An Application of Formal Argumentation: Fusing Bayes Nets in MAS

Søren Holbech Nielsen [a], Simon Parsons [b]
[a] *Department of Computer Science*
Aalborg University, Aalborg
Denmark
[b] *Department of Computer and Information Science*
Brooklyn College, City University of New York
Brooklyn, 11210 NY, USA

AbstractWe consider agents in a multi-agent system, each equipped with a Bayesian network model (BN) of its environment. We want the agents to reach consensus on one compromise network, which may not be identical to a single one of the BNs initially held by the agents, but rather a combination of aspects from each BN. The task can be characterized as the need for agents to agree on a specific state (a BN) of a variable with an enormous state space (all possible BNs). The grandness of the task is reduced by the fact that BNs are composed of local relationships, and it should therefore be possible to reach the compromise by gradually agreeing on parts of it. In the metaphor of the variable, the agents should be able to agree on successively smaller subsets of the enormous state space. However, these same local relationship can interact, and understanding the extent to which partial agreements affect the possible final compromise is a highly complex task. In this work we suggest using formal argumentation as the reasoning mechanism for agents solving this task, and suggest an open-ended agora approach that ensures agents high quality compromises in an anytime fashion.

Keywords. Argumentation, Bayesian networks, Compromises

1. Introduction

We investigate how Bayesian networks (BNs) can be used in a multi-agent setting with the help of argumentation theory. Previously the two methodologies have mainly been studied together with a view to incorporating the efficiency and precision of BNs into argumentation theory (e.g. [1]), or as an exercise in converting models of one theory into models of the other (e.g. [2] and [3]). Here, we envision equipping each agent in a MAS with a BN, as a model of the domain it is situated in, and aim at providing a framework built on formal argumentation principles in which the agents, starting from their individual domain models, can conclude on a single network representing their joint domain knowledge. This would be useful in cases where the agents only meet occasionally and in the meantime may make small changes to their models to reflect surprising observations of their surroundings. By using the two paradigms in this manner, we hope to exploit the strengths of BNs and of argumentation: Allowing individual agents to draw inferences in face of noisy observations using their BNs, and having agents extract a consistent "truth" from a set of conflicting ones through a distributed process built on argumentation.

The task of fusing several BNs into one compromise BN is made theoretically interesting by the fact that BNs by their graphical nature can be decomposed into several local relationships, and thus the aspect of gradually building a compromise BN bottom up is tempting. However, these very same local relationships can interact in complex ways, and the consequences of committing oneself to a partial compromise can be hard to estimate. Maybe because of this difficulty, the task has previously mainly been considered a centralized one-off operation, with little consideration given to these "cascading" effects. Furthermore, the task has been addressed with an a priori specified view to what constitutes an adequate compromise, with no apparent consensus on the goal of network fusion among authors (see [4], [5], [6], and [7]). In this paper, we do not commit ourselves to a specific compromise objective. Rather, we establish a general framework in which any kind of compromise on BNs can be reached in a gradual manner, with the exact nature of the proceedings specified by some parameter functions.

As presented the setup may be confused with a negotiation problem, where the agents would try to negotiate a compromise that is close to their individually held beliefs. However, unlike the standard negotiation setup, the parts of the problem cannot be valued in isolation, and hence, to the individual agent the value of an already agreed upon partial compromise, will depend on the compromise choices that remain to be made. For the same reason, the problem cannot be seen as a distribution of resources, as the individual agents utility of the "resources" would change according to how the remaining ones are distributed. Instead, we hope only to provide the agents with the ability to determine the extent to which they commit themselves at each step in the construction of the compromise. That is, the main focus of our work is to provide the reasoning mechanism individual agents can use for surveying the consequences of committing to partial compromises. The advantages of our approach over previous efforts include: That a general purpose argumentation engine can be implemented and reused in contexts with different definitions of compromise; that efficient distributed implementations are natural; that in cases where agents almost agree a priori, little information need to be shared among the agents; and that anytime compromises can be achieved.

2. Preliminaries And Problem Definition[1]

2.1. Bayesian Nets

A BN $\mathcal{B}$ is an acyclic directed graph (DAG) $\mathcal{G}$,[2] over a set of random variables $\boldsymbol{V}$, along with a conditional probability distribution for each variable in $\boldsymbol{V}$ given its parents in $\mathcal{G}$. The joint probability distribution P over $\boldsymbol{V}$, obtained by multiplying all these conditional probability distributions, adheres to a number of conditional (in)dependence constraints identifiable from $\mathcal{G}$ alone. Any other BN $\mathcal{B}'$ with a graph implying the exact same constraints on P is said to be *equivalent* to $\mathcal{B}$. [9] proved that the set of all BNs equivalent to some BN $\mathcal{B}$ can be uniquely characterized by a partially directed graph called the

[1]For actual examples and background on the topics, ideas, and algorithms presented here and later, refer to [8].

[2]We assume the reader is familiar with the basics of graph theory.

pattern of $\mathcal{B}$. The pattern of $\mathcal{B}$ is constructed by taking the skeleton of $\mathcal{B}$ and directing links as they appear in $\mathcal{B}$ iff they participate in a v-structure[3]. Any BN equivalent to $\mathcal{B}$ can be obtained from its pattern by exchanging links for directed arcs, while taking care that no directed cycles are introduced, and that no v-structures not already found in the pattern are introduced. Any DAG obtained from the pattern in this manner is called a *consistent extension* of the pattern. The two constraints imply that not all partially directed graphs are patterns of some BN, and furthermore that some links in a pattern are exchanged for similarly directed arcs in all consistent extensions of the pattern. Such arcs and arcs found in the pattern are called *compelled arcs*, and the partially directed graph obtained by exchanging links for compelled arcs wherever possible, is called the *completed* pattern of $\mathcal{B}$. The completed pattern of $\mathcal{B}$ is thus a unique characterization of $\mathcal{B}$'s equivalence class as well. The set of all partially directed graphs over $\boldsymbol{V}$ that are completed patterns of some BN thus constitute a complete and minimal encoding of all probabilistic dependencies for distributions expressible by BNs over $\boldsymbol{V}$. We denote this set of completed patterns $\mathfrak{C}^{\boldsymbol{V}}$ and $\mathfrak{C}$ when $\boldsymbol{V}$ is obvious from the context. [10] gave an elegant characterization of the individual elements of $\mathfrak{C}$. Next, we present how agreeing on BNs pose problems.

