Bayesian Decision Support for Complex Systems with Many Distributed Experts

Manuele Leonelli, James Q. Smith

Department of Statistics, The University of Warwick, CV47AL Coventry, UK

Abstract

Complex decision support systems often consist of component modules which, encoding the judgments of panels of domain experts, describe a particular sub-domain of the overall system. Ideally these disparate modules need to be pasted together to provide a comprehensive picture of the whole process. The challenge of building such an integrated system is that, whilst the qualitative features are common knowledge to all, the explicit forecasts and their associated uncertainties are expressed only locally. The structure of the integrated system therefore needs to facilitate the coherent piecing together of these separate evaluations. If such a system is not available there is a serious danger that this might drive to incoherent and so indefensible decision making. In this paper we develop a graphically based framework which embeds a set of conditions that, if satisfied in a given context, are sufficient to ensure the composite system is truly coherent. Furthermore, we develop new message passing algorithms that enable the uncertainties within each module to be fully accounted for in the evaluation of expected utility scores of this composite system.

Keywords: Bayesian decision theory; combination of expert judgment; decision support systems; graphical models; uncertainty handling;

1. Introduction

Nowadays decision centers are often required to make choices in complex and evolving environments, described through multiple and interdependent processes with many associated measurements. The objective of a real time decision making center is to agree to a sequence of efficacious countermeasures. To achieve this it is usually necessary to integrate the opinions and information from an often diverse set of stakeholders, who articulate several competing sub-objectives and knowledge over different domains of expertise. A collection of decision support systems (DSSs) can enhance such an integration, not only ensuring that all relevant evidence systematically informs policy making, but also encouraging the decision center to exhibit an underlying consistency across all its components and to address the problem as a whole.

One domain of application exhibiting such complexity is emergency management, especially to guide the choice of countermeasures after a nuclear accident. Here the decision center needs to address the diverse deleterious outworkings of an accidental release of contaminants, taking into account, for example, the effects on health, the political implications and the environmental consequences of that accident. Each of these issues is likely to be informed by a different panel of domain experts, who are the ones best able to articulate appropriate forecasts, their uncertainties and the evaluation of specific consequences arising directly from these.

Early support systems often consisted of a suite of different *component DSSs*, or *modules*, designed by an appropriate panel of experts. These used a variety of deterministic and stochastic methodologies to guide the estimation and the forecasting of the various quantities relevant to the domain under study (Ehrhardt et al. 1993, Ehrhardt 1997). Fully probabilistic component modules then began to be developed to communicate both the relevant panel's forecasts and their associated uncertainties. For nuclear emergency management the first such probabilistic DSSs tended to focus on the initial consequences of the accident. For example French et al. (1995) developed a belief network for the workings of the source term, while Smith and Papamichail (1999) presented a graphical model to study the dispersion of the contamination. In more recent years, more sophisticated probabilistic modules modeling both early and later stages of the accident have been introduced (Richter et al. 2002, Zheng et al. 2009, De and Faria 2011).

These current component DSSs are designed to function independently, albeit with inputs provided by the outputs of other modules. Under such direct and indirect guidance of the relevant panels, each delivers judgments about the processed modules and evaluate possible outcomes, for potential use by other components. For example, the forecasts of the dispersion into the atmosphere of a radioactive contaminant depend on the concentrations of toxins in emission from the power plant. The estimates of these concentrations, computed by their associated panel, provide the outputs of a source term module, which are then used as inputs to a dispersion DSS.

Such a suite of DSSs are obviously invaluable to the center. Nevertheless the challenge is to somehow join together the information from these systems in a *coherent* way. For example, an unaided center might be tempted to simply plug in point estimates of the inputs necessary for a receiving module and to ignore the delivered uncertainties associated with the values of these inputs. However, it has been known for some time that, even in very simple scenarios, ignoring these uncertainties can lead a decision center to choose the wrong course of action (see Leonelli and Smith (2013b) for a simple example of this). This is because expected utility scores for competing suites of countermeasures often formally depends on these measures of uncertainty.

When all the modules are fully probabilistic, then, under certain assumptions, we show later in this paper that formulae can be calculated to give the formal expected utility scores so that these potential dysfunctionalities can be avoided. Furthermore, message passing algorithms can be defined to enable the center to evaluate the efficacy of different policies efficiently as well as faithfully. In this paper we also discuss when and how this can be achieved. The integrating system so defined ensures that any decision center is properly supported. Policy choices drawing together disparate judgments and objectives are then guided into providing a *consistent* analysis of the complex composite problem.

Here, in order to provide this formal framework, we envisage a real or virtual system manager, or *supraBayesian (SB)*, who is responsible, with the decision center, for the aggregation of the local panels' judgments. A successful combination of the beliefs of the panels needs to give sufficient information for the SB to calculate the expected utility scores associated to different policies. Long ago French (1997) gave a vision for addressing this class of problems using Bayesian decision analysis. However until recently, due to computational and methodological constraints, it has not been feasible to actually implement this vision: now it is. Henceforth, we use the phrase **integrating decision support system (IDSS)** to denote the unifying and integrating framework around which the SB combines component DSSs into a single entity. When sufficient conditions exist for local beliefs of the individual panels to be formally combined into a coherent whole, we say that the system *distributed*.

We envisage that the group of panels can identify a common agreed, overarching structure during *decision conferences*, where a facilitator guides the panels' discussion and exploration of the implications of the possible models (French et al. 2009). During these meetings they also discuss any other necessary probabilistic assumptions needed to be made as, for example, global independence of the various components of the parameter set (Cowell et al. 1999). These behavioral methodologies are well documented and have often been successfully applied in practice (Ackermann 1996). The IDSS is then built around this agreed qualitative framework.

In Section 2 below we discuss technical conditions which are sufficient to ensure we can build a distributed IDSS. There are several advantages that derive from structuring a problem so that the ensuing support is distributed. First, most importantly, because the responsibility for each aspect of the analysis can be devolved to *appropriate* panels of experts, these are then more likely to deliver better calibrated judgments. The whole system might therefore be expected to be more robust to the misspecification of beliefs (Cooke 1991). On the other hand, if the system needs to be changed in the light of unexpected developments or unplanned consequences, under suitable conditions, the management of these new developments need only be addressed *locally* by the relevant expert panels. These simply adapt their individual forecasts and the inputs in the light of the new scenario they face. These adjustments can then be folded back into the system to provide a revised output of the relevant modules for other panels to use for their inputs.

Second, the output of a distributed IDSS can produce answers to queries by particular users about the premises on which it is based and the calculations of its outputs, by directing the query to the relevant panels. If the IDSS did not exhibit this distributivity property, then such devolution may not be possible and so any support would be much less transparent. Third, distributivity ensures that different users can be given the option of choosing different modules to model the various components of the problem. For example, in nuclear emergency management different countries often prefer to predict the spread of the contamination using the diffusion model preferred by one of their national agencies.

In the applications we have in mind the input data and the underlying processes supported by an IDSS are intrinsically dynamic and unfold in time. For this reason it is necessary to consider classes of models that can accommodate a sequential update of the model parameters and, consequently, of the relevant probabilities. This is in particular the case for emergency management, where new observations are constantly gathered throughout the crisis and decision centers need to make decisions at each time new information is available (Leonelli and Smith 2013a). For simplicity here, although this is not strictly necessary, we assume that each component module is a dynamic probabilistic model of a type we define later over the inputs and the outputs of the component DSSs within the IDSS. In addition we suppose that there is an agreed overarching framework of conditional independences - coded by a graphical model - common knowledge both to the user of the IDSS and all the panel members.

Having determined conditions under which an IDSS can be updated dynamically in a modular fashion, in Section 3 we develop message passing algorithms over the network of component DSSs between the panels and the SB. These enable the SB to compute the expected utility scores of potential policies. These also let calculations to be devolved to the relevant panels and hence define the operations of the given distributed system. Our algorithms work similarly to the many others designed for the single agent propagation of probabilities and expected utilities through, respectively, Bayesian networks (BNs) and influence diagrams (IDs) (see for example, Cowell et al. 1999, Lauritzen 1992, Jensen and Nielsen 2009). The subtlety here is to derive conditions to ensure that all calculations can be decomposed into local sub-calculations that can be made by individual panels. We are able to demonstrate that it is surprisingly simple to calculate message passing algorithms based on these local computations using standard backward induction, albeit in this novel and potentially very complex setting. Distributivity conditions thus imply that the system is able to quickly produce forecasts and expected utilities, enabling users to interrogate the IDSS in real-time. We illustrate these processes in Section 4.

As with its probabilistic counterpart, for a full expected utility decision support to be possible, a user needs to specify her utility function. It is often reasonable and almost always assumed in practice that a utility function lies in some family, which captures some form of preferential independence (Fishburn 1967, Keeney and Raiffa 1993). This then usually implies some factorization of a user's utility function. Here we assume that this factorization, jointly agreed by the panels of experts, relevant stakeholders and potential users, lies within a family we define in Section 2, customized to the needs of the network of expert systems. In particular we suppose here that each panel, being the most informed about the different *consequences* associated with the outcomes of the attributes it oversees, is separately responsible for the elicitation of a marginal utility function over those attributes.

2. The Integrating Decision Support System

To formally define the network of component DSSs constituting an IDSS we first need to introduce a theoretical framework based on a set of structural assumptions that can guarantee its distributivity. Our exposition broadly follows the structure of a Bayesian group decision analysis, as described in Figure 1. However in addition we also assume the following. First, all the panels agree on a overarching (dynamic) graphical statistical model representing the relationships existing between the quantities they agree to include in the analysis. Second, they are able to jointly determine a decision space describing the available actions that can be taken after the observation of a specific subset of the variables. Third, we assume they share a utility factorization which lies within a customized family of utilities. All these assumptions are defined formally below.

Thus, let $\{\boldsymbol{Y}_t\}_{t=1,\dots,T}$ be a multivariate time series with a finite horizon T, partitioned into n vector time series $\{\boldsymbol{Y}_t(i)\}_{t=1,\dots,T}$, $i = 1,\dots,n$, such that

$$\boldsymbol{Y}_t^{\mathrm{T}} = (\boldsymbol{Y}_t(1)^{\mathrm{T}},\dots,\boldsymbol{Y}_t(n)^{\mathrm{T}}), \qquad \boldsymbol{Y}^t = (\boldsymbol{Y}_1^{\mathrm{T}},\dots,\boldsymbol{Y}_t^{\mathrm{T}})^{\mathrm{T}}, \qquad \boldsymbol{Y}^t(i) = (\boldsymbol{Y}_1(i)^{\mathrm{T}},\dots,\boldsymbol{Y}_t(i)^{\mathrm{T}})^{\mathrm{T}}.$$

A random vector $\boldsymbol{Y}_t$ takes values in $\mathcal{Y} = \times_{i=1}^n \mathcal{Y}(i)$, where $\mathcal{Y}(i)$ is the space associated to $\boldsymbol{Y}_t(i)$. For each $t = 1,\dots,T-1$, the vector $(\boldsymbol{Y}_t^T, \boldsymbol{Y}_{t+1}^T)$ takes values in $\mathcal{Y}\times\mathcal{Y}$.

Lower case letters denotes generic instantiations of random vectors, i.e. $\boldsymbol{y}_t$, $\boldsymbol{y}_t(i)$, $\boldsymbol{y}^t$ and $\boldsymbol{y}^t(i)$ are realizations of $\boldsymbol{Y}_t$, $\boldsymbol{Y}_t(i)$, $\boldsymbol{Y}^t$ and $\boldsymbol{Y}^t(i)$ respectively. Each individual time series $\{\boldsymbol{Y}_t(i)\}_{t=1,\dots,T}$ is overseen by panel of experts, G_i, and includes all the variables associated to the i-th DSS. Let

$$\boldsymbol{Z}_t(i) \subseteq \boldsymbol{Y}_t^{i-1} = (\boldsymbol{Y}_t(1),\dots,\boldsymbol{Y}_t(i-1)), \qquad \boldsymbol{Z}^t(i) = (\boldsymbol{Z}_1(i),\dots,\boldsymbol{Z}_t(i)),$$

and suppose all $\boldsymbol{Z}_1(i),\dots,\boldsymbol{Z}_T(i)$, $i = 1,\dots,n$, include subvectors whose indices do not change between time points. Thus for example, if $\boldsymbol{Z}_t(3) = (\boldsymbol{Y}_t(1),\boldsymbol{Y}_t(2))$, then $\boldsymbol{Z}_{t+1}(3) = (\boldsymbol{Y}_{t+1}(1),\boldsymbol{Y}_{t+1}(2))$. Let also $\boldsymbol{Q}_t(i) = \boldsymbol{Y}_t^{i-1} \setminus \boldsymbol{Z}_t(i)$ and $\boldsymbol{Q}^t(i) = (\boldsymbol{Q}_1(i),\dots,\boldsymbol{Q}_t(i))$.

