An Application of Evidential Networks to Threat Assessment

A. Benavoli[†], B. Ristic[‡], A. Farina[§], M. Oxenham[‡], L. Chisci[†]

Abstract

Decision makers operating in modern defence theatres need to comprehend and reason with huge quantities of potentially uncertain and imprecise data in a timely fashion. In this paper, an automatic information fusion system is developed which aims at supporting a commander's decision making by providing a *threat assessment*, that is an estimate of the extent to which an enemy platform poses a threat based on evidence about its intent and capability. Threat is modelled by a network of entities and relationships between them, while the uncertainties in the relationships are represented by belief functions as defined in the theory of evidence. To support the implementation of the threat assessment functionality, an efficient valuation-based reasoning scheme, referred to as an *evidential network*, is developed. To reduce computational overheads, the scheme performs local computations in the network by applying an inward propagation algorithm to the underlying binary join tree. This allows the dynamic nature of the external evidence, which drives the evidential network, to be taken into account by recomputing only the affected paths in the binary join tree.

Keywords: Valuation-based system, threat assessment, local computations, theory of evidence, Dempster-Shafer theory

[†] DSI, University of Florence, Italy; email: benavoli@gmail.com, chisci@dsi.unifi.it

[‡] DSTO, ISR Division, Australia; email: {branko.ristic,martin.oxenham}@dsto.defence.gov.au

[§] SELEX Sistemi Integrati, Italy; email: afarina@selex-si.com

I. Introduction

Situation and threat assessment are two important interdependent information fusion concepts which are usually treated jointly by the military command and control (C^2) process. According to [1], situation assessment establishes a view of the battlespace in terms of the observed activities, events, manoeuvres, locations and organisational aspects of the enemy force elements and from this view infers what is happening or what is going to happen on the battlefield. Threat assessment, on the other hand, estimates the degree of severity with which the engagement events will occur and its significance is in proportion to the perceived capability of the enemy to carry out its hostile intent. An essential prerequisite for winning a battle is for the decision maker (commander) to be aware of the current situation and threat rapidly in order to act properly and in a timely fashion.

The amount of data and information potentially relevant and available to a decision maker in modern warfare far exceeds the human ability to review and comprehend them in a timely manner. Moreover, the decisions usually have to be made under very stressful conditions which adversely affect humans and make them prone to error. All this leads to the need for the development of an automatic knowledge-based information fusion system that will support the commander's decision process in a reliable, timely and consistent manner [2]. Similar problems exist in other fields of human endeavour (management of commercial enterprises, medical diagnosis, etc), although the military C^2 domain is particularly challenging due to the inherently incomplete, uncertain and imprecise data.

A review of the early (pre 1990s) attempts at building knowledge-based expert systems is presented in [1, Ch.9]. The main drawback with these early attempts was the lack of a means and associated difficulties in handling uncertain domain knowledge and imprecise or non-specific evidence. The invention of *Bayesian networks* [3] in the mid 1980s for knowledge representation and probabilistic inference represented the next important stepping stone in the development of expert systems. Since then, Bayesian networks have been the main technique reported in the literature for constructing situation assessment [4], [5], [6], threat assessment [7] and intent assessment [8], [9] systems. Bayesian networks are based on the assumption that all data (domain knowledge and accumulated evidence) can be conveniently represented by probability functions. In reality, this may not always be the case and so, as alternative to Bayesian networks, which rely on a representation of the uncertain information in terms of probability functions, other network-based systems [10], [11] employing alternative uncertainty formalisms, such as possibility theory [12], [13], [14] and the theory of evidence (or the belief function theory) [15], [16], have been developed.

In 1989 Shenoy [17], [18] introduced the concept of a *valuation-based system* (VBS) which provides a general framework for managing uncertainty in expert systems[1]. There exist specialisations of the VBS for each of the three major theories of uncertainty, namely probability theory, possibility theory and the theory of evidence[2]. In a VBS, knowledge is represented by a network of variables (nodes) corresponding to the entities of the domain (and their states), and of links (edges) representing the relationships between these entities. For solving a particular problem, we first need to build a network model in terms of these nodes and links. Then we associate a *valuation* to each link which encapsulates the information (based on our domain knowledge and prior information) about how to propagate evidence and uncertainty from one entity to another via that link. Inference within a VBS is performed via two operators called *combination* and *marginalisation*. Combination corresponds to the aggregation of knowledge, while marginalisation refers to the focussing (coarsening) of it. Typically, we draw inferences on a small subset of variables within a valuation-based network. A "brute-force" approach to reasoning within a VBS would be to compute the joint valuation for the entire network and then to marginalise it to the subset of variables of interest for decision making. The trouble with this approach, however, is that it becomes computationally intractable even for small scale problems. A better alternative to the brute-force approach is to compute the required marginals of the joint valuation without explicitly computing the joint valuation.

In the Bayesian network context, several architectures [20] have been proposed for exact computation of marginals of multivariate discrete probability distributions. One of the pioneering architectures for computing marginals was proposed by Pearl [3] for multiply connected Bayesian networks. In 1988, Lauritzen and Spiegelhalter [21] proposed an alternative architecture for computing marginals in join trees (also known as junction or clique trees) that applies to any Bayesian network. Subsequently, Jensen et al. [22], [23] proposed a modification of the Lauritzen-Spiegelhalter architecture, which is known as the Hugin architecture, since this architecture is implemented in Hugin, a software tool developed by the same group. This architecture has been generalized by Lauritzen and Jensen [24] so that it applies more generally to other domains including the Dempster-Shafer's belief function theory. Inspired by the work of Pearl, Shenoy and Shafer [11] first adapted and generalized Pearl's architecture to the case of finding marginals of joint Dempster-Shafer belief functions in join trees. Later, motivated by the work of Lauritzen and Spiegelhalter [21] for the case of probabilistic reasoning, they proposed the VBS framework for computing marginals in join trees and established the set of axioms that combination and marginalisation need to

[1]In addition to being a framework for managing uncertainty, VBSs have been used in optimisation problems, constraint satisfaction problems, etc [18].

[2]Other examples of VBSs have been developed for handling uncertain information, such as *assumption-based systems* [19] which are based on propositional logic.

satisfy in order to make the local computation concept applicable [25]. These axioms are satisfied for all three major theories of uncertainty mentioned earlier. In 1997, Shenoy [26] proposed a refinement of junction trees, called binary join trees, designed to improve the computational efficiency of the Shenoy-Shafer architecture. In this paper we have chosen to focus on VBSs in the context of the theory of evidence, due to the expressive power of belief functions which can represent both classical probability functions and possibility/necessity functions [27, Ch.2]. This is particularly important when the valuations need to represent domain knowledge that is expressed in the form of uncertain implication rules [28], [29]. In order to emphasise this aspect of our work, we refer to the resulting reasoning networks as *evidential networks*. In the paper we develop a representative model of threat in the context of air defence and implement it using an evidential network. Local computations in the network are performed using the inward propagation algorithm on the binary join tree [30]. We introduce a modified version of the standard inward propagation algorithm, which takes into account the dynamic nature of input valuations (the external evidence which drives the evidential network) by recomputing only the affected paths in the binary join tree.

The paper is organised as follows. Section 2 describes valuation-based systems and the algorithms for local computation. Section 3 reviews the main concepts and tools from the theory of evidence. Section 4 develops the entities and relationships of a threat model cast in terms of an evidential network. Section 5 presents the numerical analysis and results for the proposed reasoning scheme. Finally, Section 6 discusses the conclusions drawn from the study and possible avenues for further research.

II. Valuation based systems

A. Networks and axioms for local computation

A valuation based system is a framework for knowledge representation and inference. Real-world problems are modelled in this framework by a network of interrelated entities, called variables. The relationships between variables (possibly uncertain or imprecise) are represented by the functions called valuations. The two basic operations for performing inference in a VBS are combination and marginalization. Throughout the paper we will deal with discrete-valued variables characterised by finite sets of possible values. Let x denote a variable in a VBS; the set of its possible values will be denoted by $\mathbf{\Theta}_x$ and referred to as the *frame* of x.

In a nutshell, a VBS [18] consists of a 5-tuple $\{\mathbf{V}, \mathbf{\Theta_V}, \mathbf{\Phi_V}, \oplus, \downarrow\}$, where $\mathbf{V}$ denotes the set of all variables in the model, $\mathbf{\Theta_V} = \{\mathbf{\Theta}_x : x \in \mathbf{V}\}$ is the set of frames of all variables, $\mathbf{\Phi_V} = \cup\{\mathbf{\Phi_D} : \mathbf{D} \subseteq \mathbf{V}\}$ denotes the set of all valuations, $\oplus$ is the combination operator and $\downarrow$ is the marginalization operator. Further explanation follows.

- **Variables and Frames**. For a subset of variables $\mathbf{D} \subseteq \mathbf{V}$, frame $\mathbf{\Theta_D}$ denotes the Cartesian product of the values of the variables $x \in \mathbf{D}$, that is $\mathbf{\Theta_D} \stackrel{\triangle}{=} \times\{\mathbf{\Theta}_x : x \in \mathbf{D}\}$, with $\times$ denoting the Cartesian product. The elements of $\mathbf{\Theta_D}$ are referred to as *configurations*. For example, suppose $\mathbf{D} = \{x, y, z\}$ is a subset of variables in a VBS, and that their frames are specified as follows: $\mathbf{\Theta}_x = \{x_1, x_2\}$, $\mathbf{\Theta}_y = \{y_1, y_2\}$ and $\mathbf{\Theta}_z = \{z_1, z_2\}$. Then the frame of $\mathbf{D}$ consists of 8 configurations and is given by:

$$\mathbf{\Theta_D} = \{(x_1\, y_1\, z_1), (x_1\, y_1\, z_2), \ldots, (x_2\, y_2\, z_2)\}.$$

- **Valuations**. Valuations are primitives in the VBS framework. A valuation φ represents some knowledge about the possible values of a set of variables $\mathbf{D}$. More precisely, given $\mathbf{D} \subseteq \mathbf{V}$, a valuation $\varphi : \mathbf{\Theta_D} \to [0, 1]$ is a function mapping the frame of $\mathbf{D}$ into the interval $[0, 1]$. The set of variables, on which the valuation is defined, will be denoted as $d(\varphi)$ and called the "domain" of φ. The symbol $\mathbf{\Phi_D}$ denotes the set of valuations for the set of variables $\mathbf{D}$, that is $\mathbf{\Phi_D} \stackrel{\triangle}{=} \{\varphi : d(\varphi) = \mathbf{D}\}$.
- **Combination**. Combination $\oplus$ is a binary function on valuations, $\oplus : (\mathbf{\Phi_V}, \mathbf{\Phi_V}) \to \mathbf{\Phi_V}$. Given two valuations $\varphi_1, \varphi_2 \in \mathbf{\Phi_V}$ defined on the domains $\mathbf{D}_1 \subseteq \mathbf{V}$ and $\mathbf{D}_2 \subseteq \mathbf{V}$, respectively, the combination $\varphi_1 \oplus \varphi_2$ is a valuation on domain $\mathbf{D} = \mathbf{D}_1 \cup \mathbf{D}_2$. Formally we write this as $d(\varphi_1 \oplus \varphi_2) = d(\varphi_1) \cup d(\varphi_2) = \mathbf{D}_1 \cup \mathbf{D}_2$.
- **Marginalization**. Marginalization $\downarrow$ is a binary operation and is used for focusing the knowledge onto a smaller domain, $\downarrow$: $(\mathbf{\Phi_V}, 2^{\mathbf{V}}) \to \mathbf{\Phi_V}$. If φ is a valuation for the domain $\mathbf{D} \subseteq \mathbf{V}$ and $\mathbf{D}_1 \subseteq \mathbf{D}$, then $\varphi^{\downarrow \mathbf{D}_1}$ is a valuation on the domain $\mathbf{D}_1$. Hence, it follows that $d(\varphi^{\downarrow \mathbf{D}_1}) = \mathbf{D}_1$.