2.2. Compromising On Bayesian Networks

The problem we are posing arises in a MAS containing a finite number of cooperating agents. Each agent i has a BN $\mathcal{B}_i$ over a common set of domain variables $\boldsymbol{V}$, which we assume to be implicit in the remainder of the text. For ease of exposition, we furthermore assume that an arbitrary but fixed total ordering $\rightsquigarrow$ over the variables is known by all agents a priori. At some point agents 1 to k decide to pool their knowledge, as represented by $\mathcal{B}_1$ to $\mathcal{B}_k$, into a new BN $\mathcal{B}_*$. Facilitating this task is the problem addressed here. We expect $\mathcal{B}_1$ to $\mathcal{B}_k$ to be large but somewhat similar (as each describe relationships among the same variables), and therefore that having each agent communicate its entire model to each other agent is inefficient. We focus solely on the graphical structure of $\mathcal{B}_*$

As all consistent extensions of a completed pattern imply the exact same independence properties, it is reasonable to consider completed patterns as basic representations of domain knowledge, if domain knowledge is taken to be independence properties as in this text. That is, we only require the agents to agree on the completed pattern $\mathcal{G}_* \in \mathfrak{C}$ of $\mathcal{B}_*$.

To establish whether a graph is a good compromise for the agents, we need a measure for how well such graphs matches each of $\mathcal{B}_1$ to $\mathcal{B}_k$. Furthermore, as we plan to build this compromise gradually, we wish for this measure to be relative to an already agreed *partial compromise*. For example, it may be the case that an important dependency between two variables is already a consequence of a partial compromise, and further connections between the two variables may then be of little value. Contrarily, had the partial compromise not implied this dependency, connections that would ensure it are valuable. In general, we cannot assume that a partially specified graph is suitable as representation of a partial compromise, as this might include agreements on what should *not* be part of the final compromise. Therefore, we take a partial compromise $\mathcal{P} \equiv (\boldsymbol{P}_+, \boldsymbol{P}_-)$ to be two sets of sentences in some language, where $\boldsymbol{P}_+$ describe aspects that should be true of the compromise graph, and $\boldsymbol{P}_-$ describe aspects that cannot be true.

[3] A triple of variables (X, Z, Y) is a v-structure if X and Y are non-adjacent and both are parents of Z.

For any three partial compromises $\mathcal{P}$, $\mathcal{P}^a$, and $\mathcal{P}^b$, where $\boldsymbol{P}_+ \subseteq \boldsymbol{P}_+^a$, $\boldsymbol{P}_- \subseteq \boldsymbol{P}_-^a$, $\boldsymbol{P}_+ \subseteq \boldsymbol{P}_+^b$ and $\boldsymbol{P}_- \subseteq \boldsymbol{P}_-^b$, we assume that each agent i can compute its *compromise scores* $s_i(\mathcal{P}, \mathcal{P}^a)$ and $s_i(\mathcal{P}, \mathcal{P}^b)$ such that $s_i(\mathcal{P}, \mathcal{P}^a) > s_i(\mathcal{P}, \mathcal{P}^b)$ iff $\mathcal{P}^a$ describes $\mathcal{B}_i$ better than $\mathcal{P}^b$, given that $\mathcal{P}$ has already been accepted as being descriptive of $\mathcal{B}_i$. A simple example of $s_i((\boldsymbol{P}_+, \boldsymbol{P}_-), (\boldsymbol{P}_+^a, \boldsymbol{P}_-^b))$ could be the number of features described in $\boldsymbol{P}_+^a \setminus \boldsymbol{P}_+$ and $\boldsymbol{P}_-^a \setminus \boldsymbol{P}_-$, which are consistent with $\mathcal{B}_i$, minus those that are not. A more complex score could weigh each of these described features according to the empirical evidence the agent has in favor of or against them. We will assume s_i to be additive, i.e. for any three partial compromises $\mathcal{P}^0$, $\mathcal{P}^1$, and $\mathcal{P}^2$, where $\boldsymbol{P}_+^0 \subseteq \boldsymbol{P}_+^1 \subseteq \boldsymbol{P}_+^2$ and $\boldsymbol{P}_-^0 \subseteq \boldsymbol{P}_-^1 \subseteq \boldsymbol{P}_-^2$, it is the case that $s_i(\mathcal{P}^0, \mathcal{P}^2) = s_i(\mathcal{P}^0, \mathcal{P}^1) + s_i(\mathcal{P}^1, \mathcal{P}^2)$. Notice, that here we do not attempt to define what it means to be a "better description", since we believe that this issue can be dependent on the actual setting in which the framework is to be used, as stated in Section 1.

In addition to the compromise score, we also assume that the agents know the *combination function* $c : \mathbb{R}^k \rightarrow \mathbb{R}$, indicating how much trust should be put into the individual agents' models. Differences in trust can be justified by differences in experiences, sensor accuracies, etc. Formally, we define c as follows: Let $\mathcal{P}$, $\mathcal{P}^a$, and $\mathcal{P}^b$ be partial compromises. If

$$c(s_1(\mathcal{P}, \mathcal{P}^a), \ldots, s_k(\mathcal{P}, \mathcal{P}^a)) > c(s_1(\mathcal{P}, \mathcal{P}^b), \ldots, s_k(\mathcal{P}, \mathcal{P}^b)),$$

when $\mathcal{P}^a$ is a better compromise than $\mathcal{P}^b$ for the group of agents 1 to k, given that they have already agreed on $\mathcal{P}$, then c is the combination function for agents i to k. (An obvious choice for c would be a linear combination of its inputs.) We refer to $c(s_1(\mathcal{P}, \mathcal{P}^a), \ldots, s_k(\mathcal{P}, \mathcal{P}^a))$ as the *joint compromise score* of $\mathcal{P}^a$ given $\mathcal{P}$, and like s_i, we shall also assume that c is additive.

