Confidence analysis for nuclear arms control: SMT abstractions of Bayesian Belief Networks

Paul Beaumont[1], Neil Evans[2], Michael Huth[1], and Tom Plant[2]

[1] Department of Computing, Imperial College London, London, SW7 2AZ, UK
{m.huth, paul.beaumont09}@imperial.ac.uk
[2] AWE Aldermaston, Reading, Berkshire, RG7 4PR , UK
{Neil.Evans, Tom.Plant}@awe.co.uk

Abstract. How to reduce, in principle, arms in a verifiable manner that is trusted by two or more parties is a hard but important problem. Nations and organisations that wish to engage in such arms control verification activities need to be able to design procedures and control mechanisms that capture their trust assumptions and let them compute pertinent degrees of belief. Crucially, they also will need methods for reliably assessing their *confidence* in such computed degrees of belief in situations with little or no contextual data. We model an arms control verification scenario with what we call *constrained* Bayesian Belief Networks (cBBN). A cBBN represents a set of Bayesian Belief Networks by symbolically expressing uncertainty about probabilities and scenario-specific constraints that are not represented by a BBN. We show that this abstraction of BBNs can mitigate well against the lack of prior data. Specifically, we describe how cBBNs have faithful representations within a Satisfiability Modulo Theory (SMT) solver, and that these representations open up new ways of automatically assessing the confidence that we may have in the degrees of belief represented by cBBNs. Furthermore, we show how to perform symbolic sensitivity analyses of cBBNs, and how to compute global optima of under-specified probabilities of particular interest to decision making. SMT solving also enables us to assess the relative confidence we have in two cBBNs of the same scenario, where these models may share some information but express some aspects of the scenario at different levels of abstraction.

1 Introduction

AWE's Arms Control Verification Research programme supports and advises UK Government, through the UK Ministry of Defence (MOD), on verification measures that might be put into operation in the context of future arms control agreements. Specifically, the UK may one day be involved in a bilateral or multilateral agreement regarding the monitoring or reduction of arms. Any such agreement would very likely contain provisions for verifying that parties to this agreement are indeed compliant with their obligations expressed within said agreement. These provisions may be in the form of inspections, deployment

of monitoring equipment, use of satellite imagery, agreement of formal notice periods for certain activities and so forth.

An understanding of the reliability of such provisions and their interaction will be paramount: an agreement is more likely to be signed, and honoured, if all parties can be confident that the agreement's provisions allow them to verify compliance of other parties with the agreement. These provisions will be informed by strategic and conflicting interests of the parties. We propose to further such understanding by using mathematical analysis in this problem space, based on mathematical representations of arms control verification scenarios. In such scenarios, we are primarily interested in three types of quantities:

- **Trust**: a bias in the processing of imperfect information about another party
- **Degree of Belief**: the amount we believe a proposition is true
- **Confidence**: a measure of the uncertainty we should have in our degree of belief in a proposition.

Assuming that verification measures have been deployed, a mathematical representation of an arms control verification scenario should then allow a party to have high confidence in its degrees of belief (even if these degrees of belief are low), regardless of what trust it places in other parties. Such trust may, e.g., be based on past dealings between the parties or may be affected by the conduct exhibited within the verification activities themselves.

Mathematical representations should therefore give us measures of **Trust**, **Degree of Belief**, and **Confidence** so that we can investigate the trade-offs between these measures, assess their relative merits, or perform optimisation – e.g. to determine extremal cases of interest. Other desired capabilities of such mathematical representations are:

(1) ability of non-technical users (e.g. diplomats) to understand these representations and their results
(2) ability to represent both subjective (e.g. expert opinion) and objective data
(3) ability to determine which representational aspects or results are due to different subjective modelling decisions
(4) ability to represent and analyse dynamic, time-dependent scenarios
(5) ability to perform optimisation for measures of interest and their trade-offs
(6) ability to certify or formally prove that analysis outputs are correct.

In this paper, we explore the suitability of one such mathematical representation, Bayesian Belief Networks (BBNs) – see e.g. [10] – against the aforementioned desired capabilities in understanding the measures of **Confidence**, **Degrees of Belief**, and **Trust**.

We assume that little or no prior data is available for modelling arms control verification scenarios. This prevents us from using methods for estimating probabilities within BBNs. Moreover, control mechanisms may be subject to non-probabilistic, logical rules so we want to enrich BBNs with logical constraints. Thus we propose to use symbolic representations of both the uncertainty of probabilities within a BBN and of logical constraints of such a symbolic BBN. These

constrained BBNs (cBBN) generalise BBNs in that the latter have no such uncertainty and no logical constraints. cBBNs also generalize Credal networks (see e.g. [9]), where the latter abstract probabilities of BBNs with convex intervals – a particular form of uncertainty – but cannot capture logical constraints.

To get the ability to assess confidence in degrees of belief of cBBNs, we develop techniques for determining whether one or more cBBNs are satisfiable at the same time, where satisfiability witnesses are BBNs that meet all constraints expressed in the cBBNs. We also show how to compute optimal such witnesses for measures of interest such as the probability of a cBBN node that informs decision making or such as the worst-case sensitivity of a node in a cBBN. Technically, we achieve this by specifying cBBNs in the Satisfiability Modulo Theories (SMT) solver Z3 [20], and by formulating confidence queries directly in SMT.

The contributions of this paper are therefore in proposing a new approach and methods for representing and analysing Bayesian Belief Networks, with a concrete application in national security in mind. We also demonstrate that our new methods genuinely enrich the modelling capabilities that exist to date in this application domain, notably (2), (3), and (5) above. We also began a case study [3] demonstrating that our approach can accommodate the temporal capability in (4) but we cannot report on this within the scope of this paper.

We note that our approach, an SMT-based analysis of constrained BBNs, can not only express logical rules of arms control verification, it can also use such logical rules to ensure a consistent relationship between different levels of abstraction in the comparison of two or more cBBNs that model the same arms verification control scenario. Our SMT-based approach is also consistent with realising the capabilities listed in (1) and (6) above, and scoping out the potential for this is subject to future work. The reader is hoped to appreciate that this paper emphasises the exposition of our new methods and their utility for this case study, at the expense of providing less detail on the more routine tool building activities that support and animate these methods.