Instead of marginalization another basic operation called *variable elimination* can be defined and denoted as $\varphi^{-x} \stackrel{\triangle}{=} \varphi^{\downarrow d(\varphi) \setminus \{x\}}$ with $x \in \mathbf{V}$. Note that $x \notin d(\varphi)$ implies $\varphi^{-x} = \varphi$.

The straightforward ("brute-force") approach for making inference in a valuation network is to compute the joint valuation on $\mathbf{V}$, that is to combine sequentially all the valuations in the model and then to marginalize this joint valuation to the sub-domain of interest $\mathbf{D}^o$ afterwards. However, when there are many variables in the model, computing the joint valuation directly becomes computationally intractable. Clearly the number of variables increases with each combination and the complexity grows exponentially with the number of variables. For instance, if there are n variables and each variable can assume m different values (i.e. each variable has m configurations in its frame), then there are m^n configurations in the joint domain of all variables. One way for reducing this complexity is to take advantage of the local structure of the problem. In most cases, complex problems can be decomposed into sub-problems involving a smaller number of variables. Furthermore, only a few variables are often of interest for decision making, while the remaining ones are auxiliary (non-interesting) variables, used only to model the problem. The fundamental idea of local computation [3], [18], [21], [22], [23], [24], [30] is to exploit the local structure of the problem to calculate the marginals of the joint valuation without explicitly computing the joint valuation. This is done by combining the valuations on small groups of variables, such that the non-interesting variables are eliminated one-by-one. At the end of this process the final result is the valuation on the variables of interest. This is possible if the following axioms are satisfied [25], [26].

1) **Commutativity and associativity of combination**: combination is commutative and associative in $\mathbf{\Phi_V}$.
2) **Order of deletion does not matter**: if $\varphi \in \mathbf{\Phi_V}$ is a valuation and x_1 and x_2 are two variables in $\mathbf{D} = d(\varphi)$, then $(\varphi^{\downarrow(\mathbf{D}\backslash\{x_1\})})^{\downarrow(\mathbf{D}\backslash\{x_1,x_2\})} = (\varphi^{\downarrow(\mathbf{D}\backslash\{x_2\})})^{\downarrow(\mathbf{D}\backslash\{x_1,x_2\})}$.
3) **Distributivity of marginalization over combination**: if $\varphi_1, \varphi_2 \in \mathbf{\Phi_V}$ are valuations with domains $\mathbf{D}_1$ and $\mathbf{D}_2$, respectively, and x is a variable such that $x \in D_2$ but $x \notin D_1$ then $(\varphi_1 \oplus \varphi_2)^{\downarrow((\mathbf{D}_1\cup\mathbf{D}_2)\backslash\{x\})} = \varphi_1 \oplus (\varphi_2^{\downarrow(\mathbf{D}_2\backslash\{x\})})$.

Axiom 2 says that if a valuation has to be marginalized to a smaller sub-domain, then the order in which the variables are eliminated is irrelevant. Axiom 3 is the fundamental axiom for the local computation. It states that the valuation $\varphi_1 \oplus (\varphi_2^{\downarrow(\mathbf{D}_2\backslash\{x\})})$ can be obtained without computing $(\varphi_1 \oplus \varphi_2)$. This property allows substantial savings on computational resources, because the combination $(\varphi_1 \oplus \varphi_2)$ is on the frame of the variables in $\mathbf{D}_1 \cup \mathbf{D}_2$, while the combination $\varphi_1 \oplus (\varphi_2^{\downarrow(\mathbf{D}_2\backslash\{x\})})$ is on the frame of $(\mathbf{D}_1 \cup \mathbf{D}_2)\backslash\{x\}$.

Notice that, like the junction tree algorithm for Bayesian networks, this method is not an approximation. In fact, if these axioms are satisfied, the result obtained by applying the local computation paradigm is exactly equivalent to that provided by the brute-force approach. For all major theories of uncertainty it can be proved that combination and marginalization satisfy the axioms for local computation [25]. In the next section, we describe an algorithm for performing inference via a VBS using local computation.

B. Fusion algorithm

The core of the VBS is the *fusion algorithm* [18], [30], which allows to perform inference via a VBS using local computation. Let $\mathbf{\Psi} = \{\varphi_1, \varphi_2, \ldots, \varphi_r\} \subseteq \mathbf{\Phi}$ be a given set of valuations and $\mathbf{D}^o \subseteq \mathbf{V}$, with $\mathbf{V} = d(\varphi_1) \cup d(\varphi_2) \cup \cdots \cup d(\varphi_r)$, the domain of interest for decision making. The fundamental operation of the fusion algorithm is to delete successively all variables $x \in \mathbf{\Delta}$, where $\mathbf{\Delta} \triangleq \mathbf{V}\backslash\mathbf{D}^o$ is the set of variables of no interest in the VBS. The variables can be deleted in any sequence, since according to Axiom 2 all deletion sequences lead to the same result. However, different deletion sequences can imply a different computational burden. Finding an optimal elimination sequence is an NP-complete problem [18], but there exist several heuristics for finding a good elimination sequence [29], [31], [32].

In the fusion algorithm, the marginal of the joint valuation is computed by successively eliminating all the variables in $\mathbf{\Delta}$. With respect to the variable $x \in \mathbf{\Delta}$ to be eliminated, two subsets of valuations can be defined

$$\mathbf{\Psi}_x \triangleq \{\varphi \in \mathbf{\Psi} : x \in d(\varphi)\} \text{ and } \mathbf{\Psi}_{\bar{x}} \triangleq \{\varphi \in \mathbf{\Psi} : x \notin d(\varphi)\}$$

As a consequence of axiom 3, only the valuations in $\mathbf{\Psi}_x$ are affected by the elimination of x. Thus, the remaining set of valuations after eliminating x from $\mathbf{\Psi}$ is

$$Fus_x\{\varphi_1, \varphi_2, \ldots, \varphi_r\} \triangleq \{\oplus \mathbf{\Psi}_x\}^{\downarrow(\mathbf{S}\setminus\{x\})} \cup \mathbf{\Psi}_{\bar{x}} \triangleq \{\varphi_{r+1}\} \cup \mathbf{\Psi}_{\bar{x}} \tag{1}$$

where $\mathbf{S} \triangleq \bigcup_{\varphi_i \in \mathbf{\Psi}_x} d(\varphi_i)$. Note that the Fus_x operation in (1) amounts to the union of all valuations not involving x together with the single valuation φ_{r+1}. The latter is obtained by combining all valuations involving x and then marginalizing the resulting valuation to $\mathbf{S}\setminus\{x\}$. The valuation on the domain of interest $\mathbf{D}^o$ can thus be obtained by recursively applying the fusion algorithm and deleting all variables in $\mathbf{\Delta} = \{x_1, x_2, \ldots, x_m\}$, i.e.

$$(\varphi_1 \oplus \varphi_2 \oplus \cdots \oplus \varphi_r)^{\downarrow\mathbf{D}^o} = \oplus\left\{Fus_{x_m}\left\{Fus_{x_{m-1}}\{\ldots Fus_{x_1}\{\varphi_1, \varphi_2, \ldots, \varphi_r\}\}\right\}\right\} \tag{2}$$

This technique allows a reduction in the computational load for two reasons: the beliefs are combined on local domains and the variable elimination keeps the domains of the combined beliefs, i.e. $d(\varphi_1 \oplus \varphi_2) = d(\varphi_1) \cup d(\varphi_2)$, to a reasonably small size.

a) Example 1.: Let us consider the set of valuations $\{\varphi_1, \varphi_2, \varphi_3, \varphi_4\}$ defined respectively on the domains $d(\varphi_1) = \{x_1, x_2\}$, $d(\varphi_2) = \{x_2, x_3\}$, $d(\varphi_3) = \{x_3, x_4\}$, $d(\varphi_4) = \{x_4, x_1\}$, where $x_1, \ldots, x_4$ are the variables of the problem. Assume that x_1 is the decision variable, i.e. $\mathbf{D}^o = \{x_1\}$. Then it follows that $\mathbf{\Delta} = \{x_2, x_3, x_4\}$ is the set of variables of no interest. The objective is to apply the fusion algorithm to compute the combined valuation $(\varphi_1 \oplus \varphi_2 \oplus \varphi_3 \oplus \varphi_4)^{\downarrow\mathbf{D}^o}$. The steps of the fusion algorithm are the following:

1) $\varphi_5 = (\varphi_1 \oplus \varphi_2)^{\downarrow(d(\varphi_1)\cup d(\varphi_2))\setminus\{x_2\}}, \quad Fus_{x_2} = \{\varphi_5\} \cup \mathbf{\Psi}_{\bar{x}_2} = \{\varphi_3, \varphi_4, \varphi_5\}$
   where $d(\varphi_5) = (d(\varphi_1) \cup d(\varphi_2))\setminus\{x_2\} = \{x_1, x_3\}$ is the domain of φ_5.
2) $\varphi_6 = (\varphi_3 \oplus \varphi_5)^{\downarrow(d(\varphi_3)\cup d(\varphi_5))\setminus\{x_3\}}, \quad Fus_{x_3} = \{\varphi_6\} \cup \mathbf{\Psi}_{\bar{x}_3} = \{\varphi_4, \varphi_6\}$
   where $d(\varphi_6) = (d(\varphi_3) \cup d(\varphi_5))\setminus\{x_3\} = \{x_1, x_4\}$

3) $\varphi_7 = (\varphi_4 \oplus \varphi_6)^{\downarrow (d(\varphi_4) \cup d(\varphi_6)) \setminus \{x_4\}}, \quad Fus_{x_4} = \{\varphi_7\} \cup \mathbf{\Psi}_{\bar{x}_4} = \{\varphi_7\}$
where $d(\varphi_7) = (d(\varphi_4) \cup d(\varphi_6)) \setminus \{x_4\} = \{x_1\}$

At the end of the last step the valuation Fus_{x_4}, defined on the domain of interest $\{x_1\}$, represents the solution of the problem. ■

C. Dynamic fusion

The fusion algorithm (2) works well if the valuations are static (invariant in time). However, we may want to compute the marginal of the variables of interest more than once, for example every time one or more valuations in the VBS change. In this case, we would need to repeat the application of the fusion algorithm every time any of the valuations in $\mathbf{\Psi}$ is changed. This would clearly be inefficient, since it would result in a lot of duplication in computation. To avoid this, it is more efficient to represent the VBS in the form of a binary join tree (BJT) and then to propagate the changes. A binary join tree is a join tree such that no node has more than three neighbors, one parent and two children. The binary join tree construction process is based on the fusion algorithm and the idea that all combinations between valuations should be carried out on a binary basis, i.e. two-by-two.