With this notation in place, we can thus restate the problem more formally as finding a partial compromise $\mathcal{P}$, which uniquely identifies some graph $\mathcal{G}^* \in \mathfrak{C}$, such that

$$c(s_1((\varnothing, \varnothing), \mathcal{P}), \ldots, s_k((\varnothing, \varnothing), \mathcal{P})) \geq c(s_1((\varnothing, \varnothing), \mathcal{P}'), \ldots, s_k((\varnothing, \varnothing), \mathcal{P}')),$$

for all other partial compromises $\mathcal{P}'$, which uniquely identifies a graph $\mathcal{G}' \in \mathfrak{C}$.

As presented here, it is clear that the problem is not of a simple binary nature, as we are not trying to establish whether some proposition is true or not, and that we are furthermore dealing with a setting in which more than two agents may interact. Consequently, we cannot utilize the vast literature on dialectic proof theories directly. Rather, the problem we are trying to solve is a distributed maximization over a super exponential hypothesis space ($\mathfrak{C}$). Furthermore, as the worth of (partial) compromises are only specified in relation to already agreed upon compromises, the problem is of a highly dynamic nature.

Our solution to the problem is divided into three parts. First, we create a finite language with which graphs and some essential properties of these can be expressed; second and most importantly, we construct an argumentation system with which the agents can reason about consequences of committing to partial compromises; and thirdly, we create an agora in which the agents can reach compromise graphs in an anytime fashion. First, however, we describe the formal argumentation framework we have selected as a reasoning mechanism.

Formal argumentation takes many forms, but here we see it as an approach to extracting consistent knowledge from a possibly inconsistent knowledge base. No single methodology has yet to stand out as the main approach to argumentation (see [11] for an overview of a series of approaches), so it has been necessary to pick one from a large pool of these. The framework we have picked for our purpose is the framework of [12] (which is a proper generalization of that of [13]), as this is an abstract framework, which leaves the underlying language unspecified, and thus does not force us to specify in advance the reasons to which each agent i may attribute its belief in aspects of $\mathcal{B}_i$.

An *argumentation system* is defined as a pair $\mathcal{A} \equiv (\boldsymbol{A}, \triangleright)$, where $\boldsymbol{A}$ is a set of *arguments*, and $\triangleright \subseteq (2^{\boldsymbol{A}} \setminus \{\varnothing\}) \times \boldsymbol{A}$ is an *attack relation*. The exact nature of an argument is left unspecified, but examples could be "In $\mathcal{B}$ there is an arc from X to Y and Y and Z are adjacent, so there must be an arc from Y to Z" or "Because I have observed r, I believe there is an arc from X to Y in $\mathcal{B}$". For two sets of arguments $\boldsymbol{S} \subseteq \boldsymbol{A}$ and $\boldsymbol{S}' \subseteq \boldsymbol{S}$ and an argument A, if $\boldsymbol{S}' \triangleright A$ then $\boldsymbol{S}$ is said to *attack* A. If no proper subset of $\boldsymbol{S}'$ attacks A, then $\boldsymbol{S}'$ is called a *minimal attack* on A. An example of an attack that would make sense is "There is an arc from X to Y in $\mathcal{B}$"[4]$\triangleright$"There is an arc from Y to X in $\mathcal{B}$".

A *semantics* of an argumentation framework is a definition of the arguments in the framework that should be accepted by a rational individual. [13] and [12] work with a wide range of semantics, but we only introduce those needed here: We define a set of arguments $\boldsymbol{S} \subseteq \boldsymbol{A}$ as being *conflict-free*, if there is no argument $A \in \boldsymbol{S}$ such that $\boldsymbol{S}$ attacks A. We further define a single argument A as being *acceptable with respect to a set of arguments* $\boldsymbol{S}$, if for each set of arguments $\boldsymbol{T} \subseteq \boldsymbol{A}$, where $\boldsymbol{T} \triangleright A$, there is an argument B in $\boldsymbol{T}$, such that $\boldsymbol{S}$ attacks B. A conflict-free set $\boldsymbol{S}$, where all arguments in $\boldsymbol{S}$ are acceptable with respect to $\boldsymbol{S}$, is called *admissible*.

A credulous semantics is that of a *preferred extension*, which is an admissible set that is maximal wrt. set inclusion. Finally, an admissible set $\boldsymbol{S}$ is said to be a *stable extension*, if it attacks all arguments in $\boldsymbol{A} \setminus \boldsymbol{S}$. Clearly, a stable extension is a preferred extension as well.

In general it is hard to compute a preferred extension [14], but in [15] we have adapted a technique of [16] to the problem of enumerating preferred extensions of argumentation systems of [12]: Given $\mathcal{A} \equiv (\boldsymbol{A}, \triangleright)$, we define an $\mathcal{A}$*-candidate* as a triple $(\boldsymbol{I}, \boldsymbol{O}, \boldsymbol{U} \equiv \boldsymbol{A} \setminus (\boldsymbol{I} \cup \boldsymbol{O}))$ where

- $\boldsymbol{I} \cap \boldsymbol{O} = \varnothing$,
- every argument that is attacked by $\boldsymbol{I}$ is in $\boldsymbol{O}$, and
- every argument A, for which there exists $\boldsymbol{S} \subseteq \boldsymbol{I}$ and $B \in \boldsymbol{I}$, such that $\boldsymbol{S} \cup A \triangleright B$, is in $\boldsymbol{O}$.

(Here $\boldsymbol{I}$ is supposed to capture the intuition of arguments that are *i*n the preferred extension, as opposed to *o*ut and *u*nassigned.)