We emphasise that our methods are of general interest to those who model any security aspects with BBNs but have little contextual data that informs their models or to those that may need to constrain these BBNs logically.

Outline of paper: In Section 2 we present our arms control verification scenario and model it with a simple Bayesian Belief Network. The scenario is designed to be comprehensible to a non-expert of the application domain. Section 3 contains a gentle introduction to Satisfiability Modulo Theory solving and explains how constrained Bayesian Belief Networks can be represented as input for an SMT solver. Our methods for assessing the confidence in constrained Bayesian Belief Networks are developed in Section 4. The context of our work and related work are discussed in Section 5 and the paper concludes and discusses future work in Section 6.

2 An arms control verification scenario

Consider two fictitious nation states (referred to as "nations" below to avoid confusion with "states" of a BBN), N1 and N2, the latter of which is tasked with identifying whether boxes A and B belonging to the former and installed in a controlled inspection facility contain nuclear weapons. Nation N2 is also given declarations from nation N1 as to what is supposed to be in these boxes. The purpose of this inspection within an arms control agreement may be that the contents of the boxes are on their way for decommissioning, destroying, storage, civilian reuse, etc. Our mathematical model of the scenario does not reflect what may happen to the material post-inspection, but more detailed models may well reflect this. It should be noted that the models of this case study are created by nation N2 in order to assess this scenario.

Nation N1 declares that one box does indeed include nuclear weapons, and that the other does not. To illustrate that we can also add some gamification (unrealistic in a real scenario), let us assume that nation N1 won't reveal in which box the nuclear weapon might be, and that the inspecting party is allowed to inspect only one of these two boxes. The inspecting party, say nation N2 or some third party, is given a radiation detector with a specific, known sensitivity and known false-error reporting rate. The detector shows a green or red light based on whether nuclear materials in a particular ratio of a particular isotope are present or not. No other information bar this colour outcome is provided, which establishes an information barrier that can hide, e.g., important weapon design secrets of nation N1 – a requirement for agreeing to such inspections [18].

The design of the detector has been agreed upon by both nations N1 and N2. Nation N2 believes that it may be possible for nation N1 to spoof a radioactive signal (or indeed block a radioactive signal through under-saturation or over-saturation of gamma signals), to fool the detector, or indeed, to have just placed radioactive material in the boxes, but no weapon (in which case, nation N2 even needs to consider its degree of belief of nation N1 being able to build a nuclear weapon). There is also a possibility that the nuclear material may have been enriched to a physical state that is outside the detectable ratio range of the detector, but nation N2 thinks this is unlikely.

There are of course multiple ways of modelling this scenario. One advantage of our approach developed below is that it is able to compare different such models analytically and automatically. In this case study, we consider a simple and a more detailed model – both of which have some nodes in common (demonstrating that we can accommodate such overlap). The simple BBN model is depicted in Figure 1. In a BBN, nodes represent (probabilistic) events. Such events may be conditional on other events (their parent nodes in a dependency graph). In our simple model, we have the following set of nodes:

- Box: which box will be inspected, that choice will determine whether nation N2 expects nuclear weapons to be present or not
- Spoof: determining nation N2's belief in a spoofed signal

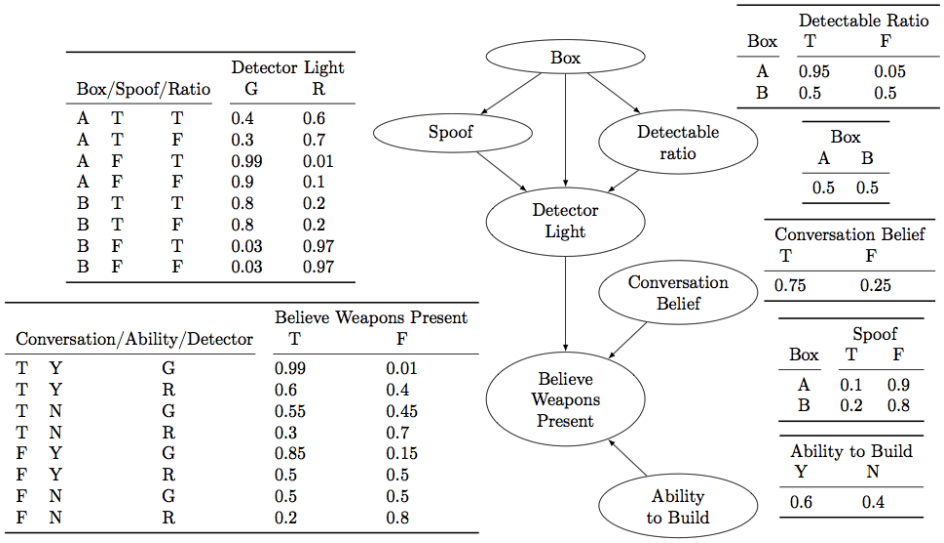

Fig. 1. Simple Bayesian Belief Network modelling our nuclear arms verification scenario

- Detectable Ratio: probability that the fissile material of the object in the box has an isotopic ratio that the system is designed to give a green light for
- Detector Light: accounts for false positive and false negative rates of the detector itself, and determines the green/red light state on the detector
- Conversation Belief: models whether or not external discussions with nation N1 would lead nation N2 to believe declarations of nation N1
- Ability to Build: captures nation N2's uncertainty over the technical abilities of nation N1, irrespective of the detector result
- Believe Weapons Present: the overall belief of nation N2 in a nuclear weapon being present in the inspected box.

The BBN $\mathcal{B}$ in Figure 1 shows the dependency graph of the simple model. For example, the belief in the presence of nuclear weapons depends (through incoming edges) on the events Detector Light, Ability to Build, and Conversation Belief. Each node in the BBN $\mathcal{B}$ has a probability table from which one can compute its probability. For node Box we see that the inspection of a box is determined by flipping a fair coin. For node Believe Weapons Present, this probability table lists the probability distributions conditionally on the three aforementioned parent events. The probabilities used in this scenario are fictitious but convey plausible perceived levels of trust and degrees of belief.

