Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 180 (2021) 344–351

Procedia Computer Science

www.elsevier.com/locate/procedia

International Conference on Industry 4.0 and Smart Manufacturing

pyBNBowTie: Python library for Bow-Tie Analysis based on Bayesian Networks

Frank T. Zurheide*, Eckehard Hermann, Harald Lampesberger

Dept. Secure Information Systems, University of Applied Sciences Upper Austria, Softwarepark 11, 4232 Hagenberg, Austria

Abstract

In addition to more conventional but often less precise methods, the risk assessment as part of the risk management process can be performed with the bow-tie analysis method. A bow-tie analysis describes the effects of causes on a top event and the resulting consequences. Bayesian networks, on the other hand, offer a mathematically concise way of describing dependencies between events under uncertainty. The mapping of bow-tie analysis into Bayesian networks is intended to make their superior calculation options available.

While the mapping algorithm of a bow-tie method into a Bayesian network is described in the literature, no computer program carrying out this mapping has been found so far. In this text, a Python library, that is validated using published examples, is presented and made publicly available for mapping bow-tie methods into Bayesian networks.

Keywords: bow-tie analysis; Bayesian network; risk analysis; risk assessment

1. Introduction

The fourth industrial revolution — Industry 4.0 — is characterized by the appearance of cyber-physical systems, Internet of Things (IoT), smart factories, and generalized decentralized production processes [7]. These systems yield a higher level of automatization in industrial production. The higher level of communication and autonomous systems require a risk assessment process that can keep up with dynamic and quick production processes [12]. One possible solution is the automation of the risk assessment process.

The bow-tie method [22] can be the foundation of such an automated process, which can also be easily integrated into a business process. It connects the causes and consequences of risks. As a tool for risk assessment, the bow-tie method is well established in the context of complex systems like the chemical industry and the nuclear industry

* Corresponding author.
E-mail address: frank.zurheide@gmail.com

10.1016/j.procs.2021.01.172

[22, 21, 23]. It can be used to describe safety and security risks. The descriptions are qualitative and quantitative. The challenge is the computation of the probability of the consequences dependent on the threats. This computation can be performed with Bayesian networks (BN).

The mapping from a bow-tie analysis to a Bayesian network is described by Khakzad et al. [11]. This mapping is used for risk assessments, e.g. Roboter-human co-working spaces [19] and leakage failure of submarine oil and gas pipelines [14].

Bayesian networks are well established in the risk community. They provide a mathematically proven way to compute probabilities for dealing with uncertainty. "*They can offer different functionality such as predictive and diagnostic analysis, model updating, and optimization*" [10]. The authors state some downsides of BNs: "*The major limitations of BNs application as standalone approaches are that there are limited formal semantic guidelines available for developing BNs for a system and they do not guarantee a coherent model*". They describe that coherent models can be created if the BNs are created from other models by model-to-model transformations. Then the correctness of the BN depends on the correctness of the source model, e.g. a bow-tie analysis.

Although the bow-tie analysis to Bayesian network mapping is described and used by several authors, no publicly available implementation could be found. Therefore the mapping algorithm is implemented, described here and the library is made available on GitHub[1] under the Apache 2.0 license [24].

In the next section we introduce risk assessment and tools like the bow-tie analysis and Bayesian networks. The algorithm to map bow-tie analysis to Bayesian networks is described in section 3. In section 4 the implementation of this mapping algorithm is described. The feasibility of the implementation is shown with a test. Section 5 concludes this text.

2. Risk assessment

The assessment of risks can be performed in different kinds. The standard for risk management ISO 31000 [9] and it's substandard ISO 31010 [8] include over 30 tools for risk assessment. In the following, a few are briefly described.

2.1. Fault-tree analysis

The fault-tree method is a tree that uses Boolean logic to describe possible failure paths. The fault tree is constructed starting from a top event. This top event is typically a critical failure or loss of control. All possibilities that cause this top event are identified and modeled in a tree structure. These possibilities can be refined and therefore result in more complex trees [8]. The end of the failure paths, the nodes of the fault tree, are basic events that are also called hazards or threats. The probability of occurrence of each basic event can be annotated, and with this data, the probability of a failure of the top event in the fault-tree analysis can be calculated. The fault-tree method was developed in the sixties of the last century [16].