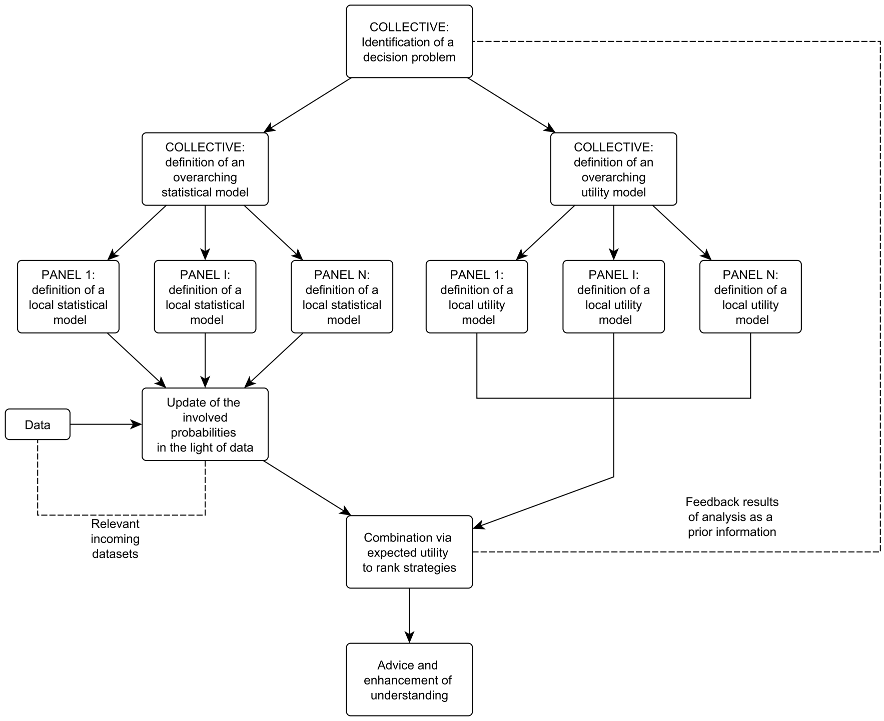

Figure 1: The structure of a Bayesian group decision analysis for an IDSS, generalizing French (1997).

Next, let UT denote the index set of the time series $\{\boldsymbol{Y}_t(i)\}_{t=1,\dots,T}$, $i = 1,\dots,n$, over which a utility function u is evaluated. Time series with an index in UT are the attributes of the utility function. Finally, the term *collective*, represented by the SB, will be used for the group. This typically consists of representatives of each panel, relevant stakeholders, potential users and interested parties of the support system. We assume here that the collective is jointly responsible for the definition of the necessary overarching probabilistic, preferential and decision structures.

2.1. The integrating system

2.1.1. The probabilistic integrating structure.

The overall statistical model the collective needs to agree upon is, for the purposes of this paper, a dynamic graphical Bayesian model customized to the needs of multi-expert systems, here called distributed dynamic model (DDM). Graphical models provide a faithful picture of the relationships existing between the main features of the problem, which can be discussed and explored by the collective (for an introduction to graph theory, see for example West 2001). In a DDM these relationships are depicted by a directed acyclic graph (DAG), whose vertices are the time series $\{\boldsymbol{Y}_t(i)\}_{t=1,\dots,T}$, $i = 1,\dots,n$. Here we assume that the vector $\boldsymbol{Y}^T$ includes all the variables the collective is planning to take into account during the analysis. We are now ready to formally define the DDM model class.

Definition 1. *A **distributed dynamic model (DDM)** for the time series $\{\boldsymbol{Y}_t\}_{t=1,\dots,T}$ consists of :*

- $n-1$ *conditional independence statements for each time point t of the form*

$$\boldsymbol{Y}_t(i) \perp\!\!\!\perp \boldsymbol{Q}^t(i) \mid \boldsymbol{Z}^t(i), \boldsymbol{Y}^{t-1}(i);$$

- *A DAG $\mathcal{G}$ whose vertices are $\{\boldsymbol{Y}_t(i)\}_{t=1,\dots,T}$, $i = 1,\dots,n$ and its edge set includes an edge $(\{\boldsymbol{Y}_t(i)\}, \{\boldsymbol{Y}_t(j)\})$ if $\boldsymbol{Y}_t(i) \in \boldsymbol{Z}_t(j)$ for any $t = 1,\dots,T$.*

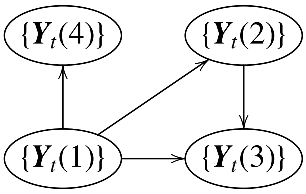

Figure 2: Example of a simple time slice DAG of a DDM. This DAG depicts relationships between processes, not variables, i.e. contemporaneous relationships between variables at the next time step given all the past observations.

We use the common notation $\boldsymbol{Y}_t(i) \perp\!\!\!\perp \boldsymbol{Q}^t(i) \mid \boldsymbol{Z}^t(i), \boldsymbol{Y}^{t-1}(i)$ (Dawid 1979) to read that the vector $\boldsymbol{Y}_t(i)$ is independent of $\boldsymbol{Q}^t(i)$ given $\boldsymbol{Z}^t(i)$ and $\boldsymbol{Y}^{t-1}(i)$, so that the only information to infer $\boldsymbol{Y}_t(i)$ from $\boldsymbol{Q}^t(i)$, $\boldsymbol{Z}^t(i)$ and $\boldsymbol{Y}^{t-1}(i)$ is from $\boldsymbol{Z}^t(i)$ and $\boldsymbol{Y}^{t-1}(i)$. Specifically in a DDM a random vector, $\boldsymbol{Y}_t(i)$ say, at a certain time point can be dependent on a subset of contemporaneous variables, $\boldsymbol{Z}_t(i)$, the values of these variables at previous time points, $\boldsymbol{Z}^{t-1}(i)$, and its previous values, $\boldsymbol{Y}^{t-1}(i)$.

An example of the DAG associated to a DDM is presented in Figure 2. This network is used throughout the paper to illustrate various features of our methodology as we develop it. Such a DAG can be thought of as specifying relevant relationships across the components of different time series. This is in contrast to the more common BN, whose DAG represents relationships between single variables. It is important to note that statements embodied within this DAG are *qualitative* in nature and so in particular can more easily provide the framework for a common knowledge base (see for example Smith 1996, for a discussion of these issues).

Within this DDM framework each time slice $\boldsymbol{Y}_t$ of a DDM, conditionally on the past, can also be described graphically by the same DAG, whose topology does not change between time slices. Each of these DAGs has vertex set equal to $\{\boldsymbol{Y}_t(i), i = 1, \dots, n\}$ and an edge from $\boldsymbol{Y}_t(i)$ to $\boldsymbol{Y}_t(j)$ if $\boldsymbol{Y}_t(i) \in \boldsymbol{Z}_t(j)$. The vertex $\boldsymbol{Y}_t(i)$ is then usually called *parent* of $\boldsymbol{Y}_t(j)$, while $\boldsymbol{Y}_t(j)$ is called *child* of $\boldsymbol{Y}_t(i)$. We call this DAG the **time slice DAG** of the DDM. Recall that a vertex with no children is usually called *leaf* of the DAG.

Associated to any time slice DAG there is a partial order over the vector $\boldsymbol{Y}_t$, such that an element $\boldsymbol{Y}_t(i)$ is called a *descendant* of $\boldsymbol{Y}_t(j)$ and $\boldsymbol{Y}_t(j)$ is called an *ancestor* of $\boldsymbol{Y}_t(i)$, if there is a directed path from $\boldsymbol{Y}_t(j)$ to $\boldsymbol{Y}_t(i)$. A directed path between any two vertices is a sequence of vertices for which any two adjacent elements of the sequence are connected by an edge. We denote with $A(\boldsymbol{Y}_t(i))$ the ancestral set of $\boldsymbol{Y}_t(i)$, consisting of the set including all the ancestors of $\boldsymbol{Y}_t(i)$ together with $\boldsymbol{Y}_t(i)$ itself. In the following, such a partial order becomes fundamental since it induces a partial order over the available decision spaces and guides the algorithm for the computation of expected utility scores we define below.

Just as for other Bayesian graphical models in the literature, the DDM can be associated with a factorization of the probability density function, which depends on the topology of the DAG. Specifically, as a direct consequence of the conditional independence structure associated with a DDM, we have the following result.

Proposition 1. *The joint probability density function f associated to a DDM for the time series $\{\boldsymbol{Y}_t\}_{t=1,\dots,T}$ can be written as*

$$f(\boldsymbol{y}^T) = \prod_{t=1}^{T}\prod_{i=1}^{n} f_{t,i}(\boldsymbol{y}_t(i)|\boldsymbol{y}^{t-1}(i), \boldsymbol{z}^t(i)), \tag{1}$$

where $\boldsymbol{z}^t(i)$ denotes a generic instantiation of $\boldsymbol{Z}^t(i)$.

For the purpose of the collective's specification of the overarching probability model, it is only relevant that the probability density can be *qualitatively* written as a product of the terms $f_{t,i}(\cdot)$. The actual algebraic form of these terms and the *quantitative* specification of the associated parameters is agreed, as we specify below and as usual in practice, by the relevant panel members only. In this sense the algorithms we derive below are built on the agreed *qualitative* framework within all the participants' common knowledge base.

We note here that the DDM model class is very large. Particular instances of the DDM have been extensively studied in the literature. For example, it can be shown that the multi-regression dynamic model (MDM) (Queen and

Smith 1993), the dynamic chain graph model (Anacleto and Queen 2013) and of course non-dynamic models such as the BN (Pearl 1988, Cowell et al. 1999) with appropriate global independence assumptions can all be seen as special cases of the DDM. Consequently, all the classes of models cited above can be used to embellish the qualitative structure of a DDM with explicit probabilistic specifications.

Before introducing an assumption concerning the collective's probabilistic agreement, we need to introduce two terms from graph theory. A DAG is said to be *decomposable* if for any vertex $\boldsymbol{Y}_t(i)$ of the DAG with more than one parent, the edge set of the DAG includes an edge $(\boldsymbol{Y}_t(j), \boldsymbol{Y}_t(k))$ for every pair of two of its parents, $\boldsymbol{Y}_t(j)$ and $\boldsymbol{Y}_t(k)$ say, for $j < k < i$. We also say that a DAG is *connected* if every two vertices are connected by a directed path.

We are now ready to make the following assumption.

Structural Assumption 1 (Probabilistic consensus). *The collective agrees to:*

- *describe the predictive factorization of $\boldsymbol{Y}^T$ by a DDM, whose time slice DAGs are connected and decomposable;*
- *assume that the elements of $\{\boldsymbol{Y}_t\}_{t=1,\dots,T}$ are observed according to the order defined by the following rules:*
  - *$\boldsymbol{Y}_{t_1}(i)$ is observed before $\boldsymbol{Y}_{t_2}(j)$ if $t_1 < t_2$, for $i, j = 1, \dots, n$;*
  - *$\boldsymbol{Y}_t(i)$ is observed before $\boldsymbol{Y}_t(j)$ if $\boldsymbol{Y}_t(i) \in A(\boldsymbol{Y}_t(j))$, $i \neq j$.*

The requirement that the graph is decomposable is simply a technical one, similar to those used in junction trees propagation algorithms (Lauritzen 1992), which provide the basis for fast computational algorithms for BNs. In particular it ensures that no new dependencies are introduced in the IDSS through the backward induction steps we define below. Note that any DAG can be converted into a decomposable one which then gives a valid (albeit inefficient) representation of the underlying processes. In this sense this assumption is not too fierce. Furthermore, we can assume without any loss of generality that the network is connected, since if this was not the case, then, when the structural assumptions we introduce below hold, the overall problem could be decomposed into smaller and independent ones that could be treated separately.

However, more critical, especially in emergency management where it can be commonly violated, is the assumption, as expressed in the second bullet, that it is possible to observe all the quantities the collective planned to observe in the order they happen. In practice we are assuming that the receipt of information is never delayed. We briefly discuss two potential practical ways of addressing violations of this assumption in the discussion below.

2.1.2. The decision space.

As we specify below, the structure of the decision space the collective shares assumes that they have the possibility of intervening after having observed any variable in the system. For this purpose let $\mathcal{D}_t(i)$ be a decision space, for $t = 1, \dots, T$ and $i = 1, \dots, n$ and let $\boldsymbol{\mathcal{D}}_t = (\mathcal{D}_t(1), \dots, \mathcal{D}_t(n))$. The decision space $\mathcal{D}_t(i)$ corresponds to the one available after having observed $\boldsymbol{Y}_t(i)$: see below. Let $\mathcal{D}_0$ denote the decision space associated to an initial decision and $\boldsymbol{\mathcal{D}}^t = (\mathcal{D}_0, \dots, \boldsymbol{\mathcal{D}}_t)$. Further define

$$\boldsymbol{A}(\mathcal{D}_t(i)) \triangleq (\mathcal{D}_t(j), \text{ for } j \neq i \text{ s.t } \boldsymbol{Y}_t(j) \in A(\boldsymbol{Y}_t(i))),$$
$$\boldsymbol{A}^t(\mathcal{D}_t(i)) \triangleq (\mathcal{D}_0, \boldsymbol{A}(\mathcal{D}_1(i)), \dots, \boldsymbol{A}(\mathcal{D}_t(i))).$$

The first quantity is equivalent to an ancestral set for decision spaces and includes those belonging to the same time slice whose decisions are made before $d_t(i) \in \mathcal{D}_t(i)$. Denote by $d_t(i)$, $\boldsymbol{d}_t$, $\boldsymbol{d}^t$, $\boldsymbol{A}(d_t(i))$, $\boldsymbol{A}^t(d_t(i))$ generic instantiations of $\mathcal{D}_t(i)$, $\boldsymbol{\mathcal{D}}_t$, $\boldsymbol{\mathcal{D}}^t$, $\boldsymbol{A}(\mathcal{D}_t(i))$ and $\boldsymbol{A}^t(\mathcal{D}_t(i))$ respectively. We further let $\boldsymbol{d}$ denote a generic instantiation of $\boldsymbol{\mathcal{D}}^T$.