A BJT is a binary tree (N, E) of nodes $N = \{n_1, n_2, \ldots, n_f\}$ and edges $E = \{(n, m) : n, m \in N, n \neq m\}$. A node without children is called a *leaf*. A node without a parent is called a *root*. As such, a BJT is only a graphical representation of the fusion algorithm [18]. For this reason, like in the fusion algorithm, the structure of the BJT (i.e. nodes and edges) strongly depends on the elimination sequence $\mathbf{\Delta}$.

A BJT has the following characteristics.

- To each node n_i a subset of variables $\mathbf{D}_i \subseteq \mathbf{V}$ and a valuation $\varphi(n_i)$, such that $d(\varphi(n_i)) = \mathbf{D}_i$ are associated.
- The domain of the root of the BJT is such that $\mathbf{D}^o \subseteq d(root)$.
- Edges represent the order in which the valuations must be combined (in order to calculate the valuation of the root on $\mathbf{D}^o$).
- Nodes and edges represent steps of the fusion algorithm.
- A BJT has to satisfy the Markov property, which means that $\mathbf{D}_i \cap \mathbf{D}_j \subseteq \mathbf{D}_k$ for every pair of nodes n_i and n_j and for every node $n_k \in Path(n_i, n_j)$, where $Path(n_i, n_j)$ denotes the set of nodes on the path between n_i and n_j.

Note that the Markov property is one of the most important properties of the BJT, as will be discussed in Sec.

V-A. An algorithm for building a BJT is given in appendix .

In a BJT, marginals are computed by means of a message-passing scheme among the nodes. Initially only the valuations of leaves of the BJT are specified. The process of propagating the valuations from the leaves toward the root of a BJT is called *inward propagation* [18], [30] and can be implemented with the algorithm reported in appendix . The key feature of the BJT and inward propagation is that the combination operator is applied only at the non-leaf nodes of the tree, between their left and right children. The advantage of using inward propagation on a BJT instead of the fusion algorithm lies in the ability to re-use the computations of the inward phase if the marginals need to be re-computed. In this way, every time one or more valuations of the leaves of the BJT change, the inward phase re-calculates the valuations for all the nodes in the BJT which are affected by the change. That is, if n_i is the leaf whose valuation has changed, then the inward phase re-computes the valuations of all the nodes of the BJT along $Path(n_i, root)$.

Suppose the BJT has been constructed for the domain of interest $\mathbf{D}^o$, and the inward propagation has been carried out. Let us also assume that the domain of interest has changed. One way to carry out the inference would be to create a new BJT and to perform again inward propagation. However, there is a more efficient alternative, the so called *outward propagation* [30]. Outward propagation distributes the knowledge from the root to the leaves of the tree, by reversing the direction in which the messages are passed between nodes [30]. Note that in the threat assessment problem the set $\mathbf{D}^o$ is fixed and hence outward propagation is not used in the sequel.

In summary, a BJT can be seen as a data structure which allows the intermediate results of the combination process to be saved and the marginals to be computed efficiently.

III. Belief functions as valuations

A VBS with valuations expressed by belief functions (as defined in the theory of evidence) will be referred to as an *evidential network*. The theory of evidence satisfies all of the VBS axioms for local computation listed in Sec.II-A. In this section we review the main components and tools of the theory of evidence.

Let frame $\mathbf{\Theta}_h = \{h_1, h_2, \dots, h_n\}$ define a finite set of possible values of variable h in an evidential network. Elementary values h_i ($i = 1, \dots, n$) of the frame $\mathbf{\Theta}_h$ are assumed to be mutually exclusive and exhaustive so that $n = |\mathbf{\Theta}_h|$ is the cardinality of the frame. The beliefs about the actual value of the variable h are expressed on the subsets of $\mathbf{\Theta}_h$. The set containing all possible subsets of $\mathbf{\Theta}_h$, i.e. the power set of $\mathbf{\Theta}_h$, is denoted by $2^{\mathbf{\Theta}_h} = \{H : H \subseteq \mathbf{\Theta}_h\}$; its cardinality is 2^n. In this formalism, belief is represented by a so-called basic belief assignment (BBA) $m : 2^{\boldsymbol{\Theta}_h} \rightarrow [0,1]$, that satisfies $\sum_{H \subseteq \boldsymbol{\Theta}_h} m(H) = 1$. Thus for $H \subset \boldsymbol{\Theta}_h$, $m(H)$ is the part of the belief that supports H (i.e. the fact that the true value of h is in H), but due to the lack of further information, does not support any strict subset of H. The subsets H such that $m(H) > 0$ are referred to as *focal elements* of the BBA. The state of complete ignorance about the variable h is represented by a *vacuous* BBA defined as $m(H) = 1$ if $H = \boldsymbol{\Theta}_h$ and zero otherwise. Since the valuations in the evidential networks are BBAs, we denote them in the sequel by m in place of φ.

A. Combination

The combination operator in the theory of evidence is carried out using Dempster's rule of combination. Let the BBA $m_1^{\mathbf{D}_1}$ be defined on a domain (subset of variables) $d(m_1) = \mathbf{D}_1 \subseteq \mathbf{V}$. Similarly let $m_2^{\mathbf{D}_2}$ be another BBA defined on a domain $d(m_2) = \mathbf{D}_2 \subseteq \mathbf{V}$. If $d(m_1) \equiv d(m_2) = \mathbf{D}$, the two BBAs are combined directly using Dempster's rule [15]:

$$(m_1^{\mathbf{D}} \oplus m_2^{\mathbf{D}})(A) = \frac{\sum\limits_{B \cap C = A} m_1^{\mathbf{D}}(B)\, m_2^{\mathbf{D}}(C)}{1 - \sum\limits_{B \cap C = \emptyset} m_1^{\mathbf{D}}(B)\, m_2^{\mathbf{D}}(C)} \tag{3}$$

where A, B, C are subsets of the frame defined by the Cartesian product of the variables in $\mathbf{D}$; i.e. $A, B, C \subseteq \boldsymbol{\Theta}_{\mathbf{D}}$. If the two domains are different, $\mathbf{D}_1 \neq \mathbf{D}_2$, then before we apply Dempster's rule, we must extend both BBAs to the joint domain $\mathbf{D}_1 \cup \mathbf{D}_2$ in such a way that they express the same information before and after the extension (hence referred to as the *vacuous extension* and denoted by $\uparrow$). The vacuous extension of $m_1^{\mathbf{D}_1}$ to $\mathbf{D}_1 \cup \mathbf{D}_2$ is defined as [18]

$$m_1^{\mathbf{D}_1 \uparrow (\mathbf{D}_1 \cup \mathbf{D}_2)}(C) = \begin{cases} m_1^{\mathbf{D}_1}(A) & \text{if } C = A \times \boldsymbol{\Theta}_{\mathbf{D}_2}, A \subseteq \boldsymbol{\Theta}_{\mathbf{D}_1} \\ 0 & \text{otherwise.} \end{cases} \tag{4}$$

b) Example 2.: Suppose $\mathbf{V} = \{x, y, z\}$ with frames $\boldsymbol{\Theta}_x = \{x_1, x_2\}$, $\boldsymbol{\Theta}_y = \{y_1, y_2\}$ and $\boldsymbol{\Theta}_z = \{z_1, z_2\}$. Let $\mathbf{D}_1 = \{x\}$, and $\mathbf{D}_2 = \{y, z\}$, i.e. $\boldsymbol{\Theta}_{\mathbf{D}_1} = \boldsymbol{\Theta}_x$ and $\boldsymbol{\Theta}_{\mathbf{D}_2} = \{(y_1\, z_1), (y_1\, z_2), (y_2\, z_1), (y_2\, z_2)\}$. Let the BBA $m_1^{\mathbf{D}_1}$ be defined such that $m_1^{\mathbf{D}_1}(\{x_1\}) = 0.7$ and $m_1^{\mathbf{D}_1}(\{x_1, x_2\}) = 0.3$. Then the vacuous extension of $m_1^{\mathbf{D}_1}$ to $\mathbf{D}_1 \cup \mathbf{D}_2$ is given by: $m_1^{\mathbf{D}_1 \uparrow (\mathbf{D}_1 \cup \mathbf{D}_2)}(\{(x_1\, y_1\, z_1), (x_1\, y_1\, z_2), (x_1\, y_2\, z_1), (x_1\, y_2\, z_2)\}) = 0.7$ with the remaining belief of 0.3 assigned to $\boldsymbol{\Theta}_{\mathbf{D}_1 \cup \mathbf{D}_2} = \{(x_1\, y_1\, z_1),\ (x_1\, y_1\, z_2),\ (x_1\, y_2\, z_1),\ (x_1\, y_2\, z_2),\ (x_2\, y_1\, z_1),\ (x_2\, y_1\, z_2),\ (x_2\, y_2\, z_1),\ (x_2\, y_2\, z_2)\}$. ■

Dempster's rule of combination in the general case of possibly non-identical domains is then defined as:

$$m_1^{\mathbf{D}_1} \oplus m_2^{\mathbf{D}_2} = m_1^{\mathbf{D}_1 \uparrow (\mathbf{D}_1 \cup \mathbf{D}_2)} \oplus m_2^{\mathbf{D}_2 \uparrow (\mathbf{D}_1 \cup \mathbf{D}_2)}. \tag{5}$$

B. Marginalisation

Marginalisation is a projection of a BBA defined on domain $\mathbf{D}$ onto a BBA defined on a coarser domain $\mathbf{D}' \subseteq \mathbf{D}$. Formally we write:

$$m^{\mathbf{D}\downarrow\mathbf{D}'}(A) = \sum_{B\downarrow A} m^{\mathbf{D}}(B) \tag{6}$$

where the summation in (6) is over all $B \subseteq \mathbf{\Theta}_{\mathbf{D}}$ such that the configurations in B reduce to the configurations in $A \subseteq \mathbf{\Theta}_{\mathbf{D}'}$ by the elimination of variables $\mathbf{D} \setminus \mathbf{D}'$.

c) Example 3.: Let $\mathbf{D} = \{x, y, z\}$ and $\mathbf{D}' = \{x, z\}$, with the frames of variables $\mathbf{\Theta}_x = \{x_1, x_2\}$, $\mathbf{\Theta}_y = \{y_1, y_2\}$ and $\mathbf{\Theta}_z = \{z_1, z_2, z_3\}$. Suppose BBA $m^{\mathbf{D}}$ has three focal sets:

$$m^{\mathbf{D}}(\{(x_1\, y_1\, z_1)\}) = 0.6$$

$$m^{\mathbf{D}}(\{(x_1\, y_1\, z_1), (x_1\, y_2\, z_2)\}) = 0.3$$

$$m^{\mathbf{D}}(\{(x_1\, y_1\, z_1), (x_1\, y_2\, z_1), (x_1\, y_2\, z_2)\}) = 0.1.$$

Then:

$$m^{\mathbf{D}\downarrow\mathbf{D}'}(\{(x_1\, z_1)\}) = 0.6$$

$$m^{\mathbf{D}\downarrow\mathbf{D}'}(\{(x_1\, z_1), (x_1\, z_2)\}) = 0.4$$

■

Remark. Marginalization is the inverse operation of extension, but, in general, extension is not the inverse of marginalization. For instance, consider a valuation φ and three generic sets $\mathbf{D}_1$, $\mathbf{D}_2$ and $\mathbf{D}_3$ such that $d(\varphi) = \mathbf{D}_2$ and $\mathbf{D}_1 \subseteq \mathbf{D}_2 \subseteq \mathbf{D}_3$; then it turns out that $(\varphi^{\uparrow\mathbf{D_3}})^{\downarrow\mathbf{D_2}} = \varphi$ but, in general, $(\varphi^{\downarrow\mathbf{D_1}})^{\uparrow\mathbf{D_2}} \neq \varphi$.