Given an $\mathcal{A}$-candidate $\mathcal{C} \equiv (\boldsymbol{I}, \boldsymbol{O}, \boldsymbol{U})$ and an argument $A \in \boldsymbol{U}$ the triples $\mathcal{C} - A \equiv (\boldsymbol{I}_{-A}, \boldsymbol{O}_{-A}, \boldsymbol{U}_{-A} \equiv \boldsymbol{A} \setminus (\boldsymbol{I}_{-A} \cup \boldsymbol{O}_{-A}))$ and $\mathcal{C} + A \equiv (\boldsymbol{I}_{+A}, \boldsymbol{O}_{+A}, \boldsymbol{U}_{+A} \equiv \boldsymbol{A} \setminus (\boldsymbol{I}_{+A} \cup \boldsymbol{O}_{+A}))$ are given by:

$$\boldsymbol{I}_{-A} \equiv \boldsymbol{I}, \qquad \boldsymbol{O}_{-A} \equiv \boldsymbol{O} \cup A \qquad \boldsymbol{I}_{+A} \equiv \boldsymbol{I} \cup A, \text{ and } \quad \boldsymbol{O}_{+A} \equiv \boldsymbol{O} \cup \boldsymbol{\Delta}_{\mathcal{C}+A},$$

[4] To reduce clutter, we leave out { and } for singleton sets.

where $\boldsymbol{\Delta}_{\mathcal{C}+A}$ contains all arguments in $\boldsymbol{U} \setminus A$ which need to be in $\boldsymbol{O}_{+A}$ in order for $C + A$ to be a candidate. If A does not participate in a minimal attack on itself (which is the case for all arguments of the argumentation system we construct in this paper), then both $\mathcal{C} - A$ and $\mathcal{C} + A$ are $\mathcal{A}$-candidates themselves, and we can thus construct *candidate trees*, where each node is an $\mathcal{A}$-candidate: Each $\mathcal{A}$-candidate $\mathcal{C}$ has two children $\mathcal{C} - A$ and $\mathcal{C} + A$, for some arbitrary chosen A in $\boldsymbol{U}$, except those candidates where $\boldsymbol{U} = \varnothing$, which act as leaves in the tree. A candidate tree having candidate $\mathcal{C}$ as root, is called a $\mathcal{C}$*-tree*.

It can be proven that if $\boldsymbol{I}$ is a preferred extension of $\mathcal{A}$, then there is a leaf $(\boldsymbol{I}, \boldsymbol{O}, \varnothing)$ of any $(\varnothing, \varnothing, \boldsymbol{A})$-tree. Conversely, for any leaf $(\boldsymbol{I}, \boldsymbol{O}, \varnothing)$ in a $(\varnothing, \varnothing, \boldsymbol{A})$-tree, where $\boldsymbol{I}$ defends itself, $\boldsymbol{I}$ is admissible. It follows that, by constructing an arbitrary $(\varnothing, \varnothing, \boldsymbol{A})$-tree, all preferred extensions can be enumerated.

3. Encoding Graphs

For the agents to conclude on the best compromise $\mathcal{G}_*$, a formal language $\boldsymbol{L}$ for expressing graphs and properties of graphs must be defined. For efficiency reasons we aim to make this language finite and as small as possible, while ensuring that it is still sufficiently powerful to describe any graph and its membership status in $\mathfrak{C}$.

First, we introduce a small language $\boldsymbol{L}^g$ for encoding graphs:

Definition 1 (Simple Graph Language)**.** *The language $\boldsymbol{L}^g$ is the set containing the sentences* `Arc(X,Y)`*,* `Arc(Y,X)`*,* `Link(X,Y)`*, and* `NonAdjacent(X,Y)` *iff X and Y ($X \rightsquigarrow Y$) are distinct variables.*

A *graph knowledge base* is a set $\boldsymbol{\Sigma}^g \subseteq \boldsymbol{L}^g$. Further:

Definition 2 (Consistent Graph Knowledgebases)**.** *Given a graph knowledge base $\boldsymbol{\Sigma}^g$, if it holds that for all pairs of variables X and Y, where $X \rightsquigarrow Y$, a maximum of one of* `Arc(X,Y)`*,* `Arc(Y,X)`*,* `Link(X,Y)`*, and* `NonAdjacent(X,Y)` *is in $\boldsymbol{\Sigma}^g$, then we call $\boldsymbol{\Sigma}^g$ a* consistent graph knowledge base (CGK)*.*

The *graph encoded by* a CGK $\boldsymbol{\Sigma}^g$ is the graph $\mathcal{G}[\boldsymbol{\Sigma}^g]$ resulting from starting with the graph with no edges, and then for any two nodes X and Y ($X \rightsquigarrow Y$) adding an arc from X to Y if `Arc(X,Y)` is in $\boldsymbol{\Sigma}^g$, an arc from Y to X if `Arc(Y,X)` is in $\boldsymbol{\Sigma}^g$, or an undirected edge if `Link(X,Y)` is in $\boldsymbol{\Sigma}^g$. It is easy to see that graph encoded by a CGK is well-defined. Furthermore, given a graph $\mathcal{G}$ there exists at least one CGK, for which $\mathcal{G}$ is the encoded graph.

We thus have that any graph can be efficiently encoded as a CGK, and Definition 2 allows us to distinguish the graph knowledge bases, which can be interpreted as graphs, from those that cannot. Next, we extend $\boldsymbol{L}^g$ into a language powerful enough for building a reasoning engine about graphs and their membership status of $\mathfrak{C}$ on top:

Definition 3 (Graph Language)**.** *The graph language $\boldsymbol{L}$ is the set containing all sentences in $\boldsymbol{L}^g$ and*

- `ArcNotAllowed(X,Y)`*,*
- `DirectedPath(X,Y)`*,*

- *`UndirectedPath(X,Y)`,*
- *`UndirectedPath(X,Y)Excluding(Z,W)`,*
- *$\neg$`DirectedPath(X,Y)`,*
- *$\neg$`UndirectedPath(X,Y)`, and*
- *$\neg$`UndirectedPath(X,Y)Excluding(Z,W)`,*

for any choice of distinct variables[5] X, Y, Z, *and* W *(*$Z \rightsquigarrow W$*). Sentences of the last six kinds will be referred to as* path *sentences.*

The sentences just introduced are supposed to be used as descriptors of attributes of the graphs encoded by CGKs: `ArcNotAllowed(X,Y)` states that an arc from X to Y would not be strongly protected[6], which is required of all arcs in a completed pattern, while the remaining sentences should be self-explanatory (e.g. $\neg$`UndirectedPath-(X,Y)Excluding(Z,W)` states that there is no undirected path between X and Y, or that any such path necessarily contains either Z or W).