Figure 2 shows how to compute probabilities for all nodes via the Law of Total Probability, by "summing out" conditional probabilities so that probabilities at a node are expressed in terms of probabilities of its parent nodes only.

$$\sum_i \sum_j \sum_k \mathbb{P}(\mathrm{BWP} = T \mid \mathrm{AB} = i, \mathrm{CB} = j, \mathrm{DL} = k) \cdot \mathbb{P}(\mathrm{AB} = i) \cdot \mathbb{P}(\mathrm{CB} = j) \cdot \mathbb{P}(\mathrm{DL} = k)$$
$$\sum_i \sum_j \sum_k \mathbb{P}(\mathrm{BWP} = T \cap \mathrm{AB} = i \cap \mathrm{CB} = j \cap \mathrm{DL} = k)$$

Fig. 2. Two ways of computing node marginal $\mathbb{P}(\mathrm{BWP} = T)$ for Believe Weapons Present (BWP): via summing (first line) over all possible combinations of the conditional probability, multiplied by the parent marginals Ability to Build (AB), Conversation Belief (CB), Detector Light (DL); or via the joint probability distributions (second line)

3 Expressing constrained BBNs in an SMT solver

Satisfiability modulo theories [20, 1] is an approach to automated deduction supported with robust and powerful tools that combine the state-of-the-art of deductive theorem proving with that of SAT solving for propositional logic. We choose Z3 as SMT solver within our tool, although it would be relatively easy to replace it with another solver such as CVC3 [2].

The SMT solver Z3 has a declarative input language for defining constants, functions, and assertions about them [20]. Figure 3 shows Z3 input code to illustrate that language and its key analysis directives. On the left, constants of Z3 type Bool and Real are declared. Then an assertion defines that the Boolean constant `q` means that x is greater than $y+1$, and the next assertion insists that `q` be true. The directives `check-sat` and `get-model` instruct Z3 to find a witness of the satisfiability of the conjunction of all visible assertions, and to report such a witness (called a model, but we will refer to Z3 models as "witnesses" to avoid any ambiguous use of the word "model" in the paper). On the right of Figure 3, we see what Z3 reports for the input on the left: `sat` states that there is a witness; other possible replies are `unsat` (there cannot be a witness), and `unknown` (Z3 does not know whether or not a witness exists).

```
(declare-const q Bool)
(declare-const x Real)
(declare-const y Real)
(assert (= q (> x (+ y 1))))
(assert q)
(check-sat)
(get-model)
```

```
sat
  (model
    (define-fun q () Bool true )
    (define-fun y () Real (-2.0) )
    (define-fun x () Real 0.0)
  )
```

Fig. 3. Left: sample Z3 input code with a directive to find and to generate a witness. Right: raw Z3 output for the left input code (edited to save space), saying that the conjunction of all assertions is satisfiable, and supporting this claim with a witness.

We encode a BBN in SMT using an automated code generator we have written; it converts a specification of a BBN given in a form similar to that seen

in Figure 1 automatically into SMT code. All state variables of a BBN node are declared in SMT by an appropriate enumeration type. In our simple BBN $\mathcal{B}$, these are mostly Boolean variables or tuples of such Boolean variables. But in general, such variables may take on other values such as integers.

The probability of a node is expressed in SMT as an arithmetic constraint that captures the definition of that probability as a function of the probability of its parent nodes and its own probability table. Although this is merely restating the familiar definitions for BBNs (see e.g. [10] and Figure 2), we carefully circumscribe any use of divisions (occurring through the use of Bayes' Theorem) as equivalent equations of multiplicative terms. This syntactic change avoids the use of division, whose presence complicates automated reasoning and often makes an SMT solver report analysis result `unknown`.

We add constraints that ensure that all probabilities for all nodes add up to 1. Doing this will likely detect any accidental transcription errors in the specifications of probability tables and, more importantly, will ensure that the semantics of a cBBN (where some or all probabilities are under-specified) is still that of a set of concrete BBN that "refine" it by resolving such under-specifications to concrete probability distributions – in the spirit of abstract interpretation [8].

Having this SMT encoding in place, it is now easy to extend it to a cBBN. For example, suppose that we want to relax the probability for when the detector reports a green light in the BBN of Figure 1 in the state in which box A is inspected, nation N2 believes that the signal is being spoofed, and nation N2 believes that a detectable ratio of radioactive material is being used. In that state, we want to change the probability distribution from 0.4 and 0.6 to α and $1 - \alpha$, respectively, where α is constrained to be in a convex interval, say the interval $[0.3, 0.4]$. The choice of such intervals may be informed by external sources such as expert opinions, and the interval may have further non-convex restrictions via logical constraints of the cBBN.

We can represent this in our SMT encoding by declaring a real variable α, and using it and its complement $1 - \alpha$ in place of 0.4 and 0.6 in the assertions of our SMT encoding that contain references to these probabilities (e.g. definitions of overall probabilities at nodes). Additionally, we add the assertion that α be in $[0.3, 0.4]$ by adding (`assert` (`and` ($<=$ `0.3` α) ($<=$ α `0.4`))) to the SMT code for this model. In this manner, we can generalise the BBN $\mathcal{B}$ in Figure 1 to a cBBN, referred to as $\mathcal{C}$ subsequently, in SMT. We note that we can relax more than one such probability in a similar manner and Z3 seems to cope well with multiple such relaxations.

Let us now turn to discussing how we can ask questions about cBBNs in SMT. The simplest possible question is to ask whether the SMT encoding of a cBBN is satisfiable, and failure of satisfiability would point out crude encoding or modelling errors. But we may use the power of an SMT solver to ask more interesting questions. For example, we may ask whether the probability of a node in a cBBN is always below a certain threshold (a form of *vacuity checking [15]*). Our tool allows us to declare such an analysis and to generate Z3 input code that, when run, will try to answer this whilst reflecting all probabilistic

constraints represented in the network (its BBN aspect) and all arithmetic or logical constraints (the relaxations of concrete probabilities and logical rules that make a BBN into a cBBN). In the next section, we discuss richer questions that would not be solvable with BBN tools, and how we use SMT to answer them.