C. Representation of uncertain implication rules

Often expert knowledge is expressed in the form of uncertain implication rules, such as “if A then B” with a certain degree of confidence. Suppose there are two disjoint domains, $\mathbf{D}_1$ and $\mathbf{D}_2$ with associated frames $\mathbf{\Theta}_{\mathbf{D}_1}$ and $\mathbf{\Theta}_{\mathbf{D}_2}$, respectively. Formally, an implication rule is an expression of the form

$$A \subseteq \mathbf{\Theta}_{\mathbf{D}_1} \Rightarrow B \subseteq \mathbf{\Theta}_{\mathbf{D}_2}. \tag{7}$$

Furthermore, let us assume that this implication rule is valid only in a certain percentage of cases, i.e. with a probability (confidence) p such that $p \in [\alpha, \beta]$, with $0 \leq \alpha \leq \beta \leq 1$.

An implication rule can be expressed by a BBA using the principle of *minimum commitment* [33] and its instantiation referred to as the *ballooning extension* [33], [28]. Thus the implication rule of (7) can be expressed by a BBA consisting of 3 focal sets on the joint domain $\mathbf{D}_1 \cup \mathbf{D}_2$ [29]:

$$m^{\mathbf{D}_1 \cup \mathbf{D}_2}(C) = \begin{cases} \alpha, & \text{if } C = (A \times B) \cup (A^c \times \mathbf{\Theta}_{\mathbf{D}_2}) \\ 1-\beta, & \text{if } C = (A \times B^c) \cup (A^c \times \mathbf{\Theta}_{\mathbf{D}_2}) \\ \beta - \alpha, & \text{if } C = \mathbf{\Theta}_{\mathbf{D}_1 \cup \mathbf{D}_2} \end{cases} \tag{8}$$

where A^c is the complement of A in $\mathbf{\Theta}_{\mathbf{D}_1}$, and accordingly B^c is the complement of B in $\mathbf{\Theta}_{\mathbf{D}_2}$.

d) Example 4.: Let $\mathbf{D}_1 = \{x\}$, $\mathbf{D}_2 = \{y\}$, $\mathbf{\Theta}_x = \{x_1, x_2, x_3\}$, $\mathbf{\Theta}_y = \{y_1, y_2, y_3\}$, $A = \{x_1, x_2\}$ and $B = \{y_2\}$. Then the BBA representation of the rule $A \Rightarrow B$ with confidence $p \in [\alpha, \beta]$ is given by:

$$m^{\{x,y\}}(\{(x_1\, y_2), (x_2\, y_2), (x_3\, y_1), (x_3\, y_2), (x_3\, y_3)\}) = \alpha$$

$$m^{\{x,y\}}(\{(x_1\, y_1), (x_1, y_3), (x_2\, y_1), (x_2, y_3), (x_3\, y_1), (x_3\, y_2), (x_3\, y_3)\}) = 1-\beta$$

$$m^{\{x,y\}}(\{(x_1, y_1), (x_1, y_2), (x_1, y_3), (x_2, y_1), (x_2, y_2), (x_2, y_3), (x_3, y_1), (x_3, y_2), (x_3, y_3)\}) = \beta - \alpha.$$

■

Note that in the special case $\alpha = \beta$, the BBA has only two focal sets. Implication rules are sometimes used to express the valuations (BBAs) on the leaf nodes of a BJT.

D. Pignistic transformation

Belief functions cannot be directly used for decision making [34], hence we need to introduce a mapping of a belief measure to a probability measure. The pignistic transformation is the only such mapping satisfying the requisite linearity property [34]. Let $m^{\mathbf{D}}$ be a BBA defined on a subset of variables $\mathbf{D}$ with corresponding frame $\mathbf{\Theta}_{\mathbf{D}}$. The pignistic transform of $m^{\mathbf{D}}$ is defined for every element of the frame $\theta \in \mathbf{\Theta}_{\mathbf{D}}$ as follows [34]:

$$BetP(\theta) = \sum_{\theta \in A \subseteq \mathbf{\Theta}_{\mathbf{D}}} \frac{1}{|A|} \frac{m^{\mathbf{D}}(A)}{1 - m^{\mathbf{D}}(\emptyset)}. \tag{9}$$

$BetP$ is the probability measure that we use for decision making on the domain of interest $\mathbf{D}^o \subseteq \mathbf{V}$ within evidential networks.

IV. Threat model

In this section we introduce a model of threat in the context of an air-to-air engagement which draws on ideas from [1] and [35]. The model is shown in the form of an evidential network in Fig.1, where the variables are represented by circular nodes and the valuations (BBAs) by diamond shapes. The list of variables with explanations and frame definitions is given in Table I. Each valuation node is connected by edges to the subset of variables which define its domain. For example, the domain of valuation (BBA) m_1 consists of variables T, HI and C. Any pair of variables which are not directly connected are assumed to be conditionally independent. The domain of interest for decision making is the singleton $\mathbf{D}^o = \{\mathrm{T}\}$.

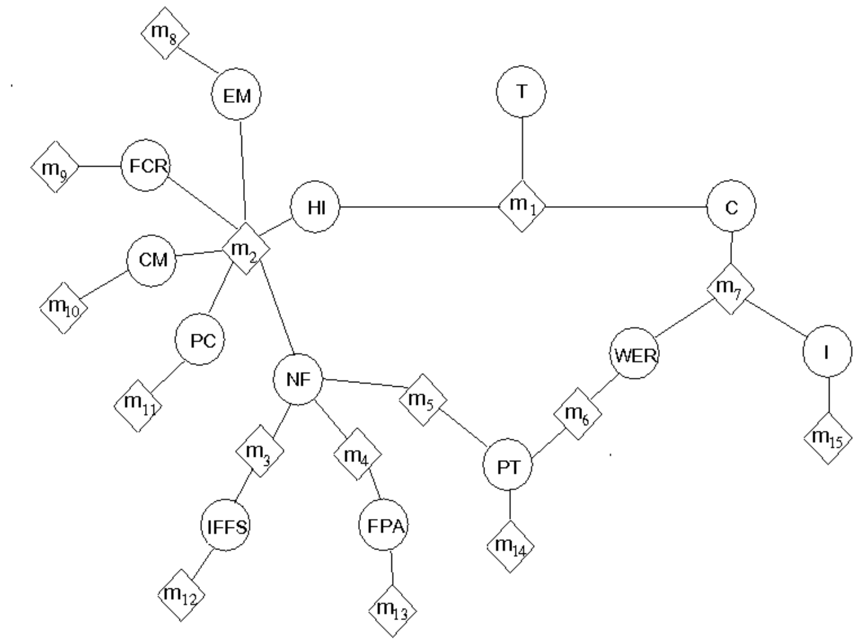

Fig. 1. *A model of threat assessment*

According to the threat model in Fig.1, variable T (threat) depends on the degree of hostile intent (HI) of the opponent and on its capability (C). Assuming the threat linearly related to both HI and C, we may choose to represent the valuation m_1 by the following rule: T=HI+C. Consider in the Cartesian product space T $\times$ HI $\times$ C the set of triples (t, h, c), such that $t = h + c$, where according to the frames of the variables in Table I, $t \in \{0, \dots, 10\}$, $h \in \{0, \dots, 6\}$ and $c \in \{0, \dots, 4\}$. Then we can represent the rule $T = HI + C$ by the following BBA:

$$\begin{aligned} m_1(\{&(0,0,0),(1,0,1),\dots,(4,0,4),\\ &(1,1,0),(2,1,1),\dots,(5,1,4),\\ &\dots\\ &(6,6,0),(7,6,1),\dots,(10,6,4)\}) = 1. \end{aligned} \tag{10}$$

TABLE I

Variables of the threat assessment model

Variable	Description	Frame	Explanation
T	Threat	$\{0, 1, \dots, 10\}$	0 none, 10 highest degree of T
HI	(Hostile) Intent	$\{0, 1, \dots, 6\}$	0 none (benign), 6 highest degree of HI
C	Capability	$\{0, 1, 2, 3, 4\}$	0 none, 4 highest degree of C
EM	Evasive manoeuvre	$\{0, 1\}$	0 is false, 1 is true
FCR	Fire Control Radar	$\{0, 1\}$	0 is OFF, 1 is ON
CM	Countermeasures	$\{0, 1\}$	0 is false, 1 is true
PC	Political climate	$\{0, 1\}$	0 is peace, 1 is war
NF	Non-friendly platform	$\{0, 1\}$	0 is false, 1 is true
IFFS	Correct IFF squawking	$\{0, 1\}$	0 is false, 1 is true
FPA	Flight plan agreement	$\{0, 1\}$	0 is false, 1 is true
PT	Platform type	$\{0, 1, \dots, 5\}$	E.g. 0 is EuroFighter, 1 is FA-22 raptor, etc.
WER	Weapon Engagement range	$\{0, 1, 2\}$	0 is small, 1 medium, 2 long range
I	Imminence	$\{0, 1, 2\}$	0 is low, 1 medium, 2 is high

This BBA has a single focal set consisting of 35 triples (t, h, c).

The degree of hostile intent (HI) is proportional to the evidence that the target (opponent) behaves in a hostile manner. In particular, the target may perform evasive manoeuvres (EM), it may employ countermeasures (CM), such as deception jamming or chaff, we may have evidence that it is not a friendly (NF) platform, and most importantly, its fire-control-radar (FCR) could be turned on (meaning it intends to fire a weapon soon). In addition, the political climate (PC) has an influence on the HI variable in the sense that the climate of political tension means that the target is more likely to have a hostile intent. The relationship between the six variables mentioned (HI,EM,FCR,CM,PC,NF), is captured by the valuation m_2. How this relationship may be represented by m_2 depends on many factors (doctrine, engagement rules, etc), but for the sake of illustration we adopt the following simple rule: HI = EM+2·FCR+CM+PC+NF. This rule reflects the fact that the FCR variable is weighted higher than other variables in contributing to the HI. The adopted rule is represented by the BBA m_2 defined on a 6 dimensional product space HI×EM×FCR×CM×PC×NF as follows:

$$\begin{aligned} & m_2(\{(0,0,0,0,0,0), (1,0,0,0,0,1), (1,0,0,0,1,0), \\ & \quad (2,0,0,0,1,1), (1,0,0,1,0,0) \dots, (2,0,1,0,0,0), \dots, (6,1,1,1,1,1)\}) = 1 \end{aligned} \tag{11}$$

Thus m_2 has a single focal set consisting of 32 six-tuples.