2.2. Event-tree analysis

Starting from an initiating event an event tree is constructed. This initiating event can have a finite number of outcomes, very often they are binary events (yes/no). For each of those outcomes, the event tree can be refined by introducing additional events. The leaf nodes of the event tree represent the consequences with regard to the events along the path. Event-tree analysis can be used qualitatively or quantitatively by annotating the probability of the outcomes of an event. If the probabilities of all events are defined, the probabilities of all consequences can be determined. Following this computation, so-called barriers can be placed in the paths to mitigate negative effects. [8, 16]

[1] https://github.com/zurheide/pybnbowtie

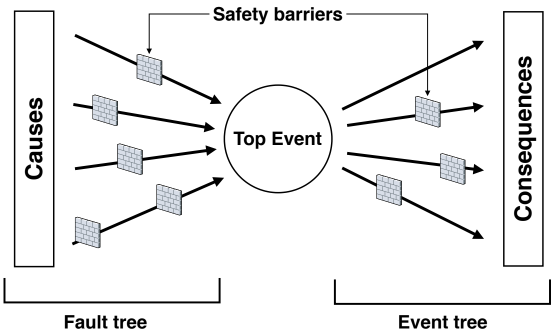

Fig. 1: Example of a generic bow-tie diagram from [16].

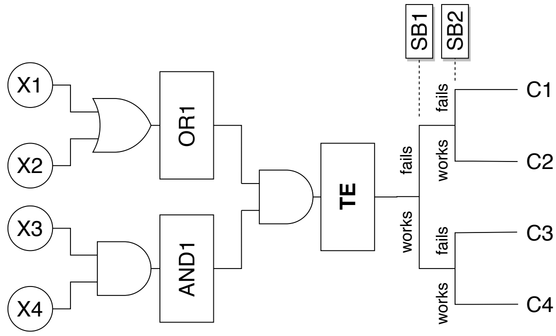

Fig. 2: Exemplary realization of a bow-tie analysis.

2.3. Bow-tie analysis

The bow-tie analysis combines the fault-tree analysis and the event-tree analysis. Both trees are connected to a top event. For the event tree, a top event is the initiating event. From a top event, the fault tree extends to the left and the event tree to the right. The bow-tie analysis therefore connects causes (or threats) and consequences (or outcomes) on the left and right sides, respectively [16].

The top event is the event that should not happen, e.g. attack, loss of containment in the chemical industry, or loss of separation in aviation [16, 18].

A generic bow-tie diagram is shown in Fig. 1. The name of the tool originates from the shape of the diagram that resembles the necktie bow tie. An exemplary realization of a bow-tie analysis is given in Fig. 2. The causes X_i, the basis events, are on the left side. Logical gates `AND1` and `OR1` connect the basis events towards the top event TE. To the right side the events — safety barriers SB_i — are branching the paths toward the consequences C_i. A basic event can also be the top event in a new bow-tie diagram, analyzing the basic event.

Bow-tie diagrams can be layered so that a top event or consequence on a layer is the basis event on an above layer. Probabilities of safety barriers can be dependent on fault trees. This enables a fine regulation of the safety barrier operation dependent on the actual situation. Note, that this description for safety barriers is available by the input data format, however, it is not jet tested.

To mitigate risks, safety barriers are introduced. Sklet [17] defines safety barriers as "*physical and/or non-physical means planned to prevent, control, or mitigate undesired events or accidents*". Safety barriers in the bow-tie analysis are placed at the fault-tree analysis for prevention and control measures, while the ones in the event-tree analysis are control and mitigation measures.

2.4. Bayesian network

A Bayesian network (BN) is a probabilistic graphical model. A BN consists of nodes that represent variables and conditional dependencies of the variables. These nodes are connected as a directed acyclic graph. The conditional dependencies are assigned in terms of conditional probability tables (CPT).

Let the BN consist of n variables $A_1, A_2, \ldots, A_n$. The joint probability distribution of the BN can be simplified as

$$P(A_1, A_2, \ldots, A_n) = \prod_{I=1}^{n} P(A_i | Parents(A_i)) \tag{1}$$

where $Parents(A_i)$ denote the set of parent nodes of the node A_i [6]. If A and B are two random events with a prior probability $P(A)$ and $P(B)$, the posterior probability of event A occurring, given that B has occurred, can be determined by the Bayes' rule [13]

$$P(A|B) = \frac{P(A)P(B|A)}{P(B)} \quad . \tag{2}$$

3. Mapping algorithm

This section covers the mapping of the risk assessment tools to BN. Before the mapping of a bow-tie analysis to a BN is shown, the mapping of a fault-tree analysis and an event-tree analysis is explained. The two later mapping algorithms are the basis for the bow-tie analysis mapping.