So far we have only discussed encodings of cBBNs that reflect no means of updating evidence. BBNs can model hard evidence, which changes the probabilities in a BBN upon observation of an event as seen for example in Figure 4. These changes propagate through the BBN and algorithms exist that compute this propagation of belief update (see e.g. [10]); our tool uses the Junction Tree Algorithm [17] to that end.

$$\mathbb{P}(\text{Box} = A \mid \text{SO} = T) = \frac{\mathbb{P}(\text{SO} = T \mid \text{Box} = A) \cdot \mathbb{P}(\text{Box} = A)}{\sum_i \mathbb{P}(\text{SO} = T \mid \text{Box} = i) \cdot \mathbb{P}(\text{Box} = i)}$$

Fig. 4. Updated marginal for node Box (Box) via Bayes' Rule once Spoof On (SO) is observed as true

Our tool can accommodate the processing of hard evidence in cBBNs as follows. Since probabilistic uncertainty is expressed via symbolic parameters, we use a Python script of the Junction Tree algorithm supplied within an open-source BBN package from eBay at `github.com/eBay/bayesian−belief−networks` to compute the updated probabilities symbolically. Then we remove the assertions in the SMT code that express the marginal probabilities and replace them (where applicable) with the symbolic assertions computed by this algorithm to reflect the marginals after their update based on this hard evidence. Note that this process is independent of any logical, non-probabilistic constraints of the cBBN and won't modify the SMT code of such constraints.

This update mechanism is external to the SMT solver and needs to post-process the SMT code of the cBBN before the modified SMT code can be further analysed, but now with the hard evidence properly reflected. Furthermore, it is important to realise that the propagation of hard evidence is different from analysing "soft" evidence. In our model we also pull out events of interest by just adding a constraint to our SMT code saying that a state variable at a node has a particular value – *without* propagating this as one would do for hard evidence.

4 Assessing Confidence in cBBNs

Our SMT encodings of constrained BBNs allow new forms of analysis, one of them being a comparison of different such models. To demonstrate this, we present the dependency graph of a more detailed model $\mathcal{B}'$ in Figure 5.

The BBN $\mathcal{B}'$ shares some nodes with the BBN $\mathcal{B}$, and has the same probability tables for these nodes. But $\mathcal{B}'$ refines some nodes of $\mathcal{B}$ to take a more nuanced

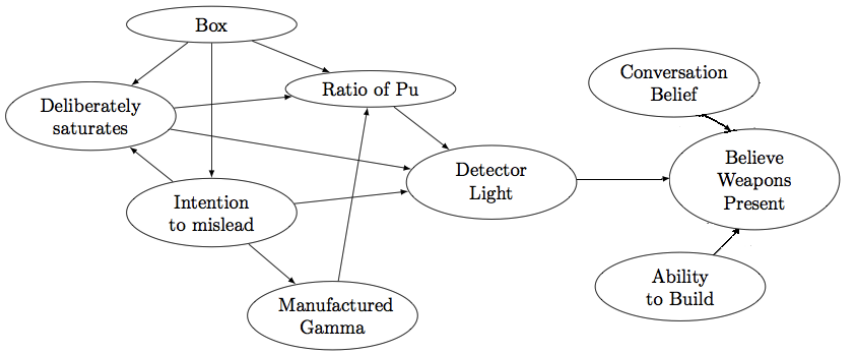

Fig. 5. More detailed BNN $\mathcal{B}'$ of scenario (probabilities of new nodes are in Figure 11 in the appendix)

view of the ability to spoof and the assessment of whether a detectable ratio of nuclear material is present. The new nodes are:

- Intention to Mislead: the belief of nation N2 about nation N1's said intent
- Deliberately Saturates: probability of nation N1 to saturate the information barrier, dependent on nation N1's intention to mislead
- Manufactured Gamma: probability of manufactured gamma being used, also dependent on nation N1's intention to mislead
- Ratio of Plutonium: models said ratio, depends on events Box, Manufactured Gamma, and Deliberately Saturates, and informs event Detector Light.

We write $\mathcal{C}'$ for the constrained BBN that relaxes $\mathcal{B}'$ with the same uncertainty α as $\mathcal{C}$ relaxes $\mathcal{B}$, and pose the following questions:

Q1 For the constrained BBN $\mathcal{C}$, what is the maximal/minimal probability of nation N2 believing that a weapon is present given that nation N2 is uncertain about the prior probability of the detector light turning green?
Q2 How different can the probabilities of nation N2 believing that a weapon is present be between cBBNs $\mathcal{C}$ and $\mathcal{C}'$, i.e. when nation N2 is uncertain about the prior probability of the detector light going green?
Q3 Can the constrained BBNs $\mathcal{C}$ and $\mathcal{C}'$ return different results when we ask whether the probability of nation N2 believing that a weapon is present can be above a threshold, which we are uncertain about?
Q4 For what threshold ranges can such different results for Q3 occur?

Almost all of these questions require us to compute optimal values of a potentially non-linear objective function. We realise this in our tool by implementing unbounded binary search through the Python API for the SMT solver Z3. The pseudo code for this computation is depicted in Figure 12 in the appendix for

the case of global maxima. This method computes optimal values within a desired accuracy $\delta > 0$, and also truncates the mantissa of witness real numbers to a size commensurate to the value of δ. We do this as larger mantissas tend to increase the complexity of reasoning in the SMT solver within the unbounded binary search.

4.1 Optimising a probability over uncertainty

Let us reconsider the cBBN $\mathcal{C}$ obtained from the BBN $\mathcal{B}$ so that the probability distribution for Detector Light is α and $1 - \alpha$ and where α is constrained to be in the interval $[0.3, 0.4]$. We then maximise the variable of the SMT encoding of $\mathcal{C}$ that represents the overall probability of event Believe Weapons Present.

`COP_Believeweaponspresent` represents the largest joint probability contributing to the marginal and `C_4_S1` is the marginal probability for event Believe Weapons Present. The witness returned by the SMT solver for this query is shown in Figure 6.

```
[Believeweaponspresent = 1,
COP_Believeweaponspresent = 72836577/320000000,
C_4_S1 = 100021751/160000000,
x = 2/5,
Abilitytobuild = 1, Conversationbelief = 1, ...]
```

Fig. 6. Excerpt of the witness of the SMT solver (hand-edited to save space) for our SMT-based encoding of cBBN $\mathcal{C}$, where variable `x` denotes the value of α from $[0.3, 0.4]$ for which event Believe Weapons Present has maximal probability `C_4_S1` given that our event of interest is `Believeweaponpresent` = `T`

The real values of a witness are rational numbers since SMT solvers use exact arithmetic – another aspect that helps to establish **Confidence** in computed degrees of belief. The maximal value of `C_4_S1` is 100021751/160000000 which equals 0.6251, and this maximal value is attained when the α (modelled as `x` in the SMT code) has value $2/5 = 0.4$. Witness values relevant to this optimisation query are those of `x` and `C_4_S1`; but the witness reported in Figure 6 also offers some states of the event for which the maximal joint probability (with the node's parents), `COP_Believeweaponspresent`, is attained.