As $\boldsymbol{L}^g$ is a subset of $\boldsymbol{L}$, it follows that a graph knowledge base is a set of sentences in $\boldsymbol{L}$ as well, and given a set $\boldsymbol{\Sigma}$ of sentences of $\boldsymbol{L}$, we denote by $\boldsymbol{\Sigma}^g$ the set $\boldsymbol{\Sigma} \cap \boldsymbol{L}^g$. In particular Definition 2 is still applicable.

4. Graph Argumentation System

Building on the language $\boldsymbol{L}$ introduced above, we define an argumentation system for distinguishing completed patterns that could be compromises for the agents. The system that we construct enjoys the properties that a graph is a member of $\mathfrak{C}$ iff there is a preferred extension of the system which encodes this graph.

Definition 4 (Graph Argumentation System)**.** *The* graph argumentation system $\mathcal{A}^g$ *is the tuple* $(\boldsymbol{L}, \triangleright^g \subseteq (2^{\boldsymbol{L}} \times \boldsymbol{L}))$*, where* $\triangleright^g$ *is defined as follows (*`{A,B}` *is short-hand for any one of* `(A,B)` *and* `(B,A)`*):*

1. `Arc(X,Y)`$\triangleright^g$ `Arc(Y,X)`
2. `Arc(X,Y)`$\triangleright^g$ `Link{X,Y}`
3. `Arc(X,Y)`$\triangleright^g$ `NonAdjacent{X,Y}`
4. `Link(X,Y)`$\triangleright^g$ `Arc{X,Y}`
5. `Link(X,Y)`$\triangleright^g$ `NonAdjacent{X,Y}`
6. `NonAdjacent(X,Y)`$\triangleright^g$ `Arc{X,Y}`
7. `NonAdjacent(X,Y)`$\triangleright^g$ `Link{X,Y}`
8. $\neg$`DirectedPath(X,Y)`$\triangleright^g$ `DirectedPath(X,Y)`
9. $\neg$`UndirectedPath(X,Y)`$\triangleright^g$ `UndirectedPath(X,Y)`
10. $\neg$`UndirectedPath(X,Y)Excluding(Z,W)`$\triangleright^g$ `UndirectedPath(X,Y)Excluding(Z,W)`
11. `Arc(X,Y)`$\triangleright^g$ $\neg$`DirectedPath(X,Y)`
12. `Link(X,Y)`$\triangleright^g$ $\neg$`UndirectedPath{X,Y}`
13. `Link(X,Y)`$\triangleright^g$ $\neg$`UndirectedPath{X,Y}Excluding(Z,W)`
14. $\{$`DirectedPath(X,Y)`, `DirectedPath(Y,Z)`$\}\triangleright^g$ $\neg$`DirectedPath(X,Z)`
15. $\{$`DirectedPath(X,Y)`, `UndirectedPath{Y,Z}`$\}\triangleright^g$ $\neg$`DirectedPath(X,Z)`

[5] Throughout the text we assume that the implicit set of variables $\boldsymbol{V}$ has at least five members. This assumption can easily be lifted, albeit with a more complex notation to follow.

[6] An arc is strongly protected in a graph $\mathcal{G}$ if it occurs in one of four specific sub-graphs of $\mathcal{G}$. See [10] for details.

16. $\{$`UndirectedPath{X,Y}`, `DirectedPath(Y,Z)`$\}\triangleright^g \neg$`DirectedPath(X,Z)`
17. $\{$`UndirectedPath{X,Y}`, `UndirectedPath{Y,Z}`$\}\triangleright^g \neg$`UndirectedPath{X,Z}`
18. $\{$`UndirectedPath{X,Y}Excluding(Z,W)`, `UndirectedPath{Y,U}Excluding(Z,W)`$\}\triangleright^g \neg$`UndirectedPath{X,U}Excluding(Z,W)`
19. `DirectedPath(X,Y)`$\triangleright^g$ `Arc(Y,X)`
20. `DirectedPath(X,Y)`$\triangleright^g$ `Link{X,Y}`
21. `UndirectedPath{X,Y}`$\triangleright^g$ `Arc(X,Y)`
22. $\{$`UndirectedPath{X,Y}Excluding(W,Z)`, `Link{X,W}`, `Link{Y,Z}`, `NonAdjacent{X,Z}`, `NonAdjacent{Y,W}`$\}\triangleright^g$ `Link{W,Z}`
23. $\{$`Arc(X,Y)`, `NonAdjacent{X,Z}`$\}\triangleright^g$ `Link{Y,Z}`
24. `ArcNotAllowed(X,Y)`$\triangleright^g$ `Arc(X,Y)`
25. $\{$`Arc(Z,X)`, `NonAdjacent{Z,Y}`$\}\triangleright^g$ `ArcNotAllowed(X,Y)`
26. $\{$`Arc(Z,Y)`, `NonAdjacent{Z,X}`$\}\triangleright^g$ `ArcNotAllowed(X,Y)`
27. $\{$`Arc(X,Z)`, `Arc(Z,Y)`$\}\triangleright^g$ `ArcNotAllowed(X,Y)`
28. $\{$`Link{X,Z}`, `Arc(Z,Y)`, `Link{X,W}`, `Arc(W,Y)`, `NonAdjacent{Z,W}`$\}\triangleright^g$ `ArcNotAllowed(X,Y)`

for all choices of distinct variables X, Y, Z, W, and U where the sentences obtained are in $\boldsymbol{L}$.