3.1. Fault-tree analysis mapping

The mapping of a fault-tree analysis to a BN has been described by Bobbio et al. [3]. To map the fault-tree analysis the authors make the following assumptions. (i) events are binary events, (ii) events are statistically independent, (iii) logical AND and OR gates represent relationships of events and causes, and (iv) the top event is the root of the fault-tree analysis [3].

In the fault-tree analysis, the negative logic is applied. A random component C that fails has a value $C = 1$ and if it works $C = 0$. For the mapping algorithm, the values of the conditional probability table for the logical gates are given by the authors. With these tables, the fault tree can be converted step by step.

3.2. Event-tree analysis mapping

Bearfield and Marsh [2] present an algorithm to map an event-tree analysis to a BN. This mapping introduces additionally to the event nodes a consequence node. This node contains states all possible consequences of the even tree and additionally a state without a consequence, the safe state. This state represents the probability of no failure at the top event $(1 - P(TE))$.

The authors distinguish between consequence arcs and causal arcs. Consequence arcs are directed from the event (safety barrier) to the consequence node. So a safety barrier influences the probability of the consequences. The causal arc connects a safety barrier with all following safety barriers. This connection is necessary if an event changes the possibility of following events.

This mapping algorithm contains steps where both arc types are removed if they are redundant. This is not necessary — the BN delivers the same results with the redundant arcs — but the numerical effort is reduced.

3.3. Bow-tie mapping

Mapping bow-tie diagrams to BN is described by Khakzad et al. [11]. They use the mapping algorithms from Bobbio et al. [3] and from Bearfield and Marsh [2] to map fault-tree analysis and event-tree analysis to BN. The algorithm introduces a connection of the nodes of the mapped fault tree to the nodes of the event tree.

The mapping algorithm is displayed in Fig. 4. The figure is adapted from Khakzad et al. [11] and Villa et al. [21]. In the fault-tree analysis, the causes (basis events) are mapped to root nodes of the BN. Safety barriers and gates are mapped as intermediate safety nodes and are connected with the root nodes. The top event is mapped as a pivot node. At the event-tree analysis, the safety barriers are mapped to safety nodes. The CPTs are set and, when necessary, a connection from the pivot node is created. A consequence node with states of all consequences and an additional safe state is created. This node receives connections from the pivot node and the safety nodes.

4. Implementation

The algorithm is implemented in the programming language Python version 3.7 [20]. The library used for the Bayesian networks computations is `pgmpy` [1]. The mapping library `pyBNBowTie` is available on GitHub [24] and is released under the Apache 2.0 license. The program consists of nearly 2200 lines of Python code.

For data exchange, the Open PSA MEF XML format is used. This data structure is intended to create exchangeable descriptions of probabilistic safety assessments (PSA) among different software solutions [5].

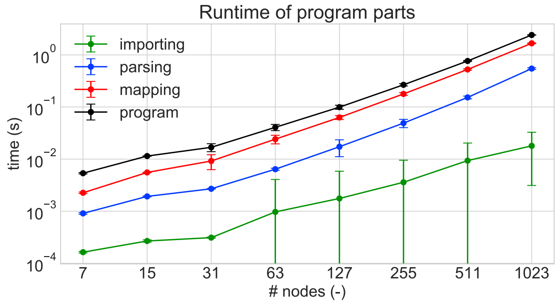

Fig. 3: Runtime of the implemented algorithm.

4.1. Runtime

The following results were produced running the code on one Amazon Web Services (AWS) c5a.large instance. The time was measured with the `time.perf_counter()` Python function. Each program was executed 100 times. A generic BT consisting of a fault-tree analysis with 31 nodes (basic events and gates) and an event-tree analysis with 31 nodes was mapped to a BN. Afterwards, the probabilities of the top event as well as the consequences were computed. Importing the bow-tie analysis, parsing the data, and mapping in a BN took 0.001 s, 0.004 s, and 0.014 s, respectively. Exact inference of the BN for the top event required 0.153 s and for the consequences 1.120 s. The runtime of the program is primarily determined by inference operations in the BN.