We find that the global minimum of `C_4_S1` is 0.6233, occurring when `x` is 0.3.

4.2 Optimisation for hard evidence

We can also optimise probabilities in cBBNs for hard evidence. The Junction Tree Algorithm (JTA) implemented in the aforementioned open-source code of eBay can also be executed for symbolic input such as for the variable `x` in the SMT representation of $\mathcal{C}$. We then take this symbolic output of the JTA and

post-process it so that divisions are expressed in terms of multiplications (where possible). Figure 7 illustrates what kind of assertions this adds to the SMT model of $\mathcal{C}$, where `x` is the variable that captures the uncertainty in $\mathcal{C}$.

```
s.add((C_4_S1*0.15) == (0.0158175*x+0.09513975))
s.add((C_4_S2*(0.095*x+0.88845)) == (0.02014*x+0.1883514))
```

Fig. 7. Some marginal probabilities revised by hard evidence via the symbolic Junction Tree Algorithm, post-processed to replace divisions with equivalent multiplications

Then we update the relevant portions of the SMT representation with this symbolic input to reflect the hard evidence. Thereafter, we can compute maxima in the same manner as described above.

To illustrate, let us now think of `Believeweaponspresent` = `T` as our parameter of interest and let us consider `Spoof` = `T` as hard evidence. Then we transform $\mathcal{C}$ and its SMT representation as just outlined, and compute the maximum for `C_4_S1` over this transformed SMT code, and find that this is 135289/200000 which equals 0.6764. We can similarly compute the minimum of `C_4_S1` and find that this equals 6659/10000 = 0.6659.

Note that if nation N2 definitely observes the crude node Spoof On, its confidence increases that a weapon is present. We can see this here since the maximal probability increases from 0.6251, when `x` is 0.4, to 0.6764, when `x` is also 0.4, but `Spoof` = `T`. If nation N2 knew that always `Box` = `A` (instead of also allowing for `Box` = `B` in our gamified scenario), then observing spoofing would lead to a drop in confidence as it would hint that there is no weapon. The results above though are in keeping with the probabilities assigned in the node tables for the gamified scenario which crudely models that a spoofed signal can be used to both hide and mimic a weapon.

4.3 Confidence in comparison of cBBNs

Consider two cBBNs that have a common node whose probability will support decision making. We want to compute the maximal difference that these respective probabilities could have, in order to assess with confidence by how much they could differ in principle. We illustrate how this can be done in SMT by considering again the CBBN $\mathcal{C}$ for the simple model, and the BBN $\mathcal{B}'$ for the detailed model (noting that BNNs are also cBBNs). The event of interest in both models is Believe Weapons Present. We want to compute the maximal difference of the joint probability of this event (with its parent nodes) in both models, expressed in our SMT model as

```
(declare-const DIFF Real)
(assert (= DIFF (abs
  (- COP_Believeweaponspresent_mod1 COP_Believeweaponspresent_mod2))))
```

where the suffixes `mod1` and `mod2` separate the name spaces for these two cBBNs within the same SMT model. Since these cBBNs contain also common aspects, we use the logical constraints of the SMT language to specify the "semantic glue" between these common aspects. Doing so prevents the computation of values for `DIFF` that would arise from inconsistent instances of these two cBBNs. Figure 8 illustrates how this is done for the two models considered here. In many cases, we just state that variables have the same meaning.

```
(assert (= Box_mod1 Box_mod2))
(assert (= DetectorLight_mod1 DetectorLight_mod2))
(assert (= Believeweaponspresent_mod1 Believeweaponspresent_mod2))
(assert (ite (= SpoofOn_mod1 2) (= RatioOfPu_mod2 3) (not (= RatioOfPu_mod2
 3))))
```

Fig. 8. Excerpts of SMT code that semantically connects common aspects of $\mathcal{C}$ and $\mathcal{B}'$: e.g. its $if-then-else$ assertion logically relates Spoof On of $\mathcal{C}$ to Ratio of Pu of $\mathcal{B}'$

In other cases, we need to provide glue between different levels of abstraction. For example, that state `SpoofOn` = `F` and only that state of the simple model is mapped to a certain level of ratio of element Pu in the detailed model. Specifically, in the scenario only a ratio of About 10:1 is deemed acceptable (see the last table in Figure 11 of the appendix for how ratio levels are modelled in $\mathcal{B}'$); all other levels would indicate a spoof. The figure shows such an assertion with `ite` (if-then-else) where integers encode states of these variables, e.g., `3` encodes ratio About10 : 1.

Now we can compute the maximum of `DIFF`, which equals 0.0921. The witness for this tells us that the simple model has probability 0.0843 and the detailed one probability 0.176 which realise this difference. In fact, we could in principle extract two BBNs from that witness to study how these probabilities come about.

4.4 Two-dimensional difference analysis

We may also compute such maximal differences for a probability of interest as a function of how uncertainty in two models gets resolved. Let x be the probability for Conversation Belief being true in the simple model, whereas y denotes the corresponding probability in the detailed model. We can now maximise `DIFF` above again, but for each data point (x, y) in $[0, 1] \times [0, 1]$ at some granularity. The result of this analysis is seen in Figure 9.

If both models have the same priors, meaning when $x = y$, we would expect both models to agree most. And we do see a trough of `DIFF` at the $x = y$ axis even though it is somewhat shifted and distorted by the different ways in which these models represent event Detector Light, for example.