Before formally defining the nature of the decision problem through the following structural assumption, we illustrate our notation using the network of Figure 2 for the time slice at time t. Suppose there are four decision spaces $\mathcal{D}_t(1), \mathcal{D}_t(2), \mathcal{D}_t(3)$ and $\mathcal{D}_t(4)$ associated to the given time slice DAG. A user then needs to commit to a decision $d_t(i) \in \mathcal{D}_t(i)$ only after having observed the value of $\boldsymbol{Y}_t(i)$, $i = 1, \dots, 4$: in our notation only after having observed $A(\boldsymbol{Y}_t(i))$ and $\boldsymbol{Y}^{t-1}$. However, since there is no fixed order in which the variables $\boldsymbol{Y}_t(4)$ and $\boldsymbol{Y}_t(2)$ are observed, then there is no fixed order in which a user commits to the decisions $d_t(2) \in \mathcal{D}_t(2)$ and $d_t(4) \in \mathcal{D}_t(4)$. Of course a decision $d_t(i) \in \mathcal{D}_t(i)$ is made after having already committed to $\boldsymbol{d}^{t-1} \in \boldsymbol{\mathcal{D}}^{t-1}$. We further assume that the overall decision space is such that $\mathcal{D}_t(4) \times \mathcal{D}_t(2)$, so that in particular these two decision spaces do not constrain one another.

We next make the following assumption.

Structural Assumption 2 (Structure consensus). *The collective agrees:*

- *the specification of the decision spaces $\mathcal{D}_0$ and $\mathcal{D}_t(i)$, $i = 1, \ldots, n$, $t = 1, \ldots, T$, defining the acts a user might take;*
- *that they are prepared to assume that the choice of a decision $d_t(i) \in \mathcal{D}_t(i)$ is not constrained by a decision $d_r(j) \in \mathcal{D}_r(j)$ if $\mathcal{D}_r(j) \notin \boldsymbol{A}^t(\mathcal{D}_t(i))$, $j < i$, $r \leq t$;*
- *that they need to commit to a decision $d_t(i) \in \mathcal{D}_t(i)$ only after having observed the value of $A(\boldsymbol{Y}_t(i))$ and $\boldsymbol{Y}^{t-1}$, and having already committed to decisions $\boldsymbol{A}(d_t(i))$ and $\boldsymbol{d}^{t-1} \in \boldsymbol{\mathcal{D}}^{t-1}$;*
- *that the underlying DDM remains valid under any policy choice open to the center.*

Assumption 2 guarantees that the graphical framework of the IDSS remains unaffected after a decision is taken, so that the system provides a coherent picture of the problem throughout the unfolding of events and actions in a particular incident. This is because under the assumption above the topology of the DAG of the DDM and subsequent time slice DAGs do not change. So the algorithms we define below are still able to compute coherent expected utility scores through message passing. Of course we can still allow for the possibility that the probability judgments *within* that structure might change in response to a decision - they usually do.

Note that the overall decision space is not necessarily a simple product of the individual decision spaces. We are therefore addressing decision problems which may not be symmetric and cannot be described by simple influence diagrams (see e.g. Jensen and Nielsen 2009). However, we have restricted some of the policies that can be considered within the IDSS framework: see the example above. In addition, we need our vector of decision spaces only to be partially ordered consistently with our DAG, but not necessarily totally ordered.

The second structural assumption about the structure of the decision problem concerns a set of irrelevance statements.

Structural Assumption 3 (Decisions' irrelevance). *The collective agrees that*

$$f_{t,i}(\boldsymbol{y}_t(i)|\boldsymbol{d}, \boldsymbol{z}^t(i), \boldsymbol{y}^{t-1}(i)) = f_{t,i}(\boldsymbol{y}_t(i)|\boldsymbol{A}^t(d_t(i)), \boldsymbol{z}^t(i), \boldsymbol{y}^{t-1}(i)), \tag{2}$$

for $i = 1, \ldots, n$ and $t = 1, \ldots, T$.

Equation (2) states that a random vector $\boldsymbol{Y}_t(i)$ does not functionally depend on the decisions that are not included in $\boldsymbol{A}^t(\mathcal{D}_t(i))$. This assumption is a very weak one. For example the *sufficiency theorem* (Smith 1989a,b) guarantees that a decision center can always find *one* Bayes optimal decision based on a decision rule which respects these statements. We further note here that within each time slice this assumption is an instance of the *causal consistency lemma* of Cowell et al. (1999), but applied to this more general setting. The lemma guarantees that decisions can have a direct influence only on variables that are yet to be observed. More generally here, Structural Assumption 3 implies the lemma holds for partially ordered decisions and decision spaces that are not simply product spaces.

Return now to the example above. Structural Assumption 3 demands that, because there is no fixed order between $\boldsymbol{Y}_t(2)$ and $\boldsymbol{Y}_t(4)$, $\boldsymbol{Y}_t(4)$ does not functionally depend on $d_t(2) \in \mathcal{D}_t(2)$. Similarly, we require that $\boldsymbol{Y}_t(3)$ and $\boldsymbol{Y}_t(2)$ do not functionally depend on $d_t(4) \in \mathcal{D}_t(4)$. These are the irrelevances the collective *needs* to be ready to assume. Of course they might believe that some decisions do not have any direct effect to additional variables and thus assume further irrelevances. For example they might believe that the initial decision space $\mathcal{D}_0$ is irrelevant for the outcomes of the variables at the first time point, $\boldsymbol{Y}_1$.

2.1.3. The utility integrating structure.

The last overarching agreement the collective needs to find concerns the utility factorization over the time series with indices in UT. We assume here that a set of assumptions implying the existence of a utility function u over $\{\boldsymbol{Y}_t(\text{UT})\}_{t=1,\ldots,T}$ is appropriate, where $\boldsymbol{Y}_t(\text{UT}) = (\boldsymbol{Y}_t(i),\ i \in \text{UT})$ (see for example French and Rios Insua 2000). For simplicity we also assume that all the leaf vertices of the time slice DAGs are attributes of the decision problem.

The utility function describes the preferential structure of the collective. When there are more than one or two attributes, a faithful elicitation of such a function is difficult, unless certain preferential independence conditions are imposed. An added problem in the multi-expert setting we study here is that joint utility elicitations across different panels in a single integrating decision conference are only rarely possible. So for example it is typically possible to elicit the scores associated with the overall weighting of one attribute over another, for example as expressed by the *criterion* of multi-attribute independent utilities. But other more detailed elicitations, for example the appropriate forms of the marginal utility functions, are better delegated to those closest to understanding the consequences of such attributes. However, for this type of delegation to be formally justified it is first necessary to assume that the collective is prepared to entertain certain sets of preferential independences in order to be able to elicit, through local experts' assessments, a joint utility function.

Here we define a multi-attribute utility factorization compatible with the DAG of the multi-expert DDM we defined above. Specifically, this corresponds to an additive decomposition (Fishburn 1967) through time of the utility of the overall time series $\{\boldsymbol{Y}_t\}_{t=1,\dots,T}$ and, within each time slice, a particular type of decomposition with mutually utility independent attribute (m.u.i.a.) (Keeney and Raiffa 1993, Smith 2010).

Definition 2. *Let $\mathcal{G}$ be the DAG of a DDM of a time series $\{\boldsymbol{Y}_t\}_{t=1,\dots,T}$. Let* $\mathrm{UT} \subseteq [n]$, *where $[n] = \{1, \dots, n\}$, be the index set of the attributes of the decision problem. We say that a utility function u is in the class $\mathbb{U}_{\mathcal{G}}$ of utility factorizations* ***compatible*** *to the graph $\mathcal{G}$, if it can be written as*

$$u^{\mathcal{G}}(\boldsymbol{y}^T(\mathrm{UT}), \boldsymbol{d}) = \sum_{t=1}^{T} k_t u_t^{\mathcal{G}}(\boldsymbol{d}^t, \boldsymbol{y}_t(\mathrm{UT})), \tag{3}$$

where

$$u_t^{\mathcal{G}}(\boldsymbol{d}^t, \boldsymbol{y}_t(\mathrm{UT})) = \sum_{i \in \mathrm{UT}} \sum_{I \in C(i)} r_t^{n_I - 1} \prod_{j \in I} k_t(j) u_{t,j}(y_t(j), \boldsymbol{A}^t(d_t(j)), \boldsymbol{d}^t(j)), \tag{4}$$

with $C(i) = \{I \in \mathcal{P}(\mathrm{UT}) : i \in I, \nexists\ j \in I \text{ s.t. } \boldsymbol{Y}_t(j) \notin A(\boldsymbol{Y}_t(i))\}$, $\mathcal{P}(\mathrm{UT})$ is the power set of UT, *n_I is the number of elements of the set I and k_t, $r_t^{n_I}, k_t(j)$ are criterion weights (French and Rios Insua 2000).*

From the definition above we note that $\mathbb{U}_{\mathcal{G}}$ is a subclass of multilinearly decomposed utilities. It contains the popular class of additive (preferentially independent) utilities as a special case. All interaction terms between attributes are dropped if they do not appear in a single ancestral set in the time slice DAG of the DDM. For related graphically based preference models see for example Abbas (2010), Boutilier et al. (2001) and Gonzales and Perny (2004). Fast elicitation routines are also available for these models so that panels need only to answer a few queries for the full utility function associated to a particular graph to be fully defined.

Consider again the DAG in Figure 2. The most general decomposition within the class $\mathbb{U}_{\mathcal{G}}$, if $\mathrm{UT} = \{2, 3, 4\}$, is for the utility to be factored as

$$u^{\mathcal{G}}(\boldsymbol{y}^T(\mathrm{UT}), \boldsymbol{d}) = k_1 u_1^{\mathcal{G}}(\boldsymbol{y}_1, \boldsymbol{d}^1) + k_2 u_2^{\mathcal{G}}(\boldsymbol{y}_2, \boldsymbol{d}^2),$$

where

$$u_1^{\mathcal{G}}(\boldsymbol{y}_1, \boldsymbol{d}^1) = k_1(2)u_{1,2} + k_1(3)u_{1,3} + r_1 k_1(2)k_1(3)u_{1,2}u_{1,3} + k_1(4)u_{1,4},$$
$$u_2^{\mathcal{G}}(\boldsymbol{y}_2, \boldsymbol{d}^2) = k_2(2)u_{2,2} + k_2(3)u_{2,3} + r_2 k_2(2)k_2(3)u_{2,2}u_{2,3} + k_2(4)u_{2,4}.$$

Note that for ease of exposition we have not explicitly written the arguments of the marginal utility functions, which have been formally defined in equation (4). For conciseness, where no confusion can arise, we henceforth write for example $u_{1,3}$ instead of $u_{1,3}(\boldsymbol{y}_1(3), d_1(3), d_1(2), d_1(1), d_0)$.

We show below that if the collective agrees to use a compatible utility factorization, then the judgments delivered individually by the panels of experts are sufficient to compute the expected utility values of the various available policies.

We now make the following assumption.

Structural Assumption 4 (Preferential consensus). *The collective is able to identify an agreed compatible multi-attribute utility decomposition over $\boldsymbol{Y}^T$*(UT) *within the class $\mathbb{U}_{\mathcal{G}}$ and to elicit the associated common criterion weights.*

As for the probabilistic counterpart, the agreement on both the form of the utility function and the values of the criterion weights takes place during decision conferences across sets of representatives of each panel in the collection.

2.1.4. The IDSS expected utility.

Under the structural assumptions introduced above, which specify the qualitative structure of the decision problem, the expected utility function factorizes into separate functions of the beliefs that particular single panels can provide themselves. To show this, for $t = 1, \ldots, T-1$, let

$$\bar{u}^t(\boldsymbol{y}^{t-1}, \boldsymbol{d}^{t-1}) \triangleq \int_{\mathcal{Y}} \left(u_t^{\mathcal{G}}(\boldsymbol{d}^t, \boldsymbol{y}_t(\text{UT})) + \bar{u}^{t+1}(\boldsymbol{y}^t, \boldsymbol{d}^t) \right) f(\boldsymbol{y}_t|\boldsymbol{y}^{t-1}, \boldsymbol{d}^t)\mathrm{d}\boldsymbol{y}_t,$$

and for $t = T$

$$\bar{u}^T(\boldsymbol{y}^{T-1}, \boldsymbol{d}^{T-1}) \triangleq \int_{\mathcal{Y}} u_T^{\mathcal{G}}(\boldsymbol{d}^T, \boldsymbol{y}_T(\text{UT})) f(\boldsymbol{y}_T|\boldsymbol{y}^{T-1}, \boldsymbol{d}^T)\mathrm{d}\boldsymbol{y}_T.$$

These two terms correspond to the expected utility scores after marginalization steps have been performed over all the variables with time index bigger or equal than t in the algorithms we define below.