Identification friend or foe (IFF) is a radio interrogator device for positive identification of friendly aircraft. Variable IFFS is true if the target responds correctly to the interrogation. In order to define the valuation m_3 on domain $\{\text{NF}, \text{IFFS}\}$, suppose that we have confidence that in 95% to 100% of the cases if the IFFS is true, than the target is indeed a friend (i.e. NF $= 0$). On the other hand, suppose the evidence indicates that the lack of response to the IFF interrogation (IFFS=0) is due to the non-friendly (NF$= 1$) target only in 10 to 30% of the cases. We can then summarise "expert" knowledge about the domain $\{\text{NF}, \text{IFFS}\}$ by the following set of independent rules:

$$(\text{IFFS} = 1) \Rightarrow (\text{NF} = 0) \text{ with confidence between } 0.95 \text{ and } 1$$

$$(\text{IFFS} = 0) \Rightarrow (\text{NF} = 1) \text{ with confidence between } 0.10 \text{ and } 0.30$$

Then according to Sec.III-C, each of the rules above can be represented by a BBA; when the BBAs are combined by Dempster's rule, we obtain the following valuation on the product space NF $\times$ IFFS:

$$\begin{array}{llllllll}
m_3 & (\{ & (0,0), & (0,1), & & & \}) = & 0.6650 \\
m_3 & (\{ & (0,0), & (0,1), & (1,0) & & \}) = & 0.1900 \\
m_3 & (\{ & & (0,1), & (1,0) & & \}) = & 0.0950 \\
m_3 & (\{ & (0,0), & (0,1), & & (1,1) & \}) = & 0.0350 \\
m_3 & (\{ & & (0,1), & (1,0), & (1,1) & \}) = & 0.0050 \\
m_3 & (\{ & (0,0), & (0,1), & (1,0), & (1,1) & \}) = & 0.0100
\end{array} \tag{12}$$

Flight plans are plans filed by pilots with the local aviation authority prior to flying. They generally include basic information such as departure and arrival points, estimated time, etc. If there is evidence that an air target is flying in accordance with a flight plan (variable FPA $= 1$), then this is a strong indication that it is a friend (or neutral), i.e. NF$= 0$. Suppose we can again summarise expert knowledge about the domain $\{\text{FPA}, \text{NF}\}$ by the following set of rules:

$$(\text{FPA} = 1) \Rightarrow (\text{NF} = 0) \text{ with confidence between } 0.95 \text{ and } 1$$

$$(\text{FPA} = 0) \Rightarrow (\text{NF} = 1) \text{ with confidence between } 0.10 \text{ and } 0.30$$

As described above, these two rules can be translated to the corresponding BBA m_4 on its domain $\{\text{FPA}, \text{NF}\}$.

Suppose we have at our disposal a sensor such as an electronic support measures (ESM) system, which can report on the platform type (PT) variable. Valuation m_5 captures the expert knowledge which relates the PT to the

NF variable. Suppose this knowledge is represented by the following implication rule:

$$(\text{NF} = 1) \Rightarrow (\text{PT} \in \{3, 4, 5\}) \text{ with confidence between } 0.50 \text{ and } 1.$$

This rule represents our prior knowledge (e.g. from intelligence sources) that non-friendly aircraft in the battlespace of interest are of type 3, 4 or 5, with confidence at least of 50%.

For each PT, it is usually known a priori what types of weapons (and its capabilities) it carries [36]. Variable m_6 represents the relationship between the weapons engagement range (WER) variable and the PT. Suppose m_6 is defined by the following set of rules:

$$(\text{PT} \in \{0, 1\}) \Rightarrow (\text{WER} = 0) \text{ with confidence between } 0.40 \text{ and } 1$$

$$(\text{PT} \in \{2, 3\}) \Rightarrow (\text{WER} \in \{1, 2\}) \text{ with confidence between } 0.40 \text{ and } 1$$

$$(\text{PT} \in \{4, 5\}) \Rightarrow (\text{WER} = 2) \text{ with confidence between } 0.40 \text{ and } 1.$$

Variable C (capability) in our threat model is related to the WER and to the imminence (I) of an attack. The degree of imminence is measured by the distance, heading and speed of the target, and according to Table I can be low, medium or high. We define valuation m_7 by the following rule on the product space C $\times$ WER $\times$ I: C=WER+I. This rule captures the simple notion that the capability is high if the WER is large and the imminence is high. Thus m_7 is a BBA given by:

$$m_7(\{(0,0,0),(1,0,1),(2,0,2),(1,1,0),(2,1,1),(3,1,2),(2,2,0),(3,2,1),(4,2,2)\}) = 1.$$

Valuations m_1, m_2, $\ldots$, m_7 represent our prior domain knowledge of the problem. The remaining valuations m_8, m_9, $\ldots$, m_{15}, referred to as *input valuations*, are the drivers of the evidential network for threat assessment. Input valuations are initially represented by vacuous BBAs. As more evidence (from the surveillance sensors and other external sources) about the intruder and the situation become available, input valuations change and become more informative. The next section will present the numerical results obtained using the described evaluation network for various combinations of input valuations.

V. Numerical results and analysis

In this section we apply the VBS framework to determine the degree of threat posed by a hypothetical intruder in the considered air-to-air engagement problem. According to Table I, the degree of threat takes integer values in the range from 0 to 10 (0 being no threat, 10 being highest threat).

In Sec. IV we have introduced the main components of the VBS framework for the problem of interest. The set of variables consists of 13 elements, $\mathbf{V} = \{$T, HI, C, EM, FCR, CM, PC, NF, IFFS, FPA, PT, WER, I$\}$; the set of all valuations (BBAs) consists of 15 elements, $\mathbf{\Phi_V} = \{m_1, m_2, \ldots, m_{15}\}$, the domain of interest is the singleton $\mathbf{D}^o = \{$T$\}$ and the set of variables to be eliminated is $\mathbf{\Delta} = \mathbf{V} \backslash \{T\}$.

The following steps describe the process for solving the problem in the VBS framework.

1) construct the binary join tree;
2) initialize the leaves of the BJT with the BBAs;
3) apply the inward propagation algorithm;
4) marginalise the belief of the root of the BJT to $\mathbf{D}^o$;
5) apply the pignistic transformation.

A. The Binary Join Tree

Only three pieces of information are necessary to build a BJT: the set of variables of interest for decision making $\mathbf{D}^o$; the set of variables to be eliminated $\mathbf{\Delta}$ and the set of the valuations $\mathbf{\Phi_V}$ with associated domains. The BJT constructed for the threat assessment problem is shown in Fig. 2. This BJT is a result of application of the algorithm presented in Appendix . The nodes in the BJT are labelled by integer numbers from 1 to 29. The leaves of the tree (the nodes labelled from 1 to 15) represent the original valuations specified by the set $\mathbf{\Phi_V}$. The remaining nodes in the BJT represent the intermediate steps of the fusion algorithm; as such they specify the order in which the valuations must be combined in order to calculate the valuation for the variable T. The vertical labels next to the nodes of the BJT denote the domains (the subsets of variables) of the nodes. The following comments provide further explanation on the construction of the BJT in Fig. 2.

- Consider the first two variables in the elimination sequence, namely IFFS and FPA. These variables are included in the domains of the nodes 3, 4, 12 and 13 whose BBAs are the first to be combined. The subtree of nodes $\{3, 4, 12, 13, 16, 17, 28\}$ represents the intermediate steps of this combination process. Node 16 represents the combination of 3 and 12, node 17 the combination of 4 and 13, and finally node 28 the combination of 16 and 17. These steps are described in (13).

$$\begin{array}{rclcclcl} m_{16} & = & m_3 & \oplus & m_{12}^{\uparrow\{NF,\ IFFS\}} & , & d(m_{16}) = \{NF,\ IFFS\} \\ m_{17} & = & m_4 & \oplus & m_{13}^{\uparrow\{NF,\ FPA\}} & , & d(m_{17}) = \{NF,\ FPA\} \\ m_{28} & = & \Big(m_{16}^{\uparrow\{NF,\ IFFS,\ FPA\}} & \oplus & m_{17}^{\uparrow\{NF,\ IFFS,\ FPA\}}\Big)^{\downarrow\{NF\}} & , & d(m_{28}) = \{NF\} \end{array} \quad (13)$$

- The BJT satisfies the Markov property defined in Sec. II-C. In fact, considering for example the subtree of nodes $\{1, 7, 15, 20, 21\}$, it can be seen that the variable C is contained in the domains of nodes 1 and 7, but also in the domain of all nodes in the path between 1 and 7, i.e. $Path(1, 7) = \{1, 7, 20, 21\}$. A BJT which does not satisfy this property cannot be a representation of the fusion algorithm. For example, let us assume that the domain of node 20 does not include C; this means that C has been eliminated during the combination of the BBAs of nodes 7 and 15. If this were true, before combining the valuations at node 20 with node 1 to produce the BBA for node 21 (the domain of 1 contains C), we should again extend the domain of node 20 to a new domain containing C. Since marginalization produces a loss of information (coarsening), which can no longer be recuperated with the extension operation (see the remark at the end of Sec. III-B), the BBA of node 21 would be incorrect, i.e. it would be different from $(m_1 \oplus m_7 \oplus m_{15})^{\downarrow\{T,HI,C,WER\}}$.

The BJT in Fig. 2 was obtained with the following variable elimination sequence: IFFS, FPA, I, C, EM, FCR, CM, PC, PT, WER, HI, NF. As it has already been explained in section II-B, finding the optimal elimination sequence is an NP-complete problem but there exist several heuristics for finding a good elimination sequence problem. The previous elimination sequence has been calculated by means of the *One Step Look Ahead - Smallest Clique, Fewest Focal sets* (OSLA-SCFF) heuristic [31, p.61]. This heuristic chooses the variable to be eliminated by minimizing the cardinality of the domain and the number of focal sets associated with the nodes of the BJT. Note that a different elimination sequence would result in a different BJT. For example, the BJT in Fig. 3 was obtained with the elimination sequence IFFS, FPA, I, EM, FCR, CM, PC, PT, WER, NF, C, HI which has been calculated by applying the *One Step Look Ahead - Fewest Fill-ins* (OSLA-FFI) heuristic [31, p.60]. Note that the final result of the application of inward propagation algorithm is independent of the elimination sequence and, thus, of the structure of the BJT. As it will be discussed in section V-E, the difference between the application of inward propagation to different BJTs is only in the computational time required to calculate the result.