Loosely speaking, if $\mathbf{\Sigma}$ is a preferred extension of $\mathcal{A}^g$, then Bullets 1–7 ensure that $\mathbf{\Sigma}^g$ is a CGK; Bullets 8–18 make sure that the path sentences in $\mathbf{\Sigma} \setminus \mathbf{\Sigma}^g$ are correct wrt. $\mathcal{G}[\mathbf{\Sigma}^g]$; Bullets 19–28 ensure that $\mathcal{G}[\mathbf{\Sigma}^g]$ is a complete pattern, cf. [10]. More precisely we have:

Lemma 1. *Let $\mathbf{\Sigma}$ be conflict free wrt. $\mathcal{A}^g$. Then $\mathbf{\Sigma}^g$ is a CGK.*

Theorem 1. *Let $\mathbf{\Sigma}$ be a preferred extension of $\mathcal{A}^g$. Then $\mathcal{G}[\mathbf{\Sigma}^g]$ is in $\mathfrak{C}$.*

Theorem 2. *If $\mathcal{G}$ is in $\mathfrak{C}$, then there is a stable extension $\mathbf{\Sigma}$ of $\mathcal{A}^g$, such that $\mathcal{G}[\mathbf{\Sigma}^g] = \mathcal{G}$.*

These results are important since they guarantee that agents arguing under the restrictions specified by $\mathcal{A}^g$ can be sure that their result is a completed pattern and that they are not restricted from agreeing on any model a priori by the relations of $\mathcal{A}^g$. However, checking whether a set of arguments constitute a preferred extension is complex. It involves checks for both admissibility and maximality. We therefore state a result that yields a computationally efficient way of testing whether an admissible set of arguments of $\mathcal{A}^g$ is a preferred extension.

Theorem 3. *Let $\mathbf{\Sigma}$ be a preferred extension of $\mathcal{A}^g$. Then $\mathbf{\Sigma}$ is a stable extension.*

For proofs of all results and further elaborations, see [8].

5. Fusing Agoras

We now address the problem of having agents agree on a preferred extension of $\mathcal{A}^g$, given that each of them has its own prior beliefs, as expressed by the compromise score function s_i, and that each know the combination function c. There has not been a lot of work done in dialectics for more than two agents, where the simple proponent/opponent dualism does not suffice. The solution that we propose here is inspired by the Risk Agoras of [17] and [18] and the traditional blackboard architecture of MAS of cooperating agents, without being an actual instantiation of any of them. We construct a *fusing agora*, which is a framework in which the agents can debate. The agora has the property that, if agents are allowed to run the debate to conclusion, they end up with the best possible compromise according to their joint compromise score, and that throughout the debate they maintain a compromise, which improves as the debate progresses.

In the agora we shall take a $\mathcal{A}^g$-candidate $(\boldsymbol{I}, \boldsymbol{O}, \boldsymbol{U})$ as a unique representatives of a partial compromise $(\boldsymbol{I}, \boldsymbol{O})$. This is possible since $\boldsymbol{I}$ and $\boldsymbol{O}$ are subsets of $\boldsymbol{L}$, and thus both contain sentences describing aspects of a graph as required, and furthermore, $\boldsymbol{U}$ is by definition determined by $\boldsymbol{I}$ and $\boldsymbol{O}$. Any leaf candidate representing a preferred extension, then uniquely identifies a completed pattern, as guaranteed by Theorem 1. Agents can explore all compromises by examining a $(\varnothing, \varnothing, \boldsymbol{L})$-tree. Continually the agents take it upon themselves to explore sub-trees of this tree, and mark other sub-trees as open for investigation by other agents. The heuristics guiding the agents choices for exploration, in addition to $s_1, \ldots, s_k$ and c, then determine the outcome.

The agora can work in a variety of ways, depending on the behavior of the individual agents (a vanilla algorithm for an individual agent is provided later in Algorithm 1), but basically builds on two elements, which we assume each agent can access in a synchronized fashion only: A *pool of candidates* $\boldsymbol{C}$ and a *current best result* $\langle \boldsymbol{I}_*, s_{\boldsymbol{I}_*} \rangle$. $\boldsymbol{C}$ consists of pairs $\langle \mathcal{C}, s \rangle$, where $\mathcal{C}$ is an $\mathcal{A}^g$-candidate and thus a sub-tree of a $(\varnothing, \varnothing, \boldsymbol{L})$-tree, and s is a real value. $\boldsymbol{I}_*$ is either the empty set or a preferred extension of $\mathcal{A}^g$, and $s_{\boldsymbol{I}_*}$ is a real value. Initially, $\boldsymbol{C}$ contains only one element $\langle (\varnothing, \varnothing, \boldsymbol{L}), 0 \rangle$, and $\langle \boldsymbol{I}_*, s_{\boldsymbol{I}_*} \rangle$ is $\langle \varnothing, -\infty \rangle$.

Each agent i can utter the following locutions:

- *ExploreFromPool*$_i$($\langle \mathcal{C}, s \rangle$) — where $\langle \mathcal{C}, s \rangle$ is a member of $\boldsymbol{C}$. The meaning of the locution is that agent i takes upon itself the responsibility to investigate the preferred extensions in a $\mathcal{C}$-tree, assuming that $\mathcal{C}$ has a joint compromise score of s.
- *PutInPool*$_i$($\langle \mathcal{C}, s \rangle$) — where $\mathcal{C}$ is an $\mathcal{A}^g$-candidate, and s is a real value. The meaning of the locution is that agent i wants someone else to investigate the preferred extensions in a $\mathcal{C}$-tree, and that $\mathcal{C}$ has a joint compromise score of s.
- *UpdateBest*$_i$($\langle \boldsymbol{I}, s \rangle$) — where $\boldsymbol{I}$ is a subset of $\boldsymbol{L}$, and s is a real value. The meaning of the locution is that agent i has identified a preferred extension $\boldsymbol{I}$ with a joint compromise score s higher than $s_{\boldsymbol{I}_*}$.
- *AskOpinion*$_i$($\mathcal{C}_1, \mathcal{C}_2$) — where $\mathcal{C}_1$ and $\mathcal{C}_2$ are $\mathcal{A}^g$-candidates. The meaning of the locution is that agent i needs to know $s_j(\mathcal{C}_1, \mathcal{C}_2)$ for all other agents j.
- *StateOpinion*$_i$($\mathcal{C}_1, \mathcal{C}_2, s_\delta$) — where $\mathcal{C}_1$ and $\mathcal{C}_2$ are $\mathcal{A}^g$-candidates, and s_δ is a real value. The meaning of the locution is that $s_i(\mathcal{C}_1, \mathcal{C}_2)$ is s_δ.