We now show that any function $\bar{u}^t$, $t = 1, \ldots, T$ can be deduced recursively as a function of the individual panels' statements. First define $Le(\mathcal{G})$ to be the set including the indices of the time series that correspond to leaf vertices in the DAG $\mathcal{G}$ and $Son(i)$ to be the set including the indices of the sons of $\boldsymbol{Y}_t(i)$ in the DAG of the DDM. We call a vertex $\boldsymbol{Y}_t(j)$ the *son* of $\boldsymbol{Y}_t(i)$, if $\boldsymbol{Y}_t(i)$ is the unique vertex in the parent set of $Y_t(j)$, $Pa(\boldsymbol{Y}_t(j))$, such that $Pa(\boldsymbol{Y}_t(i)) = \{\boldsymbol{Y}_t(j), Pa(\boldsymbol{Y}_t(j))\}$. We further call $\boldsymbol{Y}_t(i)$ the *father* of $\boldsymbol{Y}_t(j)$ and we let $Fa(j)$ denote the index of the father of $\boldsymbol{Y}_t(j)$, in this case i. For example, $\{\boldsymbol{Y}_t(2)\}$ is the father of $\{Y_t(3)\}$ in the DAG of Figure 2, while $\{\boldsymbol{Y}_t(1)\}$ has two sons, $\{\boldsymbol{Y}_t(2)\}$ and $\{\boldsymbol{Y}_t(4)\}$. Finally define

$$\boldsymbol{M}_t(i) = (\boldsymbol{Y}_t(j) \text{ s.t. there is a path between } \boldsymbol{Y}_t(i) \text{ and } \boldsymbol{Y}_t(j)), \qquad \boldsymbol{M}^t(i) = \{\boldsymbol{M}_1(i), \ldots, \boldsymbol{M}_t(i)\},$$

and let $\boldsymbol{m}_t(i)$ and $\boldsymbol{m}^t(i)$ denote generic elements of $\boldsymbol{M}_t(i)$ and $\boldsymbol{M}^t(i)$ respectively.

Theorem 1. *Under Structural Assumptions 1-4, $\bar{u}^t$, for $t = 1, \ldots, T$ can be written as*

$$\bar{u}^t = \int_{\mathcal{Y}_1} \tilde{u}_{t,1}(\boldsymbol{m}^t(1), \boldsymbol{y}^t(1), \boldsymbol{A}^t(d_t(1)), \boldsymbol{d}^t(1)) f_{t,1}(\boldsymbol{y}_t(1)|\boldsymbol{y}^{t-1}(1), \boldsymbol{d}^{t-1}(1))\mathrm{d}\boldsymbol{y}_t(1), \tag{5}$$

where

$$\tilde{u}_{t,i}(\boldsymbol{m}^t(i), \boldsymbol{y}^t(i), \boldsymbol{A}^t(d_t(i)), \boldsymbol{d}^t(i)) \triangleq \begin{cases} k_t k_t(i) u_{t,i} + \hat{u}_{t,i}(\boldsymbol{y}^t(i), \boldsymbol{A}^t(d_t(i)), \boldsymbol{m}^t(i), \boldsymbol{d}^t(i)), & i \in Le(\mathcal{G}) \\ \sum_{j \in Son(i)} \bar{u}_{t,j}(\boldsymbol{y}^{t-1}(j), \boldsymbol{d}^{t-1}(j), \boldsymbol{m}^t(j), \boldsymbol{A}^t(d_t(j))), & i \notin \text{UT} \\ k_t k_t(i) u_{t,i} + \sum_{j \in Son(i)} \left(\bar{u}_{t,j} + r_t k_t(i) u_{t,i} \bar{u}_{t,j} \right), & \textit{otherwise}, \end{cases} \tag{6}$$

$$\bar{u}_{t,i}(\boldsymbol{y}^{t-1}(i), \boldsymbol{A}^t(d_t(i)), \boldsymbol{d}^{t-1}(i), \boldsymbol{m}^t(i)) \triangleq \int_{\mathcal{Y}_i} \tilde{u}_{t,i} f_{t,i}(\boldsymbol{y}_t(i)|\boldsymbol{A}^t(d_t(i)), z^t(i), \boldsymbol{y}^{t-1}(i))\mathrm{d}\boldsymbol{y}_t(i), \tag{7}$$

and $\hat{u}_{t,i}$ is uniquely defined as the function for which it holds that

$$\bar{u}^{t+1} \triangleq \sum_{i \in Le(\mathcal{G})} \hat{u}_{t,i}(\boldsymbol{y}^t(i), \boldsymbol{A}^t(d_t(i)), \boldsymbol{m}^t(i), \boldsymbol{d}^t(i)), \tag{8}$$

The proof of this Theorem is provided in Appendix A.

We note here that again the actual algebraic form of the terms in equations (5)-(8) is not fundamental to the construction of a coherent distributed IDSS. This form depends on the individual panels' agreements concerning the quantities under their particular jurisdiction. Importantly, however, any $\bar{u}^t$ can be written as a function of these terms, whatever they are. Its computation, as we show in the following section, can therefore be obtained through a message passing algorithm between each individual panel and the SB.

The quantities appearing in the theorem above are fundamental to later developments of this paper, so we now discuss their interpretation. The definition of $\tilde{u}_{t,i}$ in equation (6) depends on whether or not $\boldsymbol{Y}_t(i)$ is a leaf of the time slice DAG. It adds $u_{t,i}$ to the relevant messages, either $\hat{u}_{t,i}$ or $\bar{u}_{t,i}$, that panel G_i receives during the backward induction we define below. On the other hand, if $i \notin \mathrm{UT}$, equation (6) simply consists of the sum of the incoming messages received by Panel G_i. Equation (7) defines $\bar{u}_{t,i}$ which consists of the result of a marginalization of $\tilde{u}_{t,i}$ with respect to the conditional density function $f_{t,i}(\cdot)$. Finally, the theorem asserts that $\bar{u}^{t+1}$ can be uniquely written as a linear combination of the functions $\hat{u}_{t,i}$, for $i \in Le(\mathcal{G})$.

2.2. *The component DSSs*

It is often recommended that the evaluation of both the marginal utilities and the conditional probabilities should be delegated to groups of individuals best able to compare the efficacy and the likelihood of different value of that attribute (see for example Neumann and Morgenstern 1947, Edwards 1954). We therefore assume:

Structural Assumption 5 (Local panel consensus). *Every expert within a panel G_i agrees on a probabilistic model for the associated component DSS, $f_{t,i}$, $i = 1, \ldots, n$, $t = 1, \ldots, T$, as a function of its inputs. In addition every expert in G_i shares a marginal utility function over $\boldsymbol{Y}_t(i)$, if $i \in \mathrm{UT}$, $t = 1, \ldots, T$.*

We can allow the experts to come to these agreements in a variety of ways, appropriate depending on the context, for example through a facilitated Bayesian decision conference across its members or by following a *Delphi Protocol* (see e.g. French et al. 2009). Similarly, the probabilistic local agreement might consist of following certain pooling axioms (see e.g. French 2011, Faria and Smith 1997, Wisse et al. 2008) or by using agreed software on expert inputs, for example a probabilistic emulator (see e.g. O' Hagan 2006).

3. The message passing algorithms

The structure of the IDSS has been fully defined. We proceed to discuss the computation of the expected utilities through message passing and the possible interrogation support the system can provide to users. We first introduce a message passing algorithm which includes partial optimization steps to deduce an optimal expected utility score. We then consider two special cases of this algorithm. The first does not include optimization steps and computes the expected utility score of a specific policy, while the second works over a non-dynamic network of expert systems.

3.1. *The optimal expected utility algorithm*

In contrast to quantities defined in equations (5)-(8), which compute the expected utility score of a particular policy, we now include optimization steps. These enable us to identify an optimal policy. For this slight generalization we need to first define a new quantity, $u^*_{t,i}$, which accounts for optimizations over decision spaces. Let

$$u^*_{t,i}(\boldsymbol{y}^t(i), \boldsymbol{d}^{t-1}(i), \boldsymbol{m}^t(i), \boldsymbol{A}^t(d_t(i))) \triangleq \max_{\mathcal{D}_t(i)} \tilde{u}_{t,i}(\boldsymbol{y}^t(i), \boldsymbol{m}^t(i), \boldsymbol{A}^t(d_t(i)), \boldsymbol{d}^t(i)). \tag{9}$$

This function is an optimized version, over the decision space $\mathcal{D}_t(i)$, of $\tilde{u}_{t,i}$. We also let $\bar{u}^*_{t,i}$ be the result of the marginalization of $u^*_{t,i}$. Specifically,

$$\bar{u}^*_{t,i}(\boldsymbol{y}^{t-1}(i), \boldsymbol{d}^{t-1}(i), \boldsymbol{m}^t(i), \boldsymbol{A}^t(d_t(i))) \triangleq \int_{\mathcal{Y}_i} u^*_{t,i} f(\boldsymbol{y}_t(i)|\boldsymbol{z}^t(i), \boldsymbol{y}^{t-1}(i), \boldsymbol{A}^t(d_t(i)) \mathrm{d}\boldsymbol{y}_t(i). \tag{10}$$

We next illustrate the algorithm using the network of Figure 2. Before that we introduce a new notation which is also used in the formal algorithms below. We denote with G_i : or SB : the entity that is responsible for the corresponding operation, while we represent with $\longrightarrow G_i$ or $\longrightarrow SB$ the fact that panel G_i and the SB, respectively, receives the value of an appropriate function. So for example $G_i : \tilde{u}_{t,i} \longrightarrow SB$ denotes that panel G_i computes the function $\tilde{u}_{t,i}$ and communicates its value to the SB.

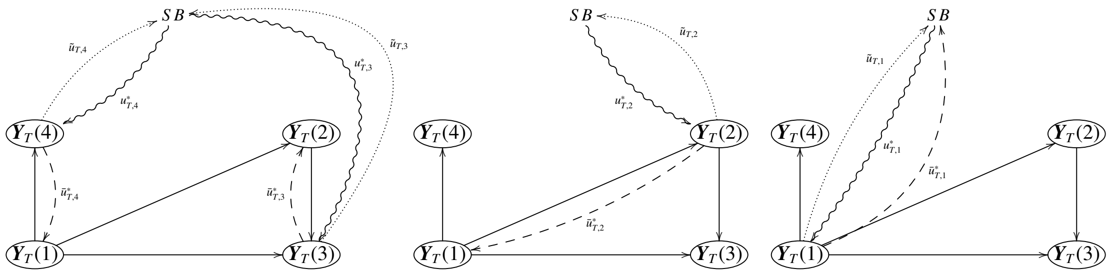

Figure 3: Optimal expected utility algorithm over the last time slice of the network in Figure 2.

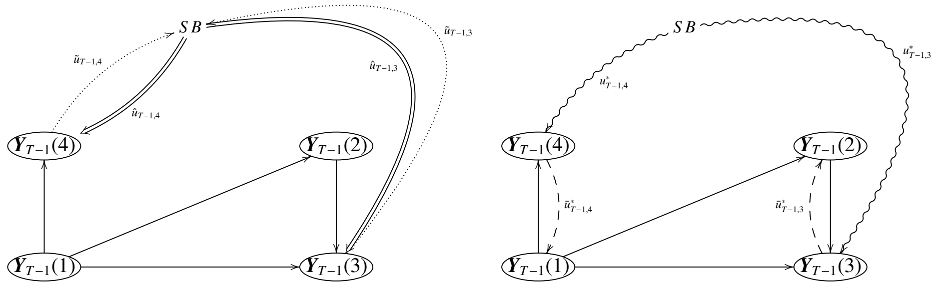

Figure 4: Optimal expected utility algorithm over the $T-1$ time slice of the network in Figure 2.

3.1.1. An illustrative example.

The algorithm starts from the leaves of the last (time T) time slice DAG. Panels $G_i : \tilde{u}_{T,i} \longrightarrow SB$, for $i = 3, 4$. In this case $\tilde{u}_{T,i}$ simply corresponds to $u_{T,i}$ multiplied by the appropriate criterion weights, as defined in equation (6). This step has been depicted by the dotted arrows on the left network of Figure 3 from $\boldsymbol{Y}_T(3)$ and $\boldsymbol{Y}_T(4)$ to the SB. Then $SB : u^*_{T,i} \longrightarrow G_i$ as in equation (9) for $i = 3, 4$. This has been depicted by the curly arrows in the left network of Figure 3. At this stage $G_4 : \bar{u}^*_{T,4} \longrightarrow G_1$ and $G_3 : \bar{u}^*_{T,3} \longrightarrow G_2$, since $\boldsymbol{Y}_T(2)$ is the father of $\boldsymbol{Y}_T(3)$ and $\boldsymbol{Y}_T(1)$ is the father of $\boldsymbol{Y}_T(4)$. These two operations are described by the dashed arrows on the left network of Figure 3.

Now $G_2 : \tilde{u}_{T,2} \longrightarrow SB$ by appropriately adding $u_{T,2}$ to the incoming message from G_3, that is $\bar{u}^*_{T,3}$. Then as before $SB : u^*_{T,2} \longrightarrow G_2$ and $G_2 : \bar{u}^*_{T,2} \longrightarrow G_1$, since $\boldsymbol{Y}_T(1)$ is the father of $\boldsymbol{Y}_T(2)$. The whole process is depicted by the network in the middle of Figure 3, where, as before, a dotted arrow is associated to $\tilde{u}_{T,i}$, a curly arrow to $u^*_{T,i}$ and a dashed one to $\bar{u}^*_{T,i}$.