B. Three extreme cases

To apply the inward propagation algorithm, the valuations of the leaves of the tree must be initialized first. The BBAs of the nodes from 1 to 7 have been already defined in Sec. IV. For the input valuations, nodes from 8 to 15, in this section we consider three "extreme" cases: (1) total ignorance; (2) high degree of threat and (3) low degree of threat. The BBAs for the input valuations in all three cases are given in Table II. For the case of the total ignorance, all input valuations are represented by vacuous BBAs. For the case of a high (low) threat, all BBAs

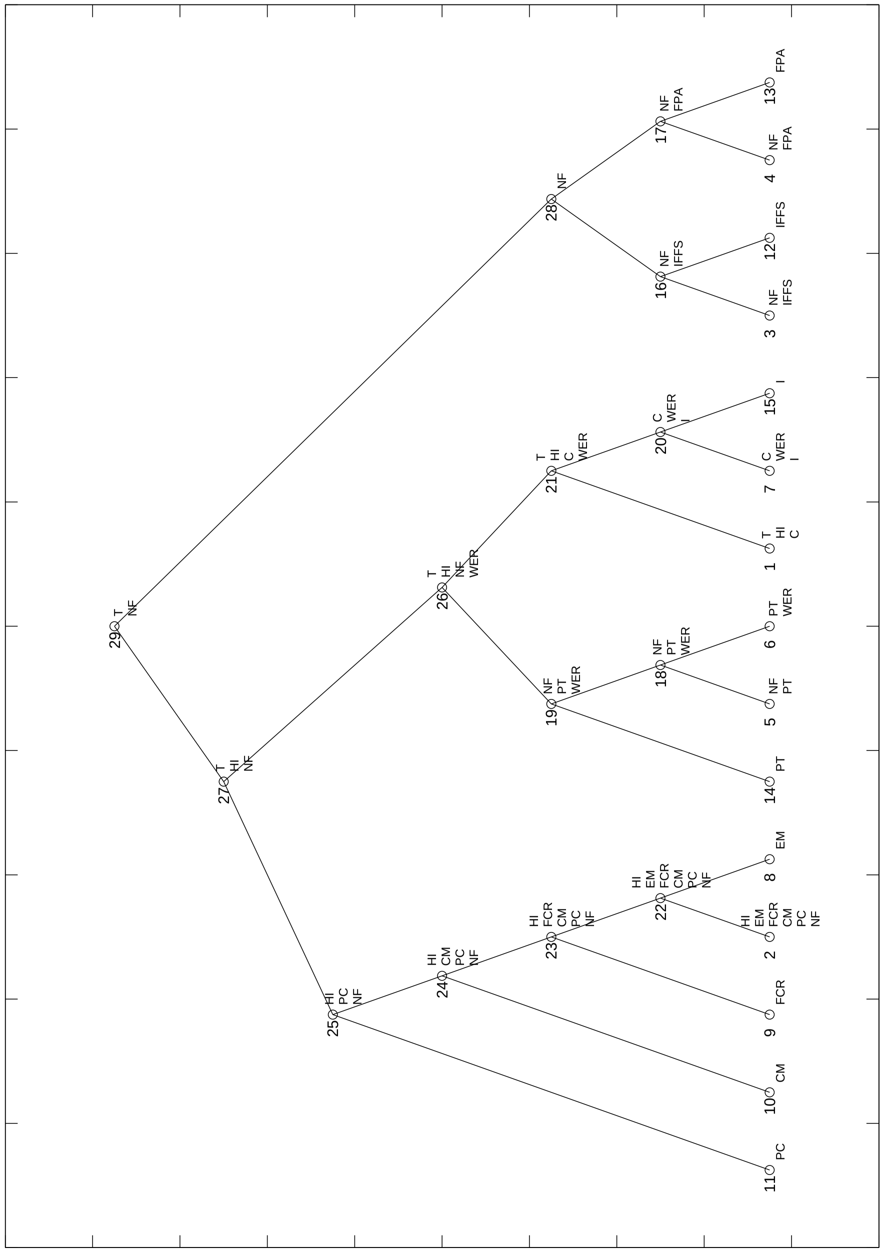

Fig. 2. Binary join tree for the threat assessment model obtained by applying the OSLA-SCFF heuristic

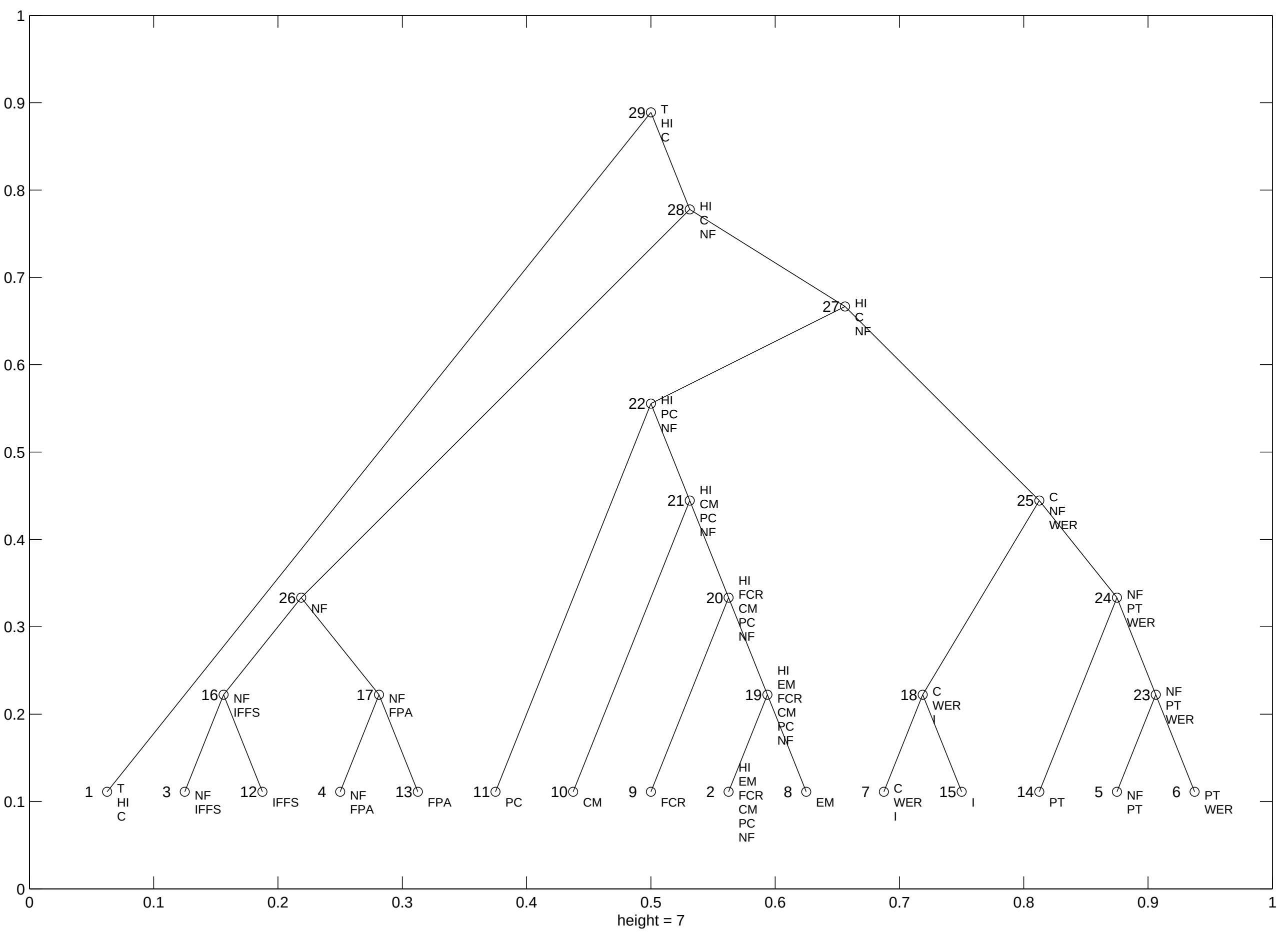

Fig. 3. Binary join tree for the threat assessment model obtained by applying the OSLA-FFI heuristic

are singletons taking high (low) threat values. Furthermore, in all three extreme cases we consider static reasoning, that is input valuations do not change with time. A dynamic case will be discussed in Sec.V-C.

TABLE II

The input belief for the no information, high degree of threat and low degree of threat cases

		no information		high threat		low threat	
BBA	domain	focal set	mass	focal set	mass	focal set	mass
m_8	EM	$\{0,1\}$	1	$\{1\}$	1	$\{0\}$	1
m_9	FCR	$\{0,1\}$	1	$\{1\}$	1	$\{0\}$	1
m_{10}	CM	$\{0,1\}$	1	$\{1\}$	1	$\{0\}$	1
m_{11}	PC	$\{0,1\}$	1	$\{1\}$	1	$\{0\}$	1
m_{12}	IFFS	$\{0,1\}$	1	$\{0\}$	1	$\{1\}$	1
m_{13}	FPA	$\{0,1\}$	1	$\{0\}$	1	$\{1\}$	1
m_{14}	PT	$\{0,1,2,3,4,5\}$	1	$\{5\}$	1	$\{0\}$	1
m_{15}	I	$\{0,1,2\}$	1	$\{2\}$	1	$\{0\}$	1

The output of *inward propagation* is the BBA of node 29, defined on domain {T,NF}. This BBA is then marginalised to domain {T} and finally transformed to the pignistic probability. Fig. 4 shows the resulting (pignistic) probability mass function (PMF) for the degrees of threat (from 0 to 10) in all three cases. From this figure it can be seen that the results are in agreement with the inputs and our intuition. When there is no information (total ignorance), the resulting BBA on domain {T} is a vacuous BBA and hence all degrees of threat have the same probability. This means that the prior valuations $m_1,\ldots,m_7$ are balanced, that is they assume that all the degrees of threat are initially equally probable. For the low and high threat cases we also obtain good results, in agreement with input valuations. Notice, however, that in the low (high) threat case the probability of the degree 0 (10) is less than 1.0. This is due to the intrinsic uncertainty in the prior valuations m_1 to m_7 (representing expert knowledge).

C. Dynamic reasoning example

In a realistic air-to-air engagement scenario the input valuations will change over time as the new pieces of evidence (from surveillance sensors and other external sources) about the intruder become available. As a result, whenever an input valuation is modified, the degree of threat is supposed to change. In our evidential network initially we set all input valuations to be vacuous BBAs, representing the initial state of ignorance. Then, every time an input valuation is changed, the network re-computes the valuations of all the nodes of the BJT along the

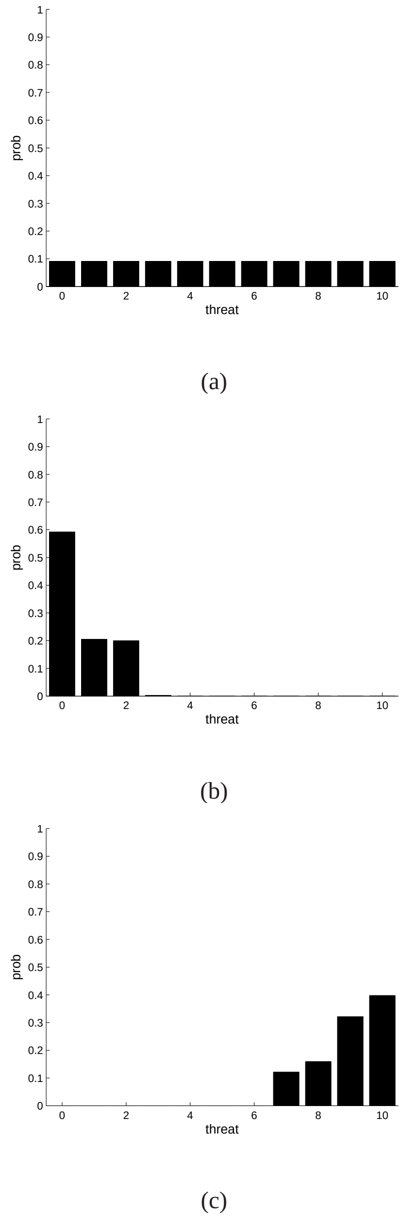

Fig. 4. Pignistic probability mass function for variable T (threat) in extreme cases: (a) total ignorance case; (b) low threat case; (c) high threat case

affected path of the tree. For example if the mass m_{15} changes, only the masses of the nodes 20, 21, 26, 27 and 29 must be re-computed (see Fig. 2).