The rules governing which locutions individual agents can utter, as well as their effects, we present as a set of pre and post conditions:

- *ExploreFromPool*$_i$($\langle \mathcal{C}, s \rangle$)
  - ∗ Pre: $\langle \mathcal{C}, s \rangle$ is in $\boldsymbol{C}$.
  - ∗ Post: $\langle \mathcal{C}, s \rangle$ is removed from $\boldsymbol{C}$
- *PutInPool*$_i$($\langle \mathcal{C}, s \rangle$)
  - ∗ Pre: There is no $\langle \mathcal{C}', s' \rangle$ in $\boldsymbol{C}$ such that $\mathcal{C}$ is a sub-tree of some $\mathcal{C}'$-tree.
  - ∗ Post: $\langle \mathcal{C}, s \rangle$ is in $\boldsymbol{C}$.
- *UpdateBest*$_i$($\langle \boldsymbol{I}, s \rangle$)

- Pre: $s > s_{\boldsymbol{I}_*}$.
- Post: $\langle \boldsymbol{I}_*, s_{\boldsymbol{I}_*} \rangle$ is set to $\langle \boldsymbol{I}, s \rangle$.

Locutions *AskOpinion*$_i$() and *StateOpinion*$_i$() have no pre or post conditions attached.

Algorithm 1 *Vanilla algorithm for agent i*

$\langle \mathcal{C}, s \rangle \leftarrow$ SELECTCANDIDATE($\boldsymbol{C}$)
ExploreFromPool$_i(\langle \mathcal{C}, s \rangle)$
$\mathcal{C}' \triangleq (\boldsymbol{I}', \boldsymbol{O}', \boldsymbol{U}') \leftarrow$ PRUNE($\mathcal{C}$)
if $\boldsymbol{U}' = \varnothing$ **then**
    if PREFERREDEXTENSION($\boldsymbol{I}'$) **then**
        AskOpinion$_i(\mathcal{C}, \mathcal{C}')$
        $s_i \leftarrow s_i(\mathcal{C}, \mathcal{C}')$
        wait for *StateOpinion*$_j(\mathcal{C}, \mathcal{C}', s_j) \forall j \neq i$
        $s' \leftarrow c(s_1, \ldots, s_k) + s$
        if $s' > s_{\boldsymbol{I}_*}$ **then**
            UpdateBest$_i(\langle \mathcal{C}', s' \rangle)$
    go to 1
else
    $A \leftarrow$ SELECTARGUMENT($\mathcal{C}'$)
    AskOpinion$_i(\mathcal{C}, \mathcal{C}' + A)$
    AskOpinion$_i(\mathcal{C}, \mathcal{C}' - A)$
    $s_i^+ \leftarrow s_i(\mathcal{C}, \mathcal{C}' + A)$
    $s_i^- \leftarrow s_i(\mathcal{C}, \mathcal{C}' - A)$
    wait for *StateOpinion*$_j(\mathcal{C}, \mathcal{C}' + A, s_j^+)$ **and** *StateOpinion*$_j(\mathcal{C}, \mathcal{C}' - A, s_j^-) \forall j \neq i$
    $s^+ \leftarrow c(s_1^+, \ldots, s_k^+)$
    $s^- \leftarrow c(s_1^-, \ldots, s_k^-)$
    if $s^+ > s^-$ **then**
        PutInPool$_i(\langle \mathcal{C}' - A, s + s^- \rangle)$
        $\mathcal{C} \leftarrow \mathcal{C}' + A$
        $s \leftarrow s + s^+$
    else
        PutInPool$_i(\langle \mathcal{C}' + A, s + s^+ \rangle)$
        $\mathcal{C} \leftarrow \mathcal{C}' - A$
        $s \leftarrow s + s^-$
    go to 3

The basic algorithm in Algorithm 1 corresponds to an exhaustive search, if it is followed by all agents. The search is gradual in two senses: The longer the search goes on, the average candidate in $\boldsymbol{C}$ will have more elements in its $\boldsymbol{I}$ and $\boldsymbol{O}$ sets, and thus be closer to describing a full compromise, and the current compromise held in $\boldsymbol{I}_*$ will have an increasingly higher score. Of course, in order for the search to be a success, each agent i would also need to keep an eye out for *AskOpinion*$_j(\cdot)$'s uttered by other agents, and reply to these with *StateOpinion*$_i(\cdot)$. It is relatively easy to verify that agents using Algorithm 1 are uttering locutions in accordance with the pre and post conditions of the fusing agora.

Algorithm 1 calls a number of functions, which we only describe informally:

PRUNE($\mathcal{C} \equiv (\boldsymbol{I}, \boldsymbol{O}, \boldsymbol{U})$) uses pruning rules to investigate whether there is an argument A in $\boldsymbol{U}$ such that either $\mathcal{C} + A$ or $\mathcal{C} - A$ contains no leaves with preferred extensions. If this is the case, the method invokes itself recursively on the sub-tree that did not get pruned away, until no further branches can be pruned. Some general pruning rules are given in [15], and more can be established for the specific case of $\mathcal{A}^g$.

SELECTCANDIDATE($\boldsymbol{C}$) picks a promising candidate from $\boldsymbol{C}$. A promising candidate could be one with a high score annotated, since these encode good partial compromises, or candidates with small $\boldsymbol{U}$ sets, as these represent partial compromises that are nearly complete. If all agents use the same criteria for picking promising candidates, this selection can be sped up by implementing the pool as a sorted list. SELECTCANDIDATE($\cdot$) is one of the areas where heuristics limiting the search space can be implemented. For instance, it makes sense to allow agents to abstain from exploring the sub-tree rooted at a candidate if it cannot contain compromises that are consistent with their own BN. This would mean that in cases where agents agree on all or most of the aspects of $\mathcal{G}_*$ only few candidates would need to be explored.