Because $\boldsymbol{Y}_T(1)$ is the father of both $\boldsymbol{Y}_T(2)$ and $\boldsymbol{Y}_T(4)$, now $G_1 : \tilde{u}_{T,1} \longrightarrow SB$, by simply adding $\bar{u}^*_{T,2}$ and $\bar{u}^*_{T,4}$, received from panels G_2 and G_4 respectively. Note that since $1 \notin \mathrm{UT}$ this panel does not have to add $u_{T,1}$ to the incoming messages. Panel G_1 then repeats the same procedure as the other panels, with the only difference that $\bar{u}^*_{T,1} \longrightarrow SB$ and not to another panel, since it oversees the unique root (i.e. a vertex with no parents) of the DAG. This is depicted by the dashed arrow on the right network of Figure 3.

Theorem 1 states that $\bar{u}^*_{T,1} \equiv \bar{u}^T$ is equal to the sum of the terms $\hat{u}_{T-1,i}$, $i \in Le(\mathcal{G})$. So, if i is the index of a leaf vertex, $SB : \hat{u}_{T-1,i} \longrightarrow G_i$. This is denoted in the left network of Figure 4 by the double arrows. Panels $G_i : \tilde{u}_{T-1,i}$, $i \in Le(\mathcal{G})$, by adding $u_{T-1,i}$ to $\hat{u}_{T-1,i}$. Analogous to the last time slice DAG, these panels then $\longrightarrow SB$. From this stage on, the message passing algorithm copies the calculations and the actions of the previous time slice. So the arrows in Figure 4 match the ones on the left network of Figure 3. The algorithm repeats the same sequence depicted by the dashed, curly, dotted and double arrows in Figure 4, until it reaches the root vertex of the first time slice. When this happens the SB, after receiving $\bar{u}^*_{1,1}$ from panel G_1, computes a final optimization step over the decision space $\mathcal{D}_0$. The algorithm is now completed and can return the expected utility score of the optimal sequence of decisions. This optimal policy can then be communicated to all members of the collective.

3.1.2. The algorithm.

Having described the algorithm on the running example, we now introduce it for a generic DDM and in particular for more realistic scenarios. Specifically, this algorithm takes as inputs all the marginal utility functions, denoted

Algorithm 3.1: THE GROUP OPTIMAL EXPECTED UTILITY$(\boldsymbol{u}, \boldsymbol{k}, \mathcal{G}, \boldsymbol{f})$

for each $i \in Le(\mathcal{G})$
  do $\{\hat{u}_{T,i} = 0$
for $t \leftarrow T$ **downto** 1
  do
    for $i \leftarrow n$ **downto** 1
      do
        if $i \in Le(\mathcal{G})$
          then $\{G_i : \tilde{u}_{t,i} = k_t k_t(i) u_{t,i} + \hat{u}_{t,i} \longrightarrow SB$
          else if $i \in \text{UT}, i \notin Le(\mathcal{G})$
          then $\{G_i : \tilde{u}_{t,i} = k_t k_t(i) u_{t,i} + \sum_{j \in Son(i)} \left(\bar{u}^*_{t,j} + r_t k_t(i) u_{t,i} \bar{u}^*_{t,j}\right) \longrightarrow SB$
          else $\{G_i : \tilde{u}_{t,i} = \sum_{j \in Son(i)} \bar{u}^*_{t,j} \longrightarrow SB$
        $SB : u^*_{t,i} = \max_{\mathcal{D}_t(i)} \tilde{u}_{t,i} \longrightarrow G_i$
        if $(i \neq 1)$
          then $\{G_i : \bar{u}^*_{t,i} = \int_{\mathcal{Y}(i)} u^*_{t,i} f(\boldsymbol{y}_t(i)|\cdot) \mathrm{d}\boldsymbol{y}_t(i) \longrightarrow G_{Fa(i)}$
          else
            $G_i : \bar{u}^*_{t,i} = \int_{\mathcal{Y}(i)} u^*_{t,i} f(\boldsymbol{y}_t(i)|\cdot) \mathrm{d}\boldsymbol{y}_t(i) \longrightarrow SB$
            if $t \neq 1$
              then
                for each $j \in Le(\mathcal{G})$
                  do $\{SB :$ computes $\hat{u}_{t-1,j} \longrightarrow G_j$
              else $SB : u^*_0 = \max_{\mathcal{D}_0} \bar{u}^*_{t,i}$

as $\boldsymbol{u}$, all the criterion weights, $\boldsymbol{k}$, all the conditional density functions $f(\boldsymbol{y}_t(i)|\cdot)$, denoted as $\boldsymbol{f}$, and all the information concerning the DAG $\mathcal{G}$. A formal definition of the algorithm can be found in Algorithm 3.1, which is called henceforth **the group optimal expected utility algorithm**. For simplicity we left implicit the arguments of various quantities the panels and the SB communicate to each other. These simply correspond to the ones defined above. The following result holds.

Theorem 2. *Under Structural Assumptions 1-5, Algorithm 3.1 produces an optimal expected utility score resulting from a unique Bayesian probability model, informed only by the individual judgments delivered by the panels.*

The proof of this theorem is provided in Appendix B.

We note here that often the utility function is a polynomial function in its attributes. When this is so, its expectation is a polynomial in which the variables are, in the continuous case, low order moments. This can dramatically simplify the message passing algorithm above as we illustrate below. As a result the IDSS often needs as inputs only a few low order moments to work coherently. Even in rather complex domains this in turn means that we can expect the algorithms defined above to be almost instantaneous if each component module can produce its forecasts efficiently. A technical study of the polynomial structure of expected utilities in the rather more complex discrete domain is presented in Leonelli et al. (2014).

3.2. Two variations of Algorithm 3.1

3.2.1. The score associated to a generic policy.

Algorithm 3.1 provides an operational guideline on how to compute the score associated to an optimal policy. Recall however that the aim of a DSS is not only to identify the decisions with highest expected utilities, but also to provide explanations and the reasoning behind the outputs it provides (French et al. 2009). It is therefore also relevant to compute the expected utility score associated with any policy that can be adopted. These scores then allow users to compare the different available options in more detail. To accommodate this feature within our IDSS we give in Algorithm 3.2 a simple variant of Algorithm 3.1 which does not include any optimization steps. We call such

Algorithm 3.2: THE GROUP EXPECTED UTILITY$(\boldsymbol{u}, \boldsymbol{k}, \mathcal{G}, \boldsymbol{f}, \boldsymbol{d})$

for each $i \in Le(\mathcal{G})$
  do $\{\hat{u}_{T,i} = 0$
for $t \leftarrow T$ **downto** 1
  do $\{$ **for** $i \leftarrow n$ **downto** 1
    do $\{$ **if** $i \in Le(\mathcal{G})$
      then $\{G_i : \tilde{u}_{t,i} = k_t k_t(i) u_{t,i} + \hat{u}_{t,i} \longrightarrow SB$
      else if $i \in \mathrm{UT}, i \notin Le(\mathcal{G})$
      then $\{G_i : \tilde{u}_{t,i} = k_t k_t(i) u_{t,i} + \sum_{j \in Son(i)} \left(\bar{u}_{t,j} + r_t k_t(i) u_{t,i} \bar{u}_{t,j}\right) \longrightarrow SB$
      else $\{G_i : \tilde{u}_{t,i} = \sum_{j \in Son(i)} \bar{u}_{t,j} \longrightarrow SB$
    if $(i \neq 1)$
      then $\{G_i : \bar{u}_{t,i} = \int_{\mathcal{Y}(i)} u^*_{t,i} f(\boldsymbol{y}_t(i)|\cdot) \mathrm{d}\boldsymbol{y}_t(i) \longrightarrow G_{Fa(i)}$
      else $\{$ $G_i : \bar{u}_{t,i} = \int_{\mathcal{Y}(i)} u^*_{t,i} f(\boldsymbol{y}_t(i)|\cdot) \mathrm{d}\boldsymbol{y}_t(i) \longrightarrow SB$
        if $t \neq 1$
          then $\{$ **for each** $j \in Le(\mathcal{G})$
            do $\{SB :$ computes $\hat{u}_{t-1,j} \longrightarrow G_j$
        else return $(\bar{u}_{t,i})$

algorithm the **the group expected utility** algorithm. Note that this has an additional input from the user, specifically the policy of interest.

Corollary 1. *Under Structural Assumption 1-5, Algorithm 3.2 provides the expected utility score associated to a policy using only the delivered panels' judgments.*

The proof of this result follows directly from Theorem 2. This is because Algorithm 3.2 corresponds to Algorithm 3.1 without lines (10) and (17): i.e. omitting any optimization steps.

As mentioned throughout the paper, the IDSS provides a fast framework for the computation of expected utility scores since it is based on known outputs of individual DSSs. But the distributed nature of the system has more than computational advantages. It gives a route along which the relevant panels can be queried as to the reasons for their contribution to the score, which can be computed using Algorithm 3.2. Such route is described by the diagram in Figure 5. Assume the IDSS is able to compute the expected utility scores of the available countermeasures, as stated in the top central box of the diagram. This is straightforward in the framework we have developed in the previous sections. A user might either be happy with the result produced by the IDSS and proceed with the suggested countermeasure or, alternatively, continue with a further inspection of the factors that are driving the decision making. In the latter case a user might want, for example, to query the IDSS and ask it to provide the individual scores associated to the different attributes for a subset of the policies (second box in the central column). The distributed IDSS we describe above can easily provide such scores. These are then observed and explored by the user, who can then either adopt the suggested action, reappraise the model, or proceed with further inspections of the current model. In the latter case, within the IDSS framework, panels can be directly and separately addressed to provide the reasoning behind their delivered judgments. This is a direct consequence of the distributed nature of the IDSS methodology we develop here. Individual panels are then able to provide explanations to justify their values' choices, usually based on common and scientific knowledge, in several different ways (i.e. graphs, natural language etc.). This additional information might convince the user about the adequacy of the model used, in which case she then might adopt the suggested policy. Alternatively, she might believe that the model is inadequate and be guided to modify the analysis on the basis of the further insights she obtained while exploring the system. In this case the user might then ask

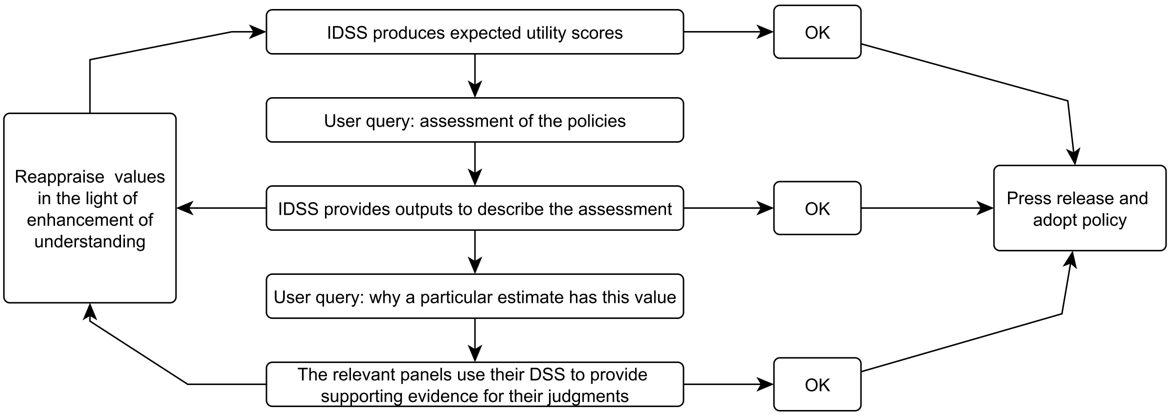

Figure 5: Description of the possible use of an IDSS for a decision analysis.

the queried panel to change the initial estimates they delivered or, for example, to use a different module to process and update the involved probabilities. The exploration of the implications of the models used then continues through the cycle in the diagram of Figure 5 until the user is happy with the outputs of the IDSS and potentially enacts the suggestions the system provided.

3.2.2. The non-dynamic case.

In some domains it may be more appropriate to model decision problems using a non-dynamic probabilistic model, as for example a BN. Within the IDSS framework, this is possible by simply adapting Algorithm 3.1 to the non-dynamic case. Algorithm 3.3, which we call henceforth **non-dynamic optimal expected utility** algorithm, shows how this can be done. Note that we adapt the notation to the non-dynamic case by dropping the dependence on the time-varying index. Although this notation is new, it is self-explanatory and follows straightforwardly from the dynamic one.

Following Algorithm 3.3 a user can identify optimal expected utility scores for the non-dynamic case, as stated by the following corollary.

Corollary 2. *Under Structural Assumption 1-5, Algorithm 3.3 provides an optimal expected utility score for the non-dynamic problem using only the delivered panels' judgments.*

The proof of this result easily follows from Theorem 2, since the last time slice of a dynamic DDM alone can be thought of as a non dynamic network. For this time slice, Theorem 2 guarantees the above corollary holds.