Consider an example of a sequence of incoming evidence shown in Table III. At time t_1 we feed into the network the current state of the political climate (PC) represented by BBA m_{11}. For argument's sake, let this BBA reflect a state of political tension in the region, so that the belief mass given to the state of war is 0.7, while the remaining 0.3 is assigned to ignorance. Then at time t_2 some evidence about the EM variable becomes available; it appears that the target is performing an evasive manoeuvre, so we assign a belief mass of 0.8 to true and 0.2 to the state of ignorance. Each time a new piece of evidence is available, the situation becomes more informative (less uncertain) which is reflected by the pignistic PMF of threat, shown in Fig 5. Note how this PMF evolves from being totaly uninformative at time t_0 to becoming concentrated (”peaky”) at time t_9. At this last time instant the degree of threat with the highest probability is 8 (on the scale from 0 to 10).

TABLE III

The sequence of incoming evidence driving the evidential network

Time	BBA	domain	focal set	mass
t_1	m_{11}	PC	$\{1\}$	0.7
			$\{0,1\}$	0.3
t_2	m_8	EM	$\{1\}$	0.8
			$\{0,1\}$	0.2
t_3	m_{15}	I	$\{0,1\}$	0.7
			$\{0,1,2\}$	0.3
t_4	m_{13}	FPA	$\{1\}$	0.9
			$\{0,1\}$	0.1
t_5	m_{15}	I	$\{1\}$	0.8
			{0,1,2}	0.2
t_6	m_{14}	PT	$\{2\}$	0.6
			$\{3\}$	0.3
			$\{4\}$	0.1
t_7	m_{12}	IFFS	$\{0\}$	0.9
			$\{0,1\}$	0.1
t_8	m_{10}	CM	$\{1\}$	0.9
			$\{0,1\}$	0.1
t_9	m_9	FCR	$\{1\}$	0.8
			$\{0,1\}$	0.2

D. Sensitivity analysis

Sensitivity analysis studies the effect of the changes in the input valuations on the valuation of the output (decision) variable. In this way, sensitivity analysis helps us to identify which inputs are more influential on decision making and how they affect the decision process. Inward propagation on a BJT is used for performing sensitivity analysis in a VBS, because it can rapidly re-compute the valuation of the decision variable when a valuation of one of

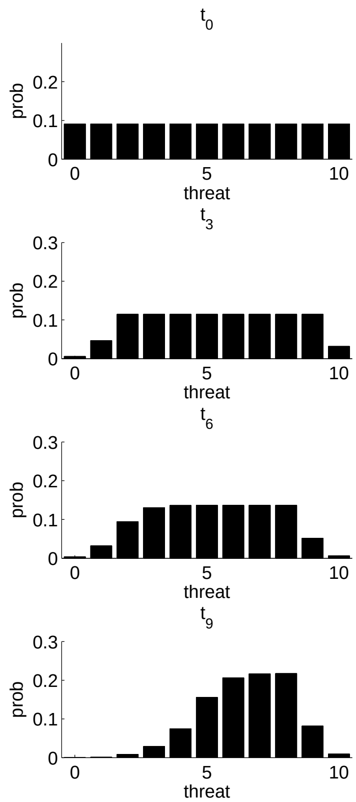

Fig. 5. Pignistic probability mass function for variable T (threat) in a dynamic situation (from time t_0 to t_9)

the leaf nodes in the BJT changes. As previously noted, when a change happens, we simply need to propagate the valuations inwards from the modified node to the root of the BJT (see Stage 2 in Appendix). The following algorithm describes the steps for performing a sensitivity analysis in a VBS.

1) Change the valuation of the input variable x;
2) Execute Stage 2 of inward propagation with updated input $UI = \{x\}$ and calculate the valuation for the decision variable;
3) Evaluate the effect of the change on the valuation of the decision variable.

For the dynamic reasoning problem described in the previous section, we investigate how the change of the input BBAs on three variables (EM, FCR and FPA) affects the BBA of the decision variable T. Table IV presents the results of the sensitivity analysis for this case. Input BBAs on EM, FCR and FPA take two contrasting values: either all mass is assigned to *true* or to *false*. Comparing the resulting pignistic PMFs of the threat variable for the considered cases, it can be seen that the most influential variable is FCR; when the BBA of FCR goes from

$m(\{1\}) = 1$ to $m(\{0\}) = 1$, the pignistic probability of threat T changes more than in the other two cases. This observation is not surprising, since FCR is weighted higher than the other variables in contributing to the HI, see (11).

TABLE IV

SENSITIVITY ANALYSIS RESULTS

		Threat - Pignistic Probability										
var	mass	0	1	2	3	4	5	6	7	8	9	10
EM	m_8({T})=1	0	0	0.06	0.02	0.06	0.15	0.21	0.23	0.23	0.08	0.01
EM	m_8({F})=1	0	0.01	0.02	0.06	0.15	0.21	0.22	0.23	0.08	0.01	0
FCR	m_9({T})=1	0	0	0	0.01	0.06	0.16	0.22	0.23	0.23	0.09	0.01
FCR	m_9({F})=1	0.01	0.06	0.16	0.22	0.23	0.23	0.09	0.01	0	0	0
FPA	m_{13}({T})=1	0	0	0.01	0.03	0.08	0.16	0.21	0.22	0.22	0.06	0
FPA	m_{13}({F})=1	0	0	0.01	0.026	0.07	0.14	0.19	0.20	0.20	0.15	0.02

E. Computational complexity

As we explained earlier, the reasoning for threat assessment can be carried out without using the VBS framework, that is by directly computing the joint belief on the domain $\mathbf{\Theta_V}$ followed by marginalisation of the resulting belief to the domain of T. The advantage of using the VBS framework is the computational efficiency. From Table I it can be seen that the number of configurations in the joint frame of $\mathbf{\Theta_V}$ is 2661120 (i.e. the Cartesian product of the frames of the single variables). This is a huge number compared with the number of elements of the maximum domains in the two BJTs shown in Figs.2 and 3. In the BJT obtained by applying the OSLA-SCFF heuristic (Fig.2), the number of elements of the maximum domain is only 1155 (for node 21). This number is even lower for the BJT obtained by applying the OSLA-FFI heuristic (Fig.3). In this case, the maximum domain has only 385 elements (for nodes 1 and 29).

Since the joint belief for $\mathbf{\Theta_V}$ is defined on the power set of $\mathbf{\Theta_V}$, for computing the joint belief we need to calculate, in the worst case, the masses for all the $2^{2661120}$ elements of the power set. When we attempted this "brute force" approach for threat assessment on the joint domain, our computer could not complete this task after 48 hours of processing. By contrast, using the VBS framework, the threat assessment was carried out on the same computer in just 5 seconds for the BJT obtained by applying the OSLA-SCFF heuristic and 3 seconds for the BJT obtained by applying the OSLA-FFI heuristic.

We point out that although for the adopted threat assessment model the OSLA-FFI heuristic allows to compute the solution faster than OSLA-SCFF, in general this may not be true: the computational complexity and the effectiveness of the heuristic for sequence elimination depend strongly on the structure of the problem. The general rule is: the more complex are the interdependencies among the variables, the smaller is the advantage in using the VBS. Real complex reasoning systems, with hundreds or even thousands of variables, are usually characterised by very localised structures. As the computational complexity grows exponentially with the domain size, the VBS framework can solve problems that otherwise would be computationally intractable.

In addition to the structure of the network, the computational complexity of an evidential network depends on the cores, i.e. the sets of focal elements of the belief functions to be combined. Note that the number of focal sets is also problem-dependent.

Finally another computational advantage of the VBS, as discussed earlier, is the possibility of re-computing the valuation of the decision variable when one or more inputs change. In this case the inward propagation re-computes only the valuations of those nodes of the BJT that belong to the path connecting the leaves with changed valuations to the root of the BJT.

VI. Conclusion

The paper has presented an automatic data fusion system for determining threat assessment in the context of air defence. Based on expert knowledge, the threat has been modelled by a network of entities (representing target behaviours or critical events) and their mutual relationships. The uncertain and imprecise prior information, expert knowledge and incoming evidence supplied by the surveillance sensors and other sources of information have been expressed as belief functions. The determination of threat assessment has been performed within the framework of valuation-based systems using local computations on the binary join tree via the inward propagation algorithm. The result is an inference engine capable of the timely and accurate processing of vast amounts of data in support of a commander's decision making. One of the major contributions of this paper has been to endow the inference engine with the capacity to manage efficiently time varying information, which is typically encountered in situation and threat assessment problems.

Our plans for future work are twofold. In terms of threat assessment, we will consider the refinement of the threat model to capture the threat assessment process more realistically and to cater for networks with more entities and larger frames (for example, the frame of platform types can have hundreds of elements). However, since the inference engine that we have developed is independent of the threat assessment application, it can also be applied in other domains provided that the variables for the given problem, the relationships that hold between them and the values they may assume based on prior information, sensor data and expert knowledge, can be identified. As such, we also plan to investigate the suitability of the approach for other defence and intelligence problems such as combat identification and possibly border protection and situation awareness for homeland security.

VII. Acknowledgement

The authors would like to thank the Department of Electrical and Electronic Engineering of The University of Melbourne for hosting A. Benavoli while the work on this project has been carried out. The authors would like also to thank Dr Norbert Lehmann and anonymous reviewers for their valuable comments.

Appendix

Let $\mathbf{\Psi} = \{\varphi_1, \varphi_2, \dots, \varphi_r\}$ be a given set of valuations and $\mathbf{D}^o \subseteq \mathbf{V}$, with $\mathbf{V} = d(\varphi_1) \cup d(\varphi_2) \cup \cdots \cup d(\varphi_r)$, the domain of interest. Let us introduce the following notation [30]:

$$\begin{aligned} L(n):&\quad \text{left child of node } n, \text{ or } nil \text{ if } n \text{ is a leaf;}\\ R(n):&\quad \text{right child of node } n, \text{ or } nil \text{ if } n \text{ is a leaf;}\\ F(n):&\quad \text{parent of node } n, \text{ or } nil \text{ if } n \text{ is the root of the tree;}\\ d(n):&\quad \text{domain of the valuation for the node } n;\\ root:&\quad \text{root of the BJT.} \end{aligned} \tag{14}$$

The algorithm for constructing a BJT is as follows [30].