Doubling the number of fault-tree analysis nodes to 64 increased the time for importing, parsing, and mapping from 0.019 s to 0.027 s while the inference of the top-event required 27.4 s and of the consequences 64.2 s. The required computation time of the BN grows exponentially with increasing the number of nodes. Changing the algorithm for inference from an exact solution to an approximate solution significantly speeds up the program.

The time for importing, parsing, mapping, and the overall execution time is given in Fig. 3. This analysis was done without inference in the BN. The numbers on the x-axis are the number of nodes in the fault-tree analysis and the event-tree analysis. Again, the exponentially growing effort for creating the CPTs determines the computation time of the algorithm.

4.2. Example bow-tie analysis

An example of the implementation of a bow-tie diagram with the basis events, logical gates, top event, safety barriers, and consequences are given in Fig. 2. The implementation of the mapping algorithm of Khakzad et al. [11] takes the description of the bow-tie analysis in the MEF XML format as input and maps it to a BN of the `pgmpy` library.

Listing 1 shows an example program to map a bow-tie diagram to a BN. The MEF input file `opsa_input.xml` is loaded and parsed with the `OPSA_Importer` class to a tree structure. The `MappingBowTie` class creates a BN (`model`) of type `pgmpy`. With this `model`, probabilities of the consequence node and in the network can be computed and inference in the network can be performed (not shown in the program).

Listing 1: Minimalistic Python program for importing and mapping a bow-tie diagram to a BN

```python
import xml.etree.ElementTree as ET
from treelib import Tree
from bowtie.io.import_opsa import OPSA_Importer
from bowtie.mapping.mapping_bowtie
  import MappingBowTie

# read XML MEF file, here opsa_input.xml
xml_root = ET.parse("opsa_input.xml").getroot()
# create and prepare tree for results
tree = Tree()
tree_root = tree.create_node('root')
```

```
# create importer and parse MEF
importer = OPSA_Importer()
importer.parse(xml_root, tree, tree_root)

# bow-tie diagram is now in tree
# map the bow-tie diagram to a Bayesian network
mapper = MappingBowTie(tree)
model = mapper.map()

# check the Bayesian network
print('check Bayesian network = {}'.format(model.check_model()))

# show nodes of Bayesian network
print('nodes')
print(model.nodes())
```

The resulting BN is shown in Fig. 5. From the basis events X_i to the top event TE, the structure of the fault-tree analysis is well preserved. The consequence node is dependent on the top event. The safety barriers SB_i in this example are independent from the top event and therefore no connection is created. Consequence arcs are pointing from the two safety barriers towards the consequences, and one causal arc points from SB_1 to SB_2. In this case, none of the arcs can be eliminated. The nodes of the BN that represent the event tree part of the bow-tie diagram are colored gray. Contrary to the fault-tree part of the bow-tie diagram, the event-tree part does not maintain its structure.

4.3. Evaluation: Mixing Tank Accident

A report by the U.S. Chemical Safety Board (CSB) investigates an accident that involved "*the ignition of a vapor cloud generated by mixing and heating a flammable liquid in an open-top tank without adequate safety controls*" [4]. Khakzad et al. [11] use this accident as an exemplary case to apply their algorithm to map a bow-tie diagram to a BN. First, they develop a bow-tie diagram from the given data in the report from CSB. Then they map this diagram to a BN. Finally, the probability of the top event and the consequences are computed from this BN. The complete bow-tie diagram is presented in the second reference.

To reproduce this case, the bow-tie diagram is manually transferred to the MEF format. Then, the bow-tie diagram is mapped — similar to the program in Listing 1 — and the probabilities are calculated from the resulting BN. The BN is displayed in Fig. 6. The causal arc from the node *Ignition* to *Sprinkler* is eliminated from the network.