4. The IDSS in practice: illustration of the typical recursions

The previous section formally presented message passing algorithms for IDSSs. These work in general and for any network of expert systems that respect the structural assumptions of Section 2. We now illustrate how the different panels of experts can communicate with each other through an IDSS using the network of Figure 2 concerning the policies after an accidental release of contaminants at a nuclear power plant. We show here the typical recursions of an IDSS in a continuous and dynamic agreed structure.

Many applications we have in mind have a geographical structure, in the sense that many of the values of the required variables are recorded at several locations in an area of interest. This is for example the case in a nuclear emergency, where levels of contamination are collected at several different locations in the surroundings of a power plant. Thus, the processes IDSSs will usually deal with are high dimensional. However, the associated utilities are

Algorithm 3.3: NON-DYNAMIC OPTIMAL EXPECTED UTILITY$(\boldsymbol{u}, \boldsymbol{k}, \mathcal{G}, f)$

for $i \leftarrow n$ **downto** 1

$$\textbf{do}\begin{cases} \textbf{if } i \in Le(\mathcal{G}) \\ \quad \textbf{then } \left\{G_i : \tilde{u}_i = k(i)u_i \longrightarrow SB \right. \\ \quad \textbf{else if } i \in \text{UT}, i \notin Le(\mathcal{G}) \\ \quad \textbf{then } \left\{G_i : \tilde{u}_i = k(i)u_i + \sum_{j \in Son(i)} \left(\bar{u}_j^* + rk(i)u_i\bar{u}_j^*\right) \longrightarrow SB \right. \\ \quad \textbf{else } \left\{G_i : \tilde{u}_i = \sum_{j \in Son(i)} \bar{u}_j^* \longrightarrow SB \right. \\ SB : u_i^* = \max_{\mathcal{D}(i)} \tilde{u}_i \longrightarrow G_i \\ \textbf{if } (i \neq 1) \\ \quad \textbf{then } \left\{G_i : \bar{u}_i^* = \int_{\mathcal{Y}(i)} u_i^* f(\boldsymbol{y}(i)|\cdot)\mathrm{d}\boldsymbol{y}(i) \longrightarrow G_{Fa(i)} \right. \\ \quad \textbf{else } \begin{cases} G_i : \bar{u}_i^* = \int_{\mathcal{Y}(i)} u_i^* f(\boldsymbol{y}(i)|\cdot)\mathrm{d}\boldsymbol{y}(i) \longrightarrow SB \\ SB : u_0^* = \max_{\mathcal{D}_0} \bar{u}_i^* \end{cases} \end{cases}$$

usually low dimensional and can consequently be evaluated transparently. Note that if the impacts of the countermeasures need to be considered at a regional level, it is straightforward to implement these into an IDSS framework. Panels then simply need to provide different scores for the different regions of interest.

Because for real problems the number of equations required to define the problem scales up prohibitively to be concisely reported, the example below illustrates how a geographic component can be included into the analysis in the simplest possible case. However in much larger scenarios the calculations are still very feasible and just as straightforward to calculate as in this example because everything is distributed and in closed form. Furthermore, each of the unknown quantities are in practice numbers provided by the component DSS and so quick to evaluate. Our algorithms therefore straightforwardly and feasibly scale up to realistically larger problems.

4.1. A multiregression dynamic model for a nuclear emergency

The network in Figure 2 gives our representation of the possible policies after an accidental release of contaminants at a nuclear power plant. Let $\{\boldsymbol{Y}_t(1)\}$ be a time series that computes the contamination in a certain area, $\{\boldsymbol{Y}_t(2)\}$ describe the intake of radioactive elements in the population of the area, $\{\boldsymbol{Y}_t(3)\}$ measure the effects on health on the population and $\{\boldsymbol{Y}_t(4)\}$ rank the political disruption in the area consequently to the accident. The topology of this network implies that, conditional on the past, the political disruption in the area is independent of the human intake and the deleterious effects on health given that the amount of contamination has been observed. Four different panels of experts have jurisdiction over one of these time series, such that G_i is responsible for $\{\boldsymbol{Y}_t(i)\}$. Assume further that each vector $\boldsymbol{Y}_t(i) = (Y_t(i,1), \dots, Y_t(i,r))^T$, $i = 1, \dots, 4$, $t = 1, \dots, T$, includes a single continuous random variable which is observed at r different locations in space. These locations are the same for all the time series and do not change through time.

To keep this illustration simple we consider here a simple linear multi-regression dynamic model (LMDM) (Queen and Smith 1993) over a finite time horizon T equal to 2. Specifically, for $i = 2, \dots, 4$, $t = 1, 2$ and $l = 1, \dots, r$, we let

$$Y_t^l(i) = \sum_{j \in Pa(i)} \theta_t^l(j,i) Y_t^l(j) + v_t^l(i), \tag{11}$$

$$\theta_2^l(j,i) = \theta_1^l(j,i) + w_2^l(j,i), \tag{12}$$

where the exponent j does not represent a power exponent but only an index and $Pa(i)$ denotes the parents of $Y_t^l(i)$. Note that equations (11) and (12) implicitly make the simplifying assumption that the processes at different locations are independent of each other. We are further assuming that the intercepts are equal to zero and that a simple steady state dynamic linear model (DLM) (West and Harrison 1997) has been assumed for the root vertex, so that

$$Y_t^l(1) = \theta_t^l(1,1) + v_t^l(1) \qquad \text{and} \qquad \theta_2^l(1,1) = \theta_1^l(1,1) + w_2^l(1,1).$$

The errors $v_t^l(i)$, $w_2^l(j,i)$ are assumed by the collective to be mutually independent with mean zero and variance $V_t^l(i)$ and $W_t^l(j,i)$ respectively. Each panel individually assumed these variances to be unknown, but provided a prior mean estimate $\lambda_t^l(i)$ for $V_t^l(i)$ and $\sigma_2^l(j,i)$ for $W_2^l(j,i)$. Assume further that each panel has provided prior information about the parameter vector at time $t = 1$, such that $a_1^l(j,i)$ is the mean prior estimate of $\theta_1^l(j,i)$, whilst its variance is elicited to be $\tau_1^l(j,i)$. Note that each of these parameters and parameter estimates are possibly a function of the available decisions. Here in order not to make the notation too heavy, we do not explicitly label this dependence. However it is important to remember that these values might be different for each available policy.

An important result associated to LMDMs, and more generally to MDMs, is that the predictive densities $f(\boldsymbol{y}_t|\boldsymbol{y}^{t-1})$, $t = 1,\dots,T$, also enjoy a factorization which respects the topology of the graph (Queen and Smith 1993, Costa et al. 2013). If the errors associated to an LMDM are assumed to be normal, than these predictive densities can be written in closed form as products of multivariate T-distributions. We note that in this example

$$f(\boldsymbol{y}_t|\boldsymbol{y}^{t-1}) = f_{t,4}(\boldsymbol{y}_t(4)|\boldsymbol{y}_t(1),\boldsymbol{y}^{t-1})f_{t,3}(\boldsymbol{y}_t(3)|\boldsymbol{y}_t(2),\boldsymbol{y}_t(1),\boldsymbol{y}^{t-1})f_{t,2}(\boldsymbol{y}_t(2)|\boldsymbol{y}_t(1),\boldsymbol{y}^{t-1})f_{t,1}(\boldsymbol{y}_t(1)|\boldsymbol{y}^{t-1}), \tag{13}$$

and thus equation (13) is an instance of equation (1) specifying the factorization of a DDM. Consequently Algorithm 3.1 can be directly applied to this class of models once these predictive distributions are provided by the individual panels. Note that, because of the distributivity of the system, panels can also provide the reasoning behind the value choices for their parameters, since these will be independent to the ones of the other panels.

Now assume also that the collective has agreed on a linear utility factorization over the attributes such that:

$$u^{\mathcal{G}}(\cdot) = \sum_{i=2}^{4}\sum_{t=1}^{2}\sum_{l=1}^{r} u_i(y_t^l(i)). \tag{14}$$

Thus, for this example we assume that decisions are not attributes of the decision problem. Furthermore the utility function, for each attribute of the decision problem, is assumed to be the same for the r geographical locations and that the overall score of an attribute is equal to the sum of the scores of each region for that attribute. Finally, assume that each panel individually agreed to model the marginal utility function using a simple quadratic function, such that

$$u_i(y_t^l(i)) = -\gamma_t(i)y_t^l(i)^2. \tag{15}$$

The quadratic utility function is a member of the family of constant relative risk aversion utilities and it has been widely used in the literature to model risk aversion (see Wakker 2008). For this marginal utility to be meaningful, we assume the variables $Y_t^l(i)$ take values on the positive real line, $t = 1,\dots,T$, $i = 1,\dots,n$, $l = 1,\dots,r$.

Now that the IDSS has been fully defined for this example, we can show how the algorithm works symbolically when the overarching structure is the LMDM. Since the utility function in equation (15) is a polynomial and has degree two, the algorithm consists of a sequential use of the tower property for the first two conditional moments. Specifically, recall that for any two random variables X and Y,

$$\mathbb{E}(X) = \mathbb{E}(\mathbb{E}(X|Y)), \tag{16}$$

$$\mathbb{V}(X) = \mathbb{V}(\mathbb{E}(X|Y)) + \mathbb{E}(\mathbb{V}(X|Y)) \tag{17}$$

where $\mathbb{E}$ denotes the expectation operator and $\mathbb{V}$ denotes the variance.

Suppose the IDSS needs to identify the expected utility score associated to a specific policy $\boldsymbol{d}$. In this case, by following Algorithm 3.2, $G_3 : \bar{u}_{2,3}$ and $G_4 : \bar{u}_{2,4}$. In this example these computations correspond to a sequential use of the two identities in equations (16) and (17). Specifically, we have that

$$\bar{u}_{2,4} = \mathbb{E}\left(-\gamma_2(4)\sum_{l=1}^{r} Y_2^l(4)^2\right) = -\gamma_2(4)\sum_{l=1}^{r}\left(\mathbb{E}\left(\theta_2^l(1,4)Y_2^l(1)\right)^2 + \mathbb{V}(\theta_2^l(1,4)Y_2^l(1)) + \lambda_2^l(4)\right) \tag{18}$$

$$\bar{u}_{2,3} = \mathbb{E}\left(-\gamma_2(3)\sum_{l=1}^{r} Y_2^l(3)^2\right) = -\gamma_2(3)\sum_{l=1}^{r}\left(\mathbb{E}\left(Y_2^l(3)\right)^2 + \mathbb{V}(Y_2^l(3))\right) \tag{19}$$

$b_1 = -\sum_{i=2}^4 \left(\gamma_2(i) \sum_{l=1}^r \lambda_2^l(i)\right) - \gamma_2(3) \sum_{l=1}^r \lambda_2^l(2)\mathbb{E}(\theta_2^l(2,3))$
$b_2 = -\gamma_2(4) \sum_{l=1}^r \left(\mathbb{E}(\theta_2^l(1,4)\theta_2^l(1,1)) + \mathbb{V}(\theta_2^l(1,4)\theta_2^l(1,1)) + \lambda_2^l(1)\mathbb{E}(\theta_2^l(1,4)^2)\right)$
$b_3 = -\gamma_2(3) \sum_{l=1}^r \mathbb{E}(\theta_2^l(1,3)\theta_2^l(1,1) + \theta_2^l(2,3)\theta_2(1,2)\theta_2^l(1,1))^2$
$b_4 = -\gamma_2(3) \sum_{l=1}^r \mathbb{V}(\theta_2^l(1,3)\theta_2^l(1,1) + \theta_2^l(2,3)\theta_2(1,2)\theta_2^l(1,1))$
$b_5 = -\gamma_2(3) \sum_{l=1}^r -\lambda_2^l(1)\mathbb{E}((\theta_2^l(1,3) + \theta_2^l(2,3)\theta_2(1,2))^2)$
$b_6 = -\gamma_2(2) \sum_{l=1}^r \left(\mathbb{E}(\theta_2^l(1,2)\theta_2^l(1,1))^2 + \mathbb{V}(\theta_2^l(1,2)\theta_2^l(1,1)) + \lambda_2^l(1)\mathbb{E}(\theta_2^l(1,2))^2\right)$

Table 1: Definition of the terms on the rhs of equation (26).

where

$$\mathbb{E}(Y_2^l(3))^2 = \mathbb{E}(\theta_2^l(1,3)Y_2^l(1) + \theta_2^l(2,3)Y_2^l(2))^2, \tag{20}$$

$$\mathbb{V}(Y_2^l(3)) = \lambda_2^l(3) + \mathbb{V}(\theta_2^l(1,3)Y_2^l(1) + \theta_2^l(2,3)Y_2^l(2)). \tag{21}$$

Now, equations (19)-(21) are functions of $Y_2^l(2)$ and $Y_2^l(1)$ only and can therefore $\longrightarrow G_2$. Now $G_2 : \tilde{u}_{2,2}$ as

$$\tilde{u}_{2,2} = -\gamma_2(2) \sum_{l=1}^r y_2^l(2) + \bar{u}_{2,3}, \tag{22}$$

where $\bar{u}_{2,3}$ consists of the rhs of equation (19), where the terms $\mathbb{E}(Y_2^l(3))^2$ and $\mathbb{V}(Y_2^l(3))$ are substituted with the results deduced in equations (20) and (21). Because of the form of these equations we note that Panel G_2 needs to compute only the following quantities for $l = 1, \ldots, r$,

$$\mathbb{E}(\theta_2^l(1,3)Y_2^l(1) + \theta_2^l(2,3)Y_2^l(2))^2 = \mathbb{E}(\theta_2^l(1,3)Y_2^l(1) + \theta_2^l(2,3)\theta_2(1,2)Y_2^l(1))^2, \tag{23}$$

$$\mathbb{V}(\theta_2^l(1,3)Y_2^l(1) + \theta_2^l(2,3)Y_2^l(2)) = \mathbb{V}((\theta_2^l(1,3) + \theta_2^l(2,3)\theta_2(1,2))Y_2^l(1)) + \lambda_2^l(2)\mathbb{E}(\theta_2^l(2,3)), \tag{24}$$

$$\mathbb{E}(Y_2^l(2)^2) = \mathbb{E}(\theta_2^l(1,2)Y_2(1))^2 + \mathbb{V}(\theta_2^l(1,2)Y_2(1)) + \lambda_2(2). \tag{25}$$

Thus, substituting equations (23)-(25) into (22), Panel $G_2 : \tilde{u}_{22}$ and $\longrightarrow G_1$, who simply adds this incoming message to the one received from Panel G_4, as shown in the previous section. Specifically for this example, Panel G_4 communicates equation (18), while Panel G_2 communicates equations (22)-(25). As the other panels, $G_1 : \bar{u}_{2,1} \equiv \bar{u}^2$ by sequentially using the identities in equations (16) and (17), to deduce that

$$\bar{u}_{2,1} = b_1 + b_2 + b_3 + b_4 + b_5 + b_6 \tag{26}$$

where the terms b_i, $i = 1, \ldots, 6$ are defined in Table 1.