Initialization:
Define the initial set of node $\mathbf{N}_\psi = \{n_1, n_2, \dots, n_r\}$ with $d(n_i) = d(\varphi_i)$, $L(n_i) = nil$, $R(n_i) = nil$ and $F(n_i) = nil$.
Fix the set of variables to be eliminated $\mathbf{\Delta} = \mathbf{V} - \mathbf{D}^0$.
function CONSTRUCT A BJT($\mathbf{N}_\psi$,$\mathbf{\Delta}$)
    $\mathbf{N} = \emptyset$; $\mathbf{\Delta}^c = \emptyset$; $root = nil$;
    repeat
        if $\mathbf{\Delta} = \emptyset$ **then**
            $\mathbf{N}_x = \mathbf{N}_\psi$;
        else
            select the next variable to be eliminated, $x \in \mathbf{\Delta}$, using some heuristic;
            $\mathbf{N}_x = \{n \in \mathbf{N}_\psi : x \in d(n)\}$;
        end if
        while $|N_x| > 1$ **do** $\triangleright$ while the cardinality of N_x is greater than 1
            generate a new node n with $F(n) = nil$;
            select distinct $n_1, n_2 \in \mathbf{N}_x$;
            $F(n_1) = n$; $F(n_2) = n$;
            $L(n) = n_1$; $R(n) = n_2$;

18: $d(n) = (d(n_1) \cup d(n_2)) - \mathbf{\Delta}^c$;
19: $\mathbf{N}_x = (\mathbf{N}_x \backslash \{n_1, n_2\}) \cup \{n\}$;
20: $\mathbf{N} = \mathbf{N} \cup \{n_1, n_2\}$;
21: **end while**
22: **if** $\mathbf{\Delta} = \emptyset$ **then**
23: $root = n$;
24: **else**
25: $\mathbf{\Delta} = \mathbf{\Delta} \backslash \{x\}$; $\mathbf{\Delta}^c = \mathbf{\Delta}^c \cup \{x\}$;
26: $\mathbf{N}_\psi = \{n \in \mathbf{N}_\psi : x \notin d(n)\} \cup \{n\}$;
27: **end if**
28: **until** $root \neq nil$
29: $\mathbf{N} = \mathbf{N} \cup \{n\}$;
30: **return** N
31: **end function**

The tree resulting from this procedure is a BJT with $2r-1$ nodes, $\mathbf{N} = \{n_1, n_2, \ldots, n_{2r-1}\}$, such that $\mathbf{D}^o \subseteq d(root)$. The only degree of freedom in the BJT construction algorithm is the order in which the variables are eliminated (Step 10).

The objective of the inward propagation algorithm is to compute the valuations for the variables of interest. Consider again the set of valuations $\mathbf{\Psi} = \{\varphi_1, \varphi_2, \ldots, \varphi_r\}$, the domain of interest $\mathbf{D}^o$ and the set $\mathbf{N} = \{n_1, n_2, \ldots, n_{2r-1}\}$ of nodes of the BJT constructed by the algorithm given in appendix . The inward propagation is performed in two stages. In Stage 1, which is executed only once, the valuations are propagated from the leaves towards the root of the BJT [30]. Stage 2 is performed every time the valuations of one or more leaves of the BJT change. In this case, inward propagation re-computes only the valuations of those nodes of the BJT that belong to the path connecting the leaves with changed valuations to the root of the BJT. The steps of the algorithm for the inward propagation are as follows.

Initialization:
Initialize $Leaves = \{n \in \mathbf{N} : n \text{ is a leaf}\}$.
if stage=1 **then**
$\varphi_s = nil$;
$UI = Leaves$;
else
$UI = \{n \in Leaves : \text{ valuation is changed w.r.t. the previous time}\}$;
end if
function INWARD PROPAGATION($Leaves$, $\mathbf{N}$, $stage$, UI, φ, φ_s)
if stage=1 **then** ▷ It is the first time that inward propagation is performed
Set $next = \{n \in \mathbf{N} : L(n) \in Leaves$ and $R(n) \in Leaves\}$.
for $n \in Leaves$ **do**
$\varphi_s(n) = \varphi(n)^{\downarrow d(F(n))}$;
end for
else ▷ inward propagation has been already performed at least one time
$next = \emptyset$;
for $n \in \mathbf{N}$ **do**
if $L(n) \in UI$ or $R(n) \in UI$ **then**

Set $next = next \cup n$.
end if
end for
end if

$visitN = \emptyset$ ▷ indicates the nodes visited during the inward propagation
while $|next| > 0$ **do** ▷ while $next$ is not empty
extract an element n from $next$;
$visitN = visitN \cup n$; $next = next - n$;
$\varphi(n) = \varphi_s(L(n)) \oplus \varphi_s(R(n))$;
$\varphi_s(n) = \varphi(n)^{\downarrow d(F(n))}$;
if $n \neq root$ **then**
$DomF = d(F(n))$;
else
$DomF = d(d_v)$
end if
$\varphi_s(n) = \varphi(n)^{\downarrow DomF}$;
if stage=1 **then** ▷ It is the first time that inward propagation is performed
for $n \in \mathbf{N}$ **do**
if $(n \notin Leaves)$ and $(n \notin next)$ and $(n \notin visitN)$ **then**
if ($L(n) \in Leaves$ or $L(n) \in visitN$) and ($R(n) \in Leaves$ or $R(n) \in visitN$) **then**
$next = next \cup n$;
end if
end if
end for
else
for $n \in \mathbf{N}$ **do**
if $(n \notin UI)$ and $(n \notin next)$ and $(n \notin visitN)$ **then**
if $(L(n) \in visitN)$ or $(R(n) \in visitN)$ **then**
$next = next \cup n$;
end if
end if
end for
end if
end while
return φ, φ_s ▷ $\varphi_s(root)$ is the valuation for the decision variables
end function

References

[1] E. Waltz and J. Llinas. *Multisensor Data Fusion*. Artech house, 1990.

[2] A. Kott, M. Pollack, and B. Krogh. The situation assessment problem: Toward a research agenda. In *Proc. DARPA-JFACC Symposium on Advances in Enterprise Control*, San Diego, CA, Nov. 1999.

[3] J. Pearl. *Probabilitic reasoning in intelligent systems*. Morgan Kaufmann Publishers, San Francisco, 1988.

[4] S. Das, R. Grey, and P. Gonsalves. Situation assessment via Bayesian belief networks. In *Proc. Fifth Int. Conf. Information Fusion*, Annapolis, MD, July 2002.

[5] P. Bladon, R. J. Hall, and W. A. Wright. Situation assessmennt using graphical models. In *Proc. Fifth Int. Conf. Information Fusion*, Annapolis, MD, July 2002.

[6] F. Mirmoeini and V. Krishnamurthy. Reconfigurable Bayesian networks for adaptive situation assessment in battlespace. In *Proc. IEEE Conf. Networking, Sensing and Control*, pages 810 – 815, March 2005.

[7] N. Okello and G. Thomas. Threat assessment using Bayesian networks. In *Proc. Sixth Int. Conf. Information Fusion*, Cairns, Australia, July 2003.

[8] X.T. Nguyen and P. Heuer. Automated intent assessment simulation environment. In *Proc. Information, Decision and Control*, Adelaide, Australia, Feb. 2002.

[9] K. D. Lee and J. Llinas. Hybrid model for intent estimation. In *Proc. Sixth Int. Conf. Information Fusion*, Cairns, Australia, July 2003.

[10] J. D. Lowrance, T. D. Garvey, and T. M. Strat. A framework for evidential reasoning systems. In *Proc. 5th Natl. Conf. on AI (AAAI-86)*, pages 896–901, Philadelphia, 1986.

[11] G. Shafer, P.P. Shenoy, and K. Mellouli. Propagating belief functions in qualitative Markov trees. *Int. Journal of Approximate Reasoning*, 1(4):349–400, 1987.

[12] L. Zadeh. Fuzzy sets as a basis for a theory of possibility. *Fuzzy Sets and Systems*, 1:3–28, 1978.

[13] L. Zadeh. A theory of approximate reasoning. In J. E. Ayes, D. Mitchie, and L. I. Mikulich, editors, *Machine Intelligence 9*, pages 149–194. Ellis Horwood, Chichester, U.K., 1979.

[14] D. Dubois and H. Prade. *Possibility theory: An approach to computerised processing of uncertainty*. Plenum Pub., 1988.

[15] G. Shafer. *A mathematical theory of evidence*. Princeton University Press, 1976.

[16] Smets. P. Data fusion in the transferable belief model. In *Proc. 3rd Int. Conf. Information Fusion*, pages PS21–33, Paris, France, 2000.

[17] P.P. Shenoy. A valuation-based language for expert systems. *Int. Journal of Approximate Reasoning*, 3(5):383–411, 1989.

[18] P. P. Shenoy. Valuation based systems: A framework for managing uncertainty in expert systems. In L. A. Zadeh and J. Kacprzyk, editors, *Fuzzy Logic and the Management of Uncertainty*, chapter 4, pages 83–104. Wiley, New York, 1992.

[19] R. Haenni. Modeling uncertainty with propositional assumption-based systems. In *Lecture Notes In Computer Science: Applications of Uncertainty Formalisms*, volume 1455, pages 446–470. Springer-Verlag, London, UK, 1998.

[20] V. Lepar and P. P. Shenoy. A comparison of lauritzen-spiegelhalter, hugin and shenoy-shafer architectures for computing marginals of probability distributions. In *Proceedings of the 14th Conference on Uncertainty in Artificial Intelligence (UAI-98)*, pages 328–337, 1998.

[21] S. L. Lauritzen and D. J. Spiegelhalter. Local computations with probabilities on graphical structures and their application to expert systems. In *Journal of Royal Statistical Society*, volume 50, pages 157–224. 1988.

[22] K. G. Olesen F. V. Jensen and S. K. Andersen. An algebra of bayesian belief universes for knowledge-based systems. In *Network*, volume 20, pages 637–659. 1990.

[23] S. L. Lauritzen F. V. Jensen and K. G. Olesen. Bayesian updating in causal probabilistic networks by local computation. In *Computational Statistics Quarterly*, volume 4, pages 269–282. 1990.

[24] S. L. Lauritzen and F. V. Jensen. Local computation with valuations from a commutative semigroup. In *Annals of Mathematics and Artificial Intelligence*, volume 21, pages 51–69. 1997.

[25] P. P. Shenoy and G. Shafer. Axioms for probability and belief-function propagation. In J. Pearl G. Shafer, editor, *Readings in uncertain reasoning*, pages 575–610. San Mateo, CA: Morgan Kaufmann, 1990.

[26] P. P. Shenoy. Binary join trees for computing marginals in the shenoy-shafer architecture. In *Int. Journal of Approximate Reasoning*, volume 17, pages 239–263, 1997.

[27] G. J. Klir and M. J. Wierman. *Uncertainty-based information: Elements of generalized information theory*. Physica-Verlag, New York, 2nd edition, 1999.

[28] B. Ristic and P. Smets. Target identification using belief functions and implication rules. *IEEE Trans. Aerospace and Electronic Systems*, 41(3):1097–1103, July 2005.

[29] R. G. Almond. *Graphical Belief Modeling*. Chapman and Hall, 1995.

[30] R. Haenni. Ordered valuation algebras: a generic framework for approximate inference. *Int. Journal of Approximate Reasoning*, 37:1–41, 2004.

[31] N. Lehmann. *Argumentation Systems and Belief Functions*. Tech. report 01-30. Department of Informatics, University of Fribourg, 2001.

[32] R. Haenni and N. Lehmann. Resource-bounded and anytime approximation of belief function computations. *Int. Journal of Approximate Reasoning*, 32(1-2):103–154, 2002.

[33] P. Smets. Belief functions: the disjunctive rule of combination and the generalized Bayesian theorem. *Int. Journal of Approximate Reasoning*, 9:1–35, 1993.

[34] P. Smets. Decision making in the TBM: The necessity of the pignistic transformation. *Int. Journal of Approximate Reasoning*, 38:133–147, 2005.

[35] M. Oxenham. Enhancing situation awareness for air defence via automated threat analysis. In *Proc. Sixth Int. Conf. Information Fusion*, Cairns, Australia, July 2003.

[36] Fighter planes. Web: http://www.fighter-planes.com/.