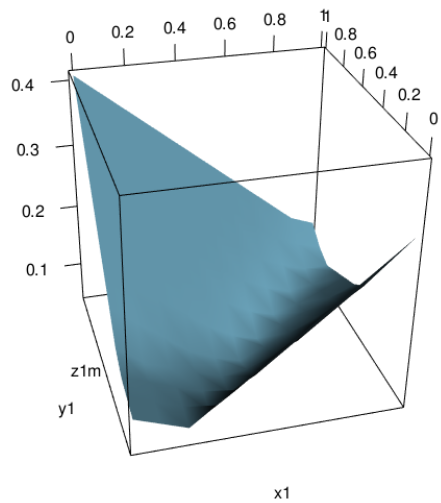

Fig. 9. 3-D plot showing the maximum of variable `DIFF` where x and y axes represent the probability of ConversationBelief = T in the simple, respectively, detailed model

4.5 Computing agreement intervals

We are interested in the probability of event Believe Weapons Present in both cBBNs $\mathcal{C}$ and $\mathcal{B}'$. Let us write pr and pr' for this probability in these models, respectively. Consider a threshold th such that truth of $th < pr$, respectively, $th < pr'$, would support a decision, e.g., for nation N2 to declare that the inspection has been successful. We want to understand for which values th these two cBBNs would agree on that decision. Using their common SMT representation discussed above, we can ask whether

$$((th < pr) \wedge (pr' \leq th)) \vee ((th < pr') \wedge (pr \leq th))$$

is satisfiable in that SMT model. If not, then the two models would support the same decision for threshold value th. Using our global maximum method, where th is now the variable to optimise, we can compute ranges of th for which both models agree in their support of the decision of successful inspections. One such interval of agreement that we can compute for these models is $[0.307, 1.0]$, implying that thresholds at or above 0.307 render the same decisions.

4.6 Sensitivity analyses

We refer to [16] for a discussion of pertinent sensitivity analysis of Bayesian Belief Networks called *Bound*, *Score*, and *Vertex Proximity* (respectively), and their use in an application of digital forensics.

We conducted such sensitivity analyses for our models as well (not shown here). Such methods don't rely on SMT and complement the approach advocated in this paper. But we claim that there is benefit in leveraging our SMT-based approach to compute symbolic sensitivity results. Figure 10 shows such results for sensitivity analysis *Score* [16], for the cBBN $\mathcal{C}$. These symbolic assertions can then be further analysed within an SMT model, e.g., to compute maximal values of these expressions to learn worst-case sensitivities.

Name	**Sensitivity Score**
Box	-148.0*(0.0004275*x - 0.00336683986467894)*(0.0158175*x + 0.633562925005377)
Spoof On	-44.4444444444444*(0.0158175*x - 0.0548602500019255)*(0.0158175*x + 0.633562925005377)
Detectable ratio	-1.90249702734839*(0.0158175*x - 0.238873075001069)*(0.0158175*x + 0.633562925005377)
Detector Light	1.0*(0.01007*x + 0.113679700000125)*(0.0158175*x + 0.633562925005377)/(0.0475*x + 0.536225)**2
Conversation belief	-1.0*(0.01189875*x - 0.255675637493819)*(0.0158175*x + 0.633562925005377)/(1.73472347597681e-18*x + 0.75)**2
Ability to build	-2.77777777777778*(0.01083*x - 0.132740700000797)*(0.0158175*x + 0.633562925005377)
Average	(-0.702999999999999*x + 2.43823333341891)*(0.0158175*x + 0.633562925005377)/6 + (-0.06327*x + 0.498292299972483)*(0.0158175*x + 0.633562925005377)/6 + (-0.0300927467300832*x + 0.454455315103103)*(0.0158175*x + 0.633562925005377)/6 + (-0.0300833333333334*x + 0.368724166668881)*(0.0158175*x + 0.633562925005377)/6 + (-0.01189875*x + 0.255675637493819)*(0.0158175*x + 0.633562925005377)/(6*(1.73472347597681e-18*x + 0.75)**2) + (0.01007*x + 0.113679700000125)*(0.0158175*x + 0.633562925005377)/(6*(0.0475*x + 0.536225)**2)

Fig. 10. Sensitivity of cBBN $\mathcal{C}$ with respect to event Believe Weapons Present, as a function of the sole parameter x that is under-specified in $\mathcal{C}$. The last row computes symbolic averages of all node sensitivities

5 Wider context of work and related work

We first put the research reported in this paper into a wider context of the problem space. Article VI of the Treaty on the Non-Proliferation of Nuclear Weapons (NPT) states that each of the parties to the Treaty

> "... undertakes to pursue negotiations in good faith on effective measures relating to cessation of the nuclear arms race at an early date and to nuclear disarmament, and on a treaty on general and complete disarmament under strict and effective international control."

The UK and Norway have explored, since 2007, how effective verification procedures could be established that could play a vital part in meeting the obligations set out in Article VI [18, 14]. This collaboration made clear that security and safety requirements are essential for the creation of verification technologies and processes, and that more effort is needed at devising such technologies and processes such that all parties can gain and maintain confidence in them. In the

past 15 years, the US and UK engaged in a technical cooperation that explored and evaluated methodologies and technologies for the verification of arms control treaties [24]. This work showed that it is feasible to monitor and verify nuclear warheads, components and processes; but it also identified the need for further research. Our work reported here can be seen as making a contribution to methods that would allow parties to build and sustain confidence in particular arms control verification mechanisms.

We now discuss related work outside this problem space. Robust optimisation is an approach to optimisation in which one seeks a measure of robustness against deterministic uncertainty in parameters of the optimisation problem [4]. Robust optimisation has already been applied in computer security, e.g. to model human adversaries in complex security resource allocation problems [21]. Mixed Integer Linear Programming (MILP) and its non-linear variant MINLP can express constraints stated in propositional logic, but – unlike SMT – seem unable to express relational or functional structure within atomic propositions. Robust optimisation has been applied to MINLP problems of scheduling under bounded uncertainty [13]. This is related to our work in [3] where we use SMT to robustly compute optimal schedules for nuclear arms inspection regimes over measures of interest to participating parties.

Z3opt is an SMT solver based on Z3 that incorporates optimisation within the SMT solver itself, including the ability to compute Pareto fronts [19]. Our work in [3] scales better if we use Z3opt instead of Z3 plus our own optimisation seen in Figure 12 in the appendix. However, we were not able to use Z3opt successfully for the work reported in this paper, which may be due to the fact that we here work with non-linear objective functions.