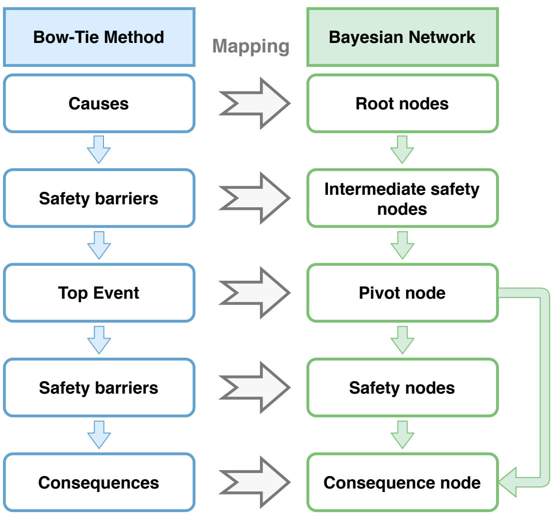

Fig. 4: Algorithmus to map a bow-tie diagram to a BN.

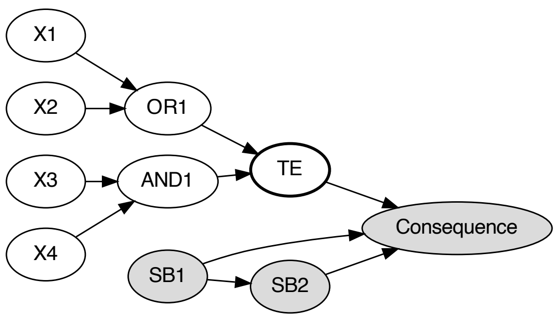

Fig. 5: BN created with the implemented mapping algorithm from the bow-tie diagram in Fig. 2.

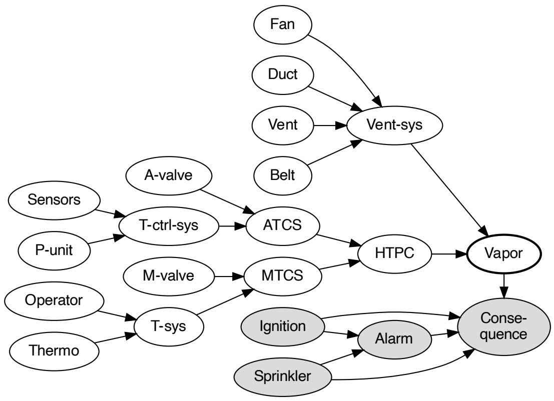

Fig. 6: BN created with implemented mapping algorithm for the mixing tank accident.

Table 1: Probabilities for the mixing tank accident. Values are given for the top event and the consequences.

Symbol	**Probability**		
	Khakzad et al.	ETA	Bow-tie diagram
Vapor	1.68×10^{-3}	1.68×10^{-3}	1.68×10^{-3}
C_1	0.00	1.13×10^{-3}	1.13×10^{-3}
C_2	0.00	3.27×10^{-4}	3.27×10^{-4}
C_3	1.04×10^{-3}	4.69×10^{-5}	4.69×10^{-5}
C_4	3.03×10^{-4}	1.36×10^{-5}	1.36×10^{-5}
C_5	3.23×10^{-4}	1.61×10^{-4}	1.61×10^{-4}
C_6	4.21×10^{-7}	2.10×10^{-7}	2.10×10^{-7}
C_7	1.35×10^{-5}	6.71×10^{-6}	6.71×10^{-6}
C_8	1.75×10^{-8}	8.74×10^{-9}	8.74×10^{-9}

Results from Khakzad et al. and the implemented algorithm are shown in Tab. 1. Values given by Khakzad et al. and computed values in this work differ. The largest differences are at the consequences of C_1 and C_2. With the developed program, C_1 is the largest probability, magnitude(s) larger than the other consequences. Note that the probability of the top event $Vapor$ is the same for both proceedings.

To verify the results, the event tree is modeled with the software Logan [15]. The results are given in Tab. 1 in the column ETA. The results from Logan and our software are the same. The cause of the difference is not further followed up because the data from Khakzad et al. are not publicly available. CPTs of the nodes in a BN grow quickly with additional entries. Assumptions are, that filling these tables manually caused errors — something that can be avoided by an automatic procedure, like the one presented in this work.

5. Conclusion

The Bow-tie analysis is a widely used method to describe consequences caused by threats. This tool allows a qualitative and quantitative description of risks.

The algorithm by Khakzad et al. [11] for mapping a bow-tie analysis into a Bayesian network enables superior calculation options in risk assessments based on bow-tie analysis. In this work, this mapping algorithm has been implemented in the programming language Python. To keep the algorithm broadly usable, Open PSA MEF is selected as input format.