At this stage the algorithm continues to work as shown for the last time slice, through a sequential use of the properties of the conditional moments. It can be shown that the final expected utility, $\bar{u}$, associated to a policy $\boldsymbol{d}$ can be written as a function of the elicited beliefs delivered by the panels. Specifically,

$$\bar{u} = \sum_{l=1}^r \sum_{t=1}^2 \bar{\lambda}_t^l + \bar{k}_t^l + \gamma_t(3)\bar{m}_t^l \tag{27}$$

where

$$\bar{\lambda}_t^l = \sum_{i=2}^4 \gamma_t(i)\lambda_t^l(i), \qquad \bar{k}_t^l = \eta_t^l(1,1) \sum_{i=2}^4 \gamma_t(i)k_t^l(1,i), \tag{28}$$

$$\bar{m}_t^l = k_t^l(2,3)(\eta_t^l(1,1)k_t^l(1,2) + \lambda_t^l(2)) + 2a_t^l(1,2)a_t^l(1,3)a_t^l(1,4)\eta_t^l(1,1) \tag{29}$$

and

$$k_1^l(i,j) = a_1^l(i,j)^2 + \tau_1^l(i,j), \qquad \eta_1^l(1,1) = a_1^l(1,1)^2 + \tau_1^l(1,1) + \lambda_1^l(1), \tag{30}$$

$$k_2^l(i,j) = a_1^l(i,j)^2 + \tau_1^l(i,j) + \sigma_2(i,j), \qquad \eta_2^l(1,1) = a_1^l(1,1)^2 + \tau_1^l(1,1) + \lambda_2^l(1) + \sigma_2(1,1). \tag{31}$$

There are a few important points to notice here:

- Because of the form of the utility factorization in equation (14), the expected utility consists of the sum of the expected scores at each location l and of the sum, at each of these locations, of the scores associated to the two time slices. This result is a direct consequence of the independence of the processes at different locations. However, dependence between processes could have been straightforwardly included into this example by defining a hierarchical model over the parameter set;

- The terms in equations (28)-(31) all have a meaning which can be explained to users whenever they query the outputs of the system. Specifically, $k_t^l(i,j)$ corresponds to the score associated to the interaction between $Y_t^l(i)$ and $Y_t^l(j)$, while $\eta_t^l(1,1)$ is associated to the root vertex at time t and at location l. Note that these terms differ between time $t = 1$ and $t = 2$ only by the introduction in the latter case of the estimate of the system error variance $\sigma_2(i,j)$. Thus, $\bar{k}_t^l$ can be thought of as a weighted average of the scores of the interactions between any random variable and $Y_t^l(1)$ at time t and location l, where the weighting factors are the utility coefficients $\gamma_t(i)$. Similarly, $\bar{\lambda}_t^l$ can be seen as a weighted average of the observational variances $\lambda_t^i(l)$ at time t and location l. Lastly, $\bar{m}_t^l$ consists of an interaction term describing the joint relationship between $Y_t^l(1)$, $Y_t^l(2)$ and $Y_t^l(3)$;

- The expected utility in equation (27) is a polynomial, where the unknown quantities are the individual delivered judgments of the panels. This polynomial has for this example degree five and it is not a simple multilinear combination of the unknowns. Note that knowing the shape of the expected utility allows potential users to understand how different factors influence the decision making process;

- The IDSS could have not computed the expected utility of equation (27) if the panels had only delivered the mean estimates of the variables under their jurisdiction. The quantities $\lambda_t^l(i)$, $\sigma_2^l(i,j)$, $\tau_t^l(i,j)$ represent levels of uncertainty concerning these mean estimates. Users would be provided with non coherent scores from DSSs that do not fully integrate the component modules by including such measures. Without these quantities, the overall expected utility consisting only of the mean estimates of the attributes would correspond to

$$\bar{u} = \sum_{l=1}^{r}\sum_{t=1}^{2} a_t^l(1,1)^2 \left(\sum_{i=2}^{4}\gamma_t(i)a_t^l(1,i)^2 + \gamma_t(3)(a_t^l(2,3)^2 a_t^l(1,2)^2 + 2a_t^l(2,3)a_t^l(1,2)a_t^l(1,3))\right), \tag{32}$$

  and this is way different from (27). A DSS that provides expected utility scores from equation (32) could thus lead users to behave as non expected utility maximizer and put them in danger of adopting indefensible countermeasures.

5. Discussion

The implementation of Bayesian methods for a group decision analysis of the type described above has often been considered too difficult to be developed. In this paper we have considered both a dynamic and a non-dynamic framework in which, from a theoretical point of view at least, it is possible to feasibly deal with such a class of problems. As shown by the above example all calculations are straightforward and scale up, albeit with a large number of moments or probabilities to be computed, stored and transmitted between panels. However, these quantities can be provided by an IDSS. So the large number of computations necessary for coherently evaluating different policies are actually trivial ones and computable in real time. We note that the algorithms we defined in this multi-expert system are closely related to the already cited ones for the propagation of probabilities and expected utilities in graphical structures, which have now been successfully implemented in many large applications. So we can be confident that our methods scale up.

The critical assumption of our methodology is that the collective observes all the data they planned to collect. Of course it is very common in practice for data about the development of an accident to arrive non-sequentially. For example, a van collecting deposition measurements can be delayed in transmitting these, so that one has readings from the last hour but not the current ones. In such a situation the distributivity of the system is broken. However there are two possible practical solutions here:

- Panels can accommodate only a subset of the data into the system that is appropriately time ordered. This gives the basis for a framework to analyze the system. This can then be elaborated by the appropriate communication of extra information that does not satisfy the criteria;
- Methods can be developed where the distributivity property can only be approximatively satisfied. These methods are beyond the scope of this paper. We have some encouraging new results in this area which will be formally reported in Smith et al. (2014).

So it is feasible for an IDSS to support rapid policy evaluations even when drawing together judgments from diverse panels of probabilistic expert systems, provided conditions ensure distributivity (or this is approximately so). Distributivity can be guaranteed if the density associated with the graphical statistical model the collective agrees upon factorizes appropriately. In this paper we chose a specific, although rather large, class of directed graphical models where this is so. However, other classes of graphical models can also entertain a distributed analysis. For example chain graph models (Lauritzen and Richardson 2002), CEGs (Smith and Anderson 2008) and their dynamic variant (Barclay et al. 2013) might be able to provide in some domains a better representation of the involved uncertainties and consequently a more focused decision making. We plan to extend our results to additional classes of model in future research.

References

Abbas, A. E., 2010. General decompositions of multiattribute utility functions with partial utility independence. Journal of Multi-Criteria Decision Analysis 17 (1-2), 37–59.

Ackermann, F., Jan. 1996. Participants' perceptions on the role of facilitators using Group Decision Support Systems. Group Decision and Negotiation 5 (1), 93–112.

Anacleto, O., Queen, C. M., 2013. Dynamic chain graph models for high-dimensional time series. Tech. rep., Department of Statistics, Open University.

Barclay, L. M., Smith, J. Q., Thwaites, P. A., Nicholson, A. E., 2013. The Dynamic Chain Event Graph. Tech. Rep. 14, Crism research report, The University of Warwick.

Boutilier, C., Bacchus, F., Brafman, R. I., 2001. UCP-Networks: A Directed Graphical Representation of Conditional Utilities. In: In Proceedings of the Seventeenth Conference on Uncertainty in Artificial Intelligence. pp. 56–64.

Cooke, R. M., 1991. Experts in uncertainty: opinion and subjective probability in science. Oxford University Press.

Costa, L., Smith, J. Q., Nichols, T., 2013. On the selection of multiregression dynamic models of fMRI networked time series. Tech. rep., The University of Warwick.

Cowell, R. G., Dawid, P. A., Lauritzen, S. L., Spiegelhalter, D. J., 1999. Probabilistic networks and expert systems. Springer.

Dawid, P. A., 1979. Conditional independence in statistical theory. Journal of the Royal Statistical Society. Series B (Methodological) 41 (1), 1–31.

De, S., Faria, A. E., Nov. 2011. Dynamic spatial Bayesian models for radioactivity deposition. Journal of Time Series Analysis 32 (6), 607–617.

Edwards, W., 1954. The theory of decision making. Psychological bulletin 51 (4), 380–417.

Ehrhardt, J., 1997. The RODOS system: decision support for nuclear off-site emergency management in Europe. Radiation Protection Dosimetry 73 (1-4), 35–40.

Ehrhardt, J., Pasler-Sauer, J., Schule, O., Benz, G., Rafat, M., Richter, J., 1993. Development of RODOS*, A Comprehensive Decision Support System for Nuclear Emergencies in Europe-An Overview. Radiation Protection Dosimetry 50 (2-4), 195–203.

Faria, A. E., Smith, J. Q., Aug. 1997. Conditionally externally Bayesian pooling operators in chain graphs. The Annals of Statistics 25 (4), 1740–1761.

Fishburn, P. C., 1967. Interdependence and additivity in multivariate, unidimensional expected utility theory. International Economic Review (8), 335–342.

French, S., Sep. 1997. Uncertainty Modelling, Data Assimilation and Decision Support for Management of Off-Site Nuclear Emergencies. Radiation Protection Dosimetry 73 (1), 11–15.

French, S., Feb. 2011. Aggregating expert judgement. Revista de la Real Academia de Ciencias Exactas, Fisicas y Naturales. Serie A. Matematicas 105 (1), 181–206.

French, S., Maule, J., Papamichail, N., 2009. Decision behaviour, analysis and support. Cambridge University Press, UK.

French, S., Ranyard, D. C., Smith, J. Q., 1995. Uncertainty in RODOS. Tech. Rep. 94, University of Leeds, School of Computer Science.

French, S., Rios Insua, D., 2000. Statistical decision theory. Arnold.

Gonzales, C., Perny, P., 2004. GAI Networks for Utility Elicitation. In: Proceedings of the 9th International Conference on the Principles of Knowledge Representation and Reasoning. pp. 224–233.

Jensen, F. V., Nielsen, T. D., 2009. Bayesian Networks and Decision Graphs. Springer.

Keeney, R. L., Raiffa, H., 1993. Decisions with multiple objectives: preferences and value trade-offs. Cambridge University Press.

Lauritzen, S. L., 1992. Propagation of probabilities, eans and variances in mixed graphical association models. Journal of the American Statistical Association 87 (420), 1098–1108.

Lauritzen, S. L., Richardson, T. S., Aug. 2002. Chain graph models and their causal interpretations. Journal of the Royal Statistical Society: Series B (Statistical Methodology) 64 (3), 321–348.

Leonelli, M., Riccomagno, E., Smith, J. Q., 2014. The algebra of discrete decision problems. In preparation.

Leonelli, M., Smith, J. Q., 2013a. Dynamic Uncertainty Handling for Coherent Decision Making in Nuclear Emergency Response. In: Winter Meeting of the American Nuclear Society (to appear).

Leonelli, M., Smith, J. Q., 2013b. Using graphical models and multi-attribute utility theory for probabilistic uncertainty handling in large systems, with application to the nuclear emergency management. In: Data Engineering Workshops (ICDEW), 2013 IEEE 29th International Conference on. IEEE, pp. 181–192.

Neumann, L. J., Morgenstern, O., 1947. Theory of games and economic behavior. Vol. 60. Princeton university press Princeton, NJ.

O' Hagan, A., Oct. 2006. Bayesian analysis of computer code outputs: A tutorial. Reliability Engineering & System Safety 91 (10-11), 1290–1300.