PREFERREDEXTENSION($\boldsymbol{I}$) is a Boolean valued function that returns true if the conflict-free set $\boldsymbol{I}$ is a preferred extension of $\mathcal{A}^g$. The task of answering this is simplified by Theorem 3, as it states that $\boldsymbol{I}$ is a preferred extension iff $\boldsymbol{I}$ attacks each argument in $\boldsymbol{L} \setminus \boldsymbol{I}$.

SELECTARGUMENT($\mathcal{C} \equiv (\boldsymbol{I}, \boldsymbol{O}, \boldsymbol{U})$) simply selects an element A of $\boldsymbol{U}$. This selection can be based on the agent's own score increase going from $\mathcal{C}$ to $\mathcal{C} + A$ or $\mathcal{C} - A$, or it might involve negotiations or argumentation with other agents.

Of course, the debate in the agora can be stopped at any time, and $\mathcal{G}[\boldsymbol{I}_*^g]$ will then be the best compromise encountered so far, as it is only ever replaced by compromises having a higher joint compromise score.

It is worth stressing that Algorithm 1 is a vanilla algorithm, and that the agora is open for more aggressive behaviour. One such behaviour could be to have agents skip the asking for opinions part in Lines 14 to 22 for most additions of arguments (and basing the decision only on the agents own beliefs), and only ask when the agent itself is indifferent. Another behaviour could be to never perform Lines 23 and 27, which would correspond to a myopic greedy construction of the compromise. Alternatively, these two lines could be carried out only when the difference between s^+ and s^- is very small. We could even have setups where the agents show different behaviours, or where individual agents change behaviour during debate depending on their available resources and utility of a good compromise. Moreover, the agora does not require that agents wait for a candidate to be in the pool, before somebody can start exploring this candidate; so even when one agent is pursuing an aggressive strategy and fails to leave candidates for others to explore, other agents can still decide to explore these. The point is, that no matter what behaviour is requested, the basics of the agora and the agents remains the same.

6. Conclusion

We have introduced a problem which we believe is a challenging one for the argumentation community, due to its mix of complexity and conditional decomposability as well as its origin in conflicting knowledge bases. Our own solution enables agents to judge the possible compromises resulting from a partial compromise, by constructing a candidate tree rooted in this partial compromise, and the agora we have proposed ensures that such exploration can take place in a distributed fashion. One problem with the vanilla algorithm we have given, is that agents exploring a branch of a candidate-tree can end up putting a lot of candidates into the pool of candidates. The space requirements for storing the pool of candidates can be prohibitive, so it might be required that the candidates in the pool are defined in relation to each other, which imposes restrictions on which candidates an agent can choose to explore, as these are removed from the pool. Furthermore, it might be necessary to construct heuristics for thinning the pool of candidates. These issues, as well as finding good heuristics for selecting candidates to explore are challenging topics for future research.

References

[1] S. Saha and S. Sen. A Bayes net approach to argumentation based negotiation. In *ArgMAS 1*, pages 208–222. Springer, 2004.

[2] G. Vreeswijk. Bayesian inference and defeasible reasoning: suggestions for a unified framework. Working paper?, 1999.

[3] G. Vreeswijk. Argumentation in Bayesian belief networks. In *ArgMAS 1*, Lecture Notes in Artificial Intelligence, pages 111–129. Springer, 2004.

[4] I. Matzkevich and B. Abramson. The topological fusion of bayes nets. In *UAI 8*, pages 191–198. Morgan Kaufmann, 1992.

[5] J. Del Sagrado and S. Moral. Qualitative combination of bayesian networks. *International Journal of Intelligent Systems*, 18(2):237–249, 2003.

[6] M. R. and P. Domingos. Learning with knowledge from multiple experts. In *ICML 20*, pages 624–631. AAAI Press, 2003.

[7] P. M-R. II and U. Chajewska. Aggregating learned probabilistic beliefs. In *UAI 17*, pages 354–361. Morgan Kaufmann, 2001.

[8] S. H. Nielsen and S. Parsons. Fusing Bayesian networks using formal argumentation in multi-agent systems. Technical report, Aalborg University, 2006. http://www.cs.auc.dk/˜holbech/-tr0106NielsenParsons06.ps.

[9] T. Verma and J. Pearl. Equivalence and synthesis of causal models. In *UAI 6*, pages 220–227. Elsevier Science Publishing Company, Inc., 1990.

[10] S. A. Andersson, D. Madigan, and M. D. Perlman. A characterization of Markov equivalence classes for acyclic digraphs. *Annals of Statistics*, 25(2):505–541, 1997.

[11] H. Prakken and G. Vreeswijk. Logics for defeasible argumentation. In *Handbook of Philosophical Logic*. Kluwer Academic Publishers, 2000.

[12] S. H. Nielsen and S. Parsons. A generalization of Dung's abstract framework for argumentation. In *Proceedings of Third Workshop on Argumentation in Multi-agent Systems*. Springer, 2006.

[13] P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artificial Intelligence*, 77(2):321–358, 1995.

[14] Y. Dimopoulos, B. Nebel, and F. Toni. Preferred arguments are harder to compute than stable extension. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence*, pages 36–43. Morgan Kaufmann, 1999.

[15] S. H. Nielsen and S. Parsons. Computing preferred extensions for argumentation systems with sets of attacking arguments. In *COMMA06 Proceedings*. 2006.

[16] S. Doutre and J. Mengin. Preferred extensions of argumentation frameworks: Query, answering and computation. In *Proceedings of International Joint Conference on Automated Reasoning*, volume 2083, pages 272–288. Springer, 2001.

[17] P. McBurney and S. Parsons. Risk agoras: Dialectical argumentation for scientific reasoning. In *UAI 16*. Stanford, 2000.

[18] P. McBurney and S. Parsons. Chance discovery using dialectical argumentation. In *Workshop on Chance Discovery, JSAI 15*. Matsue, 2001.