Next, we discuss additional work on sensitivity analysis of BBNs. One such analysis studies the sensitivity of queries in BBNs to changes of a sole parameter, including an understanding of which changes would realise a given query constraint [7]. It seems possible to extend such work to multiple parameters at moderate computational overheads [6]. In [23] it is noted that naive Bayesian classifiers perform quite well even in the presence of inaccuracies, and that standard sensitivity functions suffice to describe scenario sensitivities [23].

In [11], methods from constructive real analysis are used to decide whether a formula is satisfiable if the values occurring in it can be perturbed by at most a specified, uniform value $\delta > 0$. This approach can support a good range of non-linear functions, including some transcendental ones, and can be applied to solutions of Lipschitz-continuous ordinary differential equations. This should therefore also enable a form of robust optimisation.

Our introduction is similar to the motivation given in [22], which poses a problem to the European Study Group with Industry (ESGI 107), held in Manchester in March 2015 [22]. Although our introduction shares this exposition of the problem, our paper advocates the use of BBNs, and cBBNs as their suitable abstractions, as one method of probing scenarios in application domain.

We view our approach as complementing other approaches in that problem domain – be they based on game theory, economic considerations of trust cul-

tures, policy and reputation based formalisms, dynamical systems and so forth. For example, we considered predicates asking whether the probability of a cBBN node can be above some threshold; and such predicates may inform rules within policy-based languages that evaluate trust – of which the language Peal and its tool PEALT is a more recent example [12].

6 Conclusions

We have proposed the use of BBNs in the modelling and assessment of nuclear arms verification scenarios, because such networks have several desirable features, e.g., their ability to represent both subjective and objective data that can interact in the model. BBNs formulated for this problem domain contain **Trust**, e.g., in the form of biases expressed as probabilities; and they capture **Degrees of Belief** by computing probabilities of events. However, in this problem domain it is paramount to assess the **Confidence** that we have in such degrees of belief. Yet in this problem space **Confidence** is hard to come by, given that little or no prior data are available to inform probabilities within model BBNs.

In this paper, we addressed this modelling problem by abstracting BBNs to constrained BBNs, which are subject to logical constraints and whose probabilities may contain symbolic uncertainties. We then addressed the corresponding analysis problem by representing these cBBNs in Satisfiability Modulo Theories, so that SMT solvers can answer queries about one or more of such cBBNs.

We demonstrated these new capabilities by developing constrained BBNs that model a particular arms control verification scenario, and by then analysing scenario-specific queries over those constrained BBNs but expressed in SMT. The types of queries that we analysed included the optimisation of the overall probability of an event in a constrained BBN, optimisation for hard evidence and its resulting model update, optimisation to determine worst-case differences between two cBBNs that model the same scenario, the computation of threshold ranges for which two constrained BBNs would inform decisions in the same manner, and worst-case sensitivities of critical nodes in a constrained BBN.

Our approach has several advantages: the query language is open-ended, queries merely have to be expressible in SMT; satisfiability witnesses for queries found by an SMT solver subsume the description of concrete BBNs that can subsequently be fed into BBN tools for external validation and feedback to non-expert users; and we may add logical constraints freely, for example to provide consistency between levels of abstractions of two or more constrained BBNs.

In future work, we want to design a domain-specific language in which we can specify constrained BBNs as well as a host of analysis methods, including those represented in this paper. And we want to write code generators that transform such specifications into SMT code. Finally, it would be of great interest to certify unsatisfiability results (e.g. that a computed maximum probability really is a global maximum). In that context, it is worth noting that Z3 can provide proofs of unsatisfiability, and there has been work on independently certifying such proofs in interactive theorem provers [5].

Acknowledgements: The authors from Imperial College London would like to thank AWE for sponsoring a PhD studentship under which the research reported in this paper was carried out.

Open Access of Research Data and Code: The Python code for the queries and models of this paper, and raw SMT analysis results are publicly available at `https://bitbucket.org/pjbeaumont/beaumontevanshuthplantesorics2015/`

References

1. Barrett, C., de Moura, L.M., Ranise, S., Stump, A., Tinelli, C.: The SMT-LIB initiative and the rise of SMT - (HVC 2010 award talk). In: Hardware and Software: Verification and Testing - 6th International Haifa Verification Conference, HVC 2010, Haifa, Israel, October 4-7, 2010. Revised Selected Papers. p. 3 (2010)
2. Barrett, C., Tinelli, C.: CVC3. In: Damm, W., Hermanns, H. (eds.) Proceedings of the 19th International Conference on Computer Aided Verification (CAV '07). Lecture Notes in Computer Science, vol. 4590, pp. 298–302. Springer-Verlag (Jul 2007), berlin, Germany
3. Beaumont P., Evans N., Huth M., and Plant T.: Modelling and analysis of constrained iterative systems: a case study in nuclear arms control. Submitted to AVoCS 2015 in June 2015.
4. Ben-Tal A., El Ghaoui L., and Nemirovski, A.: Robust Optimization. Princeton Series in Applied Mathematics, Princeton University Press, 9-16 (2009)
5. Böhme, S., Weber, T.: Fast LCF-style proof reconstruction for Z3. In: Interactive Theorem Proving, First International Conference, ITP 2010, Edinburgh, UK, July 11-14, 2010. Proceedings. pp. 179–194 (2010)
6. Chan, H., Darwiche, A.: Sensitivity analysis in bayesian networks: From single to multiple parameters. CoRR abs/1207.4124 (2012)
7. Chan, H., Darwiche, A.: When do numbers really matter? CoRR abs/1408.1692 (2014)
8. Cousot, P., Cousot, R.: Abstract interpretation: past, present and future. In: Joint Meeting of the Twenty-Third EACSL Annual Conference on Computer Science Logic (CSL) and the Twenty-Ninth Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), CSL-LICS '14, Vienna, Austria, July 14 - 18, 2014. p. 2 (2014)
9. Cozman, F.G.: Credal networks. Artif. Intell. 120(2), 199–233 (2000)
10. Fenton, N., Neil, M.: Risk Assessment and Decision Analysis with Bayesian Networks. CRC Press (2013)
11. Gao, S., Avigad, J., Clarke, E.M.: Delta-complete decision procedures for satisfiability over the reals. CoRR abs/1204.3513 (2012)
12. Huth, M., Kuo, J.H.: PEALT: an automated reasoning tool for numerical aggregation of trust evidence. In: Tools and Algorithms for the Construction and Analysis of Systems - 20th International Conference, TACAS 2014, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2014, Grenoble, France, April 5-13, 2014. Proceedings. pp. 109–123 (2014)
13. Lin X., Janak S. L., and Floudas Ch. A.: A new robust optimization approach for scheduling under uncertainty: I. Bounded uncertainty. Computers & Chemical Engineering 28(6-7): 1069-1085 (2004)