Pearl, J., 1988. Probabilistic reasoning in intelligent systems: networks of plausible inference. Morgan Kaufmann.

Queen, C. M., Smith, J. Q., 1993. Multiregression dynamic models. Journal of the Royal Statistical Society. Series B (Methodological) 55 (4), 849–870.

Richter, K., Gering, F., Müller, H., 2002. Data assimilation for assessing radioactive contamination of feed and foodstuffs. In: Development and application of computer techniques to environmental studies. WIT Press, pp. 393–402.

Smith, J. Q., 1989a. Influence diagrams for Bayesian decision analysis. European Journal of Operational Research 40 (3), 363–376.

Smith, J. Q., 1989b. Influence diagrams for statistical modelling. The Annals of Statistics 17 (2), 654–672.

Smith, J. Q., 1996. Plausible Bayesian games. In: Bayesian Statistics 5. pp. 387–406.

Smith, J. Q., 2010. Bayesian decision analysis: principles and practice. Cambridge University Press.

Smith, J. Q., Anderson, P. E., 2008. Conditional independence and chain event graphs. Artificial Intelligence 172 (1), 42–68.

Smith, J. Q., Barons, M., Leonelli, M., 2014. Coherent inference for integrated decision support systems. In preparation.

Smith, J. Q., Papamichail, N., 1999. Artificial Intelligence Fast Bayes and the dynamic junction forest 107, 99–124.

Wakker, P. P., 2008. Explaining the characteristics of the power (CRRA) utility family. Health Economics 17 (12), 1329–1344.

West, D. B., 2001. Introduction to graph theory. Vol. 2. Prentice hall Upper Saddle River.

West, M., Harrison, J., 1997. Bayesian forecasting and dynamic models. Springer.

Wisse, B., Bedford, T., Quigley, J., 2008. Expert judgement combination using moment methods. Reliability Engineering & System Safety 93 (4), 675–686.

Zheng, D. Q., Leung, J. K. C., Lee, B., Apr. 2009. Online update of model state and parameters of a Monte Carlo atmospheric dispersion model by using ensemble Kalman filter. Atmospheric Environment 43 (12), 2005–2011.

Appendix A. Proof of Theorem 1.

We develop this proof via backward induction both through the vertices of the DAG and through time. For the purpose of this proof define for $t = T$

$$\bar{u}^{T,i}(m^T(i), \boldsymbol{A}^T(d_T(n))) \triangleq \int_{\mathcal{Y}(i)} \cdots, \int_{\mathcal{Y}(n)} k_T u_T f_{T,n} \mathrm{d}\boldsymbol{y}_T(i) \cdots \mathrm{d}\boldsymbol{y}_T(n), \tag{A.1}$$

and note that $\bar{u}^{T,1} \equiv \bar{u}^T$.

First, without any loss of generality, fix a policy $\boldsymbol{d}$. Then start the backward induction from $\boldsymbol{Y}_t(n)$, which, by construction, is a leaf of the time slice DAG. For a leaf $\boldsymbol{Y}_T(i)$ of the time slice DAG at time T, it holds that $k_t k_(i) u_{T,i} = \tilde{u}_{T,i}$. The overall utility function is a function of $\boldsymbol{Y}_T(n)$ only through $\tilde{u}_{T,n}$. Therefore $\tilde{u}_{T,n}$ can then be simply marginalized as in equation (7) to obtain $\bar{u}_{T,n}$. Furthermore

$$\bar{u}^{T,n} = k_T \sum_{i \in \mathrm{UT}} \sum_{I \in C(i)} r_T^{n_I - 1} \prod_{j \in I} k_T(j) \left(\mathbb{1}_{\{j=n\}} \bar{u}_{T,j} + \mathbb{1}_{\{j \neq n\}} u_{T,j} \right). \tag{A.2}$$

Now consider $\boldsymbol{Y}_T(n-1)$. The vertex associated with this random vector in the time slice DAG is either the father of $\boldsymbol{Y}_{T,n}$ or a leaf of the DAG. In the latter case, the exact same method followed for $\boldsymbol{Y}_T(n)$ can be applied to $\boldsymbol{Y}_T(n-1)$, and thus

$$\bar{u}^{T,n-1} = k_T \sum_{i \in \mathrm{UT}} \sum_{I \in C(i)} r_T^{n_I - 1} \prod_{j \in I} k_T(j) \left(\mathbb{1}_{\{j=n,n-1\}} \bar{u}_{T,j} + \mathbb{1}_{\{j \neq n,n-1\}} u_{T,j} \right). \tag{A.3}$$

If on the other hand $\boldsymbol{Y}_T(n-1)$ is the father of $\boldsymbol{Y}_T(n)$, then equation (A.2) is a function of $\boldsymbol{Y}_T(n-1)$ through both $u_{T,n-1}$ and $\bar{u}_{T,n}$. In this case we have that

$$\bar{u}^{T,n-1} = k_T \sum_{i \in \{\mathrm{UT} \setminus n\}} \sum_{I \in C(i)} r_T^{n_I - 1} \prod_{j \in I} k_T(j) \left(\mathbb{1}_{\{j=n-1\}} \bar{u}_{T,j} + \mathbb{1}_{\{j \neq n-1\}} u_{T,j} \right). \tag{A.4}$$

Continuing through this process over the last time slice DAG up to the vertex $\boldsymbol{Y}_T(l)$, $1 \leq l \leq n-1$, we can therefore deduce that

$$\bar{u}^{T,l} = k_T \sum_{i \in \mathrm{UT}^l} \sum_{I \in C(i)} r_T^{n_I - 1} \prod_{j \in I} k_T(j) \left(\mathbb{1}_{\{j \in A_l\}} \bar{u}_{T,j} + \mathbb{1}_{\{j \notin A_l\}} u_{T,j} \right), \tag{A.5}$$

where $\mathrm{UT}^l = \{\mathrm{UT} \setminus \{l+1, \ldots, n\}\}$ and $A_i = \{j \geq l \text{ s.t. } \boldsymbol{Y}_T(j) \text{ has father } \boldsymbol{Y}_T(k) \text{ for } k < l\}$. In particular for $\boldsymbol{Y}_T(2)$ we can write equation (A.5) as

$$\bar{u}^{T,2} = k_T \left(\mathbb{1}_{\{1 \in \mathrm{UT}\}} \left(k_T(1) u_{T,1} + (1 + r_T k_T(1) u_{T,1}) \sum_{j \in Son(1)} \bar{u}_{T,j} \right) + \mathbb{1}_{\{1 \notin \mathrm{UT}\}} \left(\sum_{j \in Son(1)} \bar{u}_{T,j} \right) \right). \tag{A.6}$$

Note that equation (A.6) corresponds to $\tilde{u}_{T,1}$ as defined in equation (6). It then straightforwardly follows that $\bar{u}^T$ can be written as in equation (5).

Now, since $\boldsymbol{Y}_T(1)$ is the unique root of the time slice DAG if $i, j \in Son(1)$, then

$$\boldsymbol{M}_T(i) \cap \boldsymbol{M}_t(j) = \boldsymbol{Y}_t(1), \tag{A.7}$$

where $\boldsymbol{M}_T(i)$ denotes the vector including the vertices of the time slice DAG connected by a path containing $\boldsymbol{Y}_t(i)$. Suppose that any vertex $\boldsymbol{Y}_T(j)$, for $j \in Son(1)$, is either connected by a path to one only leaf of the DAG or is a leaf of the graph itself. Because of the identity in equation (A.7) and because of the algebraic form of equation (A.6), which consists of a linear combination of the terms $\bar{u}_{T,j}$, for $j \in Son(1)$, we can deduce that equation (8) holds for the last time slice.

Now, consider the case where one vertex $\boldsymbol{Y}_t(j)$ with index in $Son(1)$ is connected to more than one leaf. Equation (6) guarantees the existence of a vertex $\boldsymbol{Y}_t(i)$, $j < i$ connected to both $\boldsymbol{Y}_t(j)$ and the above mentioned leaves, such that $\tilde{u}_{T,i}$ can be written as a linear combination of terms $\bar{u}_{T,j}$, for which each of these terms is a function of one of the leaves only. It therefore follows that equation (8) also holds in this case.

At this stage $\boldsymbol{Y}_{T-1}(i)$, for $i \in Le(\mathcal{G})$ appears as a function of $u_{T-1,i}$ and $\hat{u}_{T-1,i}$ only and consequently of $\tilde{u}_{T-1,i}$ only. For $t = 1, \ldots, T-1$, now define

$$\bar{u}^{t,i}(z^t(i), m^{t-1}(i)\boldsymbol{A}^T(d_T(n))) \triangleq \int_{\mathcal{Y}(i)} \cdots , \int_{\mathcal{Y}(n)} (k_t u_t + \bar{u}^{t+1}) f_{t,n} \mathrm{d}\boldsymbol{y}_t(i) \cdots \mathrm{d}\boldsymbol{y}_t(n), \tag{A.8}$$

and note that

$$\bar{u}^{T-1} = \int_{\mathcal{Y}(1)} \cdots \int_{\mathcal{Y}(n)} k_{T-1} \sum_{i \in \mathrm{UT}} \sum_{I \in C(i)} r_{T-1}^{n_I - 1} \prod_{j \in I} u'_{T-1,j} \mathrm{d}\boldsymbol{y}_{T-1}(1) \cdots \mathrm{d}\boldsymbol{y}_{T-1}(n) \tag{A.9}$$

where

$$u'_{T-1,j} = \mathbb{1}_{\{j \in Le(\mathcal{G})\}}(u_{T-1,j} + \hat{u}_{T-1,j}) + \mathbb{1}_{\{j \notin Le(\mathcal{G})\}}(u_{T-1,j}) \tag{A.10}$$

Since equation (39) enjoys the same factorization of the utility function of the last time slice, as specified in Equation (3), the exact same steps that we followed for the time slice T, through equations (A.2)-(A.7), can be followed for time $T-1$. This then also holds for any time slice t, $1 \leq t \leq t-1$ because the utility function is a linear combination of the utilities at each time point.

Appendix B. Proof of Theorem 2.

To prove Theorem 2 we proceed as follows:

- We relate the lines of the pseudo-code of Algorithm 3.1 to the equations (5)-(8) of Theorem 1 and their variations which include optimization steps in equations (9) and (10);
- We then show that each panel and the SB have sufficient information to perform the steps of the algorithm they are responsible for;
- We conclude by showing that the optimization steps, which in the algorithm correspond to lines (10) and (17), are able to identify optimal decisions using only combinations of quantities individual panels are able to calculate.

We start with the first two bullets. The first two lines correspond to a simple initialization step, which sets $\hat{u}_{T,i} = 0$, $i \in Le(\mathcal{G})$, since these do not exist. Line (3) describes the backward induction step over the time index, t, while line (4) does the same over the index of the vertices of the graph, i. Now note that in lines (5)-(9), Panel $G_i : \tilde{u}_{t,i}$ using equation (6). Each panel has enough information to do this, since line (12) guarantees that the scores are communicated to the panels overseeing father vertices and line (16) denotes the fact that the SB transmits $\hat{u}_{t,i}$ to the appropriate panels. The functions $\tilde{u}_{t,i} \longrightarrow SB$, who performs an optimization step in line (10) and communicates the result back to the panel. We address the validity of this step below.

Since the $SB : u^*_{t,i} \longrightarrow G_i$, each panel is able to compute $\bar{u}^*_{t,i}$ (lines 12-13) following equation (10). As noted before, if i is not the root of the DAG, $\bar{u}^*_{t,i}$ is sent to the appropriate panel, whilst if $i = 1$, as specified by the if statement in line (11), $\longrightarrow SB$. For each time slice with time index $t \neq 1$ lines (15)-(16) compute $\hat{u}_{t,i}$, as in equation (8). These are sent to the appropriate panels, which can then continue the backward inductive process from the time slice with a lower time index. If on the other hand $t = 1$, then the expected utility is a function of the initial decision space $\mathcal{D}_t(0)$ only. The SB can then perform a final optimization step over this space and thus conclude the algorithm (line 17).

We now address the optimization steps. The influence on the scores associated with time slices with index bigger than t of a decision space $\mathcal{D}_t(i)$ are included, by construction, only in the terms $\hat{u}_{t,k}$, where k is the index of a descendant $\boldsymbol{Y}_t(k)$ of $\boldsymbol{Y}_t(i)$. Further note that the same decision space $\mathcal{D}_t(i)$ can affect the scores of terms including descendants of $\boldsymbol{Y}_t(i)$ at the same time point. It can further appear as an argument of $u_{t,i}$. Thus the whole contribution of $\mathcal{D}_t(i)$ is summarized within $\tilde{u}_{t,i}$, as it can be seen by recursively using equations (6) and (7).

Now, as specified by equation (9), the optimization step over $\mathcal{D}_t(i)$ is performed by maximizing $\tilde{u}_{t,i}$, which carries all the information concerning this decision space. More specifically, no other term is an explicit function of $\mathcal{D}_t(i)$ at this stage of the algorithm, as guaranteed by equations (2) and (4). Finally, Structural Assumption 2 guarantees that all the elements that appears as arguments of $\tilde{u}_{t,i}$ are observed and therefore known at the time the decision associated to this decision space needs to be made.