12. Malik S (1988) Logic verification using binary decision diagrams in logic synthesis environment. In Proceedings of the IEEE International Conference on Computer Aided Design pp 6–9
13. Mallo C, Merlo J (1995) Control de gesti*ó*n y control presupuestario. McGraw-Hill pp 5–30
14. Minato S, Ishiura N, Yajim S (1990) Shared binary decision diagrams with attributed edges for efficient boolean function manipulation. Proceedings of the 27th ACM/IEEE Design Automation Conference pp 52–57
15. Moret B (1982) Decision trees and diagrams. Computing Surveys 14:413–416
16. Nikolskaia M, Rauzy A, Sherman D (1998) Almana: A bdd minimization tool integrating heuristic and rewriting methods. Proceedings of Formal Methods in Computer Aided Design Conference, FMCAD'98 1522:100–114
17. Rauzy A (1993) New algorithms for fault trees analysis. Reliability Engineering & System Safety 5:203–211

18. Xie M, Tan K, Goh Kea (2000) Optimum prioritisation and resource allocation based on fault tree analysis. International Journal of Quality & Reliability Management 17:189–99

Author Index

J. Xu et al. (eds.), *International Conference on Management Science and Engineering Management 2015*,
Advances in Intelligent Systems and Computing XXX, DOI: XXXXXXXXXX