An accident of a mixing tank has been used to show the functionality and correctness of the implementation. The created Bayesian network is shown, and the computed values of the consequence probabilities are presented and compared to the ones given by Khakzad et al. [11].

The software presented in this paper is made publicly available on GitHub [24].

References

[1] Ankan, A., Panda, A., 2015. pgmpy: Probabilistic graphical models using python, in: Proceedings of the 14th Python in Science Conference (SCIPY 2015), Citeseer. pp. 6–11.

[2] Bearfield, G., Marsh, W., 2005. Generalising event trees using bayesian networks with a case study of train derailment, in: International Conference on Computer Safety, Reliability, and Security, Springer. pp. 52–66.

[3] Bobbio, A., Portinale, L., Minichino, M., Ciancamerla, E., 2001. Improving the analysis of dependable systems by mapping fault trees into bayesian networks. Reliability Engineering & System Safety 71, 249–260.

[4] CSB, 2007. Mixing and Heating a Flammable Liquid in an Open Top Tank. Investigation No. 2006-08-I-IL. U.S. Chemical Safety Board. Washington, DC.

[5] Epstein, S., Rauzy, A., 2008. Open-PSA Model Exchange Format.

[6] Fenton, N., Neil, M., 2019. Risk assessment and decision analysis with Bayesian networks. 2 ed., CRC Press.
[7] Hermann, M., Pentek, T., Otto, B., 2016. Design principles for industrie 4.0 scenarios, in: 2016 49th Hawaii international conference on system sciences (HICSS), IEEE. pp. 3928–3937.
[8] IEC 31010, 2019. IEC 31010:2019: Risk management — Risk assessment techniques. Standard. International Organization for Standardization. Geneva, CH.
[9] ISO 31000, 2018. ISO 31000:2018: Risk management – Guidelines. Standard. International Organization for Standardization. Geneva, CH.
[10] Kabir, S., Papadopoulos, Y., 2019. Applications of bayesian networks and petri nets in safety, reliability, and risk assessments: A review. Safety science 115, 154–175.
[11] Khakzad, N., Khan, F., Amyotte, P., 2013. Dynamic safety analysis of process systems by mapping bow-tie into Bayesian network. Process Safety and Environmental Protection 91, 46–53. doi:10.1016/j.psep.2012.01.005.
[12] Kiss, M., Breda, G., Muha, L., 2019. Information security aspects of industry 4.0. Procedia manufacturing 32, 848–855.
[13] Koller, D., Friedman, N., 2009. Probabilistic graphical models: principles and techniques. MIT press.
[14] Li, X., Chen, G., Zhu, H., 2016. Quantitative risk analysis on leakage failure of submarine oil and gas pipelines using bayesian network. Process Safety and Environmental Protection 103, 163–173.
[15] LOGAN, 2018. LOGAN Fault and Event Tree Analysis. URL: loganfta.com/download/LoganManual.pdf.
[16] de Ruijter, A., Guldenmund, F., 2016. The bowtie method: A review. Safety science 88, 211–218.
[17] Sklet, S., 2006. Safety barriers: Definition, classification, and performance. Journal of loss prevention in the process industries 19, 494–506.
[18] Talbot, J., Jakeman, M., 2009. Security risk management. body of knowledge.
[19] Teimourikia, M., 2017. Co-engineering safety and security in risk-prone smart work environments. Ph.D. thesis. Politecnico di Milano Department of Electronics, Information and Bioengineering.
[20] Van Rossum, G., Drake Jr, F.L., 1995. Python tutorial. Centrum voor Wiskunde en Informatica Amsterdam, The Netherlands.
[21] Villa, V., Paltrinieri, N., Khan, F., Cozzani, V., 2016. Towards dynamic risk analysis: A review of the risk assessment approach and its limitations in the chemical process industry. Safety science 89, 77–93.
[22] Weber, P., Medina-Oliva, G., Simon, C., Iung, B., 2012. Overview on bayesian networks applications for dependability, risk analysis and maintenance areas. Engineering Applications of Artificial Intelligence 25, 671–682.
[23] Zio, E., 2018. The future of risk assessment. Reliability Engineering & System Safety 177, 176–190.
[24] Zurheide, F.T., 2020. pybnbowtie. https://github.com/zurheide/pybnbowtie.