14. Kingdom of Norway and the United Kingdom of Great Britain and Northern Ireland: The Uniited Kingdom – Norway Initiative: Further Research into the Verification of Nuclear Warhead Dismantlement. In: 2015 Review Conference on the Parties to the Treaty on the Non-Proleferation of Nuclear Weapons, NPT/CONF 2015, WP.31, New York, USA, 27 April - 22 May 2015
15. Kupferman, O., Vardi, M.Y.: Vacuity detection in temporal model checking. STTT 4(2), 224–233 (2003)
16. Kwan, M.Y.K., Overill, R.E., Chow, K., Tse, H., Law, F.Y.W., Lai, P.K.Y.: Sensitivity analysis of bayesian networks used in forensic investigations. In: Advances in Digital Forensics VII - 7th IFIP WG 11.9 International Conference on Digital Forensics. pp. 231–243 (2011)
17. Lauritzen, S.L., Spiegelhalter, D.J.: Local computations with probabilities on graphical structures and their application to expert systems. Journal of the Royal Statistical Society. Series B (Methodological) 50(2), 157–224 (1988)
18. Ministry of Defence of the United Kingdom: the UK/Norway initiative: report on the UKNI nuclear weapons states workshop (March 2010)
19. Bjørner N., Phan A.-D., and Fleckenstein L.: νZ - An Optimizing SMT Solver. In: Proc. of TACAS 2015, LNCS 9035, Springer, pp. 194-199 (2015)
20. de Moura, L.M., Bjørner, N.: Z3: An efficient SMT solver. In: TACAS. pp. 337–340 (2008)
21. Pita J., John R., Maheswaran R. T., Tambe M., Yang R., and Kraus S.: A robust approach to addressing human adversaries in security games. In Proc. of AAMAS 2012: 1297-1298 (2012)
22. Plant, T., Stapleton, M.: Decision support for nuclear arms control. Problem Statement for ESGI 107, Manchester, UK (23-27 March 2015)
23. Renooij, S., van der Gaag, L.C.: Evidence and scenario sensitivities in naive Bayesian classifiers. Int. J. Approx. Reasoning 49(2), 398–416 (2008)
24. US NISA, US NAPC, UK Ministry of Defence and AWE: Joint U.S. - U.K. Report on Technical Cooperation for Arms Control, 2015.

A Ancillary material

In order to make this paper more self-contained, we provide in this appendix two figures that show probability tables of our more detailed Bayesian Belief Network and details of our optimisation algorithms, respectively.

Box	
A	B
0.5	0.5

Ability to Build	
Y	N
0.6	0.4

Conversation Belief	
T	F
0.75	0.25

Saturates/Intention		Manufactured Gamma T	Manufactured Gamma F
T	T	0.99	0.01
T	F	0.01	0.99
F	T	0.3	0.7
F	F	0.0	1.0

Box	Intention to mislead T	Intention to mislead F
A	0.2	0.8
B	0.1	0.9

Box/Intention		Deliberately saturates T	Deliberately saturates F
A	T	0.8	0.2
A	F	0.02	0.98
B	T	0.7	0.3
B	F	0.01	0.99

Ratio of Pu	Detector Light G	Detector Light R
Significantly less	0.05	0.95
Less	0.2	0.8
Around 10:1	0.01	0.99
More	0.2	0.8
Significantly more	0.95	0.05

Ability/Conversation/Detector			Believe Weapons Present T	Believe Weapons Present F
Y	T	G	0.99	0.01
Y	T	R	0.6	0.4
Y	F	G	0.55	0.45
Y	F	R	0.3	0.7
N	T	G	0.85	0.15
N	T	R	0.5	0.5
N	F	G	0.5	0.5
N	F	R	0.2	0.8

Saturates	Gamma	Box	Significantly less	Less	Around 10:1	More	Significantly more
T	T	G	0.4	0.09	0.02	0.09	0.4
T	T	R	0.4	0.1	0.0	0.1	0.4
T	F	G	0.5	0.4	0.1	0.0	0.0
T	F	R	0.2	0.2	0.2	0.2	0.2
F	T	G	0.0	0.1	0.25	0.25	0.4
F	T	R	0.0	0.0	0.1	0.3	0.6
F	F	G	0.0	0.15	0.7	0.15	0.0
F	F	R	0.2	0.2	0.2	0.2	0.2

Fig. 11. Probability tables for model $\mathcal{B}'$ of the arms verification control scenario. The last two tables specify node Believe Weapons Present and node Ratio of Pu, respectively

```
def maxopt(X, delta):

# unbounded search begins
    r = s.check()
    if r == unsat:
        return unsat
    else:
        t = s.model()

    while r == sat:
        s.push()
        s.add(X > 2*t[X])
        r = s.check()
        if r == sat:
            t = s.model()
        s.pop()
# unbounded search ended

# bisection method begins
    v = t[X]
    v = float(v.as_decimal(10)[:-1])
    max = 2*v
    min = v
    while (max-min) > delta:
        s.push()
        s.add(((max-min)/2)+min <= X)
        r = s.check()
        if r == sat:
             min = ((max-min)/2)+min
        else:
             max = ((max-min)/2)+min
        s.pop()
    y = (max_min)/2
# bisection method ended
return y
```

Fig. 12. Pseudo-code that returns the global maximum of real variable X within an accuracy of $\delta > 0$, where X is declared in the SMT input and is the subject of this optimisation. Variable `s` is an instance of the SMT solver, and expression `s.model()` refers to a witness found by that solver on its current input. The use of `float` above controls the mantissas of `max` and `min` to be commensurate with the desired accuracy δ to mitigate the complexity of reasoning. Directives `push` and `pop` control the visibility of assertions for incremental satisfiability checks