Refining a Heuristic for Constructing Bayesian Networks from Structured Arguments

Remi Wieten[1(✉)], Floris Bex[1], Linda C. van der Gaag[1], Henry Prakken[1,2], and Silja Renooij[1]

[1] Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands
g.m.wieten@uu.nl

[2] Faculty of Law, University of Groningen, Groningen, The Netherlands

Abstract. Recently, a heuristic was proposed for constructing Bayesian networks (BNs) from structured arguments. This heuristic helps domain experts who are accustomed to argumentation to transform their reasoning into a BN and subsequently weigh their case evidence in a probabilistic manner. While the underlying undirected graph of the BN is automatically constructed by following the heuristic, the arc directions are to be set manually by a BN engineer in consultation with the domain expert. As the knowledge elicitation involved is known to be time-consuming, it is of value to (partly) automate this step. We propose a refinement of the heuristic to this end, which specifies the directions in which arcs are to be set given specific conditions on structured arguments.

Keywords: Bayesian Networks · Structured argumentation

1 Introduction

In recent years, efforts have been made to gain a better understanding of the relation between different normative frameworks for evidential reasoning, such as argumentative and probabilistic approaches [9]. Argumentative approaches are particularly suited for adversarial settings, where arguments for and against a specific conclusion are constructed from evidence. The inferences which are used to draw conclusions from evidence are generally defeasible, in that the conclusion of an argument does not universally hold given the evidence. Arguments can be attacked by other arguments; it can then be established which arguments are accepted and which are rejected. In current argumentative approaches, however, there is no emphasis on incorporating graded uncertainty.

In contrast, probabilistic approaches are well suited for handling graded uncertainty. In particular, Bayesian networks (BNs) [2,3] are powerful tools to this end. BNs are compact graphical models of joint probability distributions, which allow for evidence evaluation by calculating the probability of the truth

B. Verheij and M. Wiering (Eds.): BNAIC 2017, CCIS 823, pp. 32–45, 2018.
https://doi.org/10.1007/978-3-319-76892-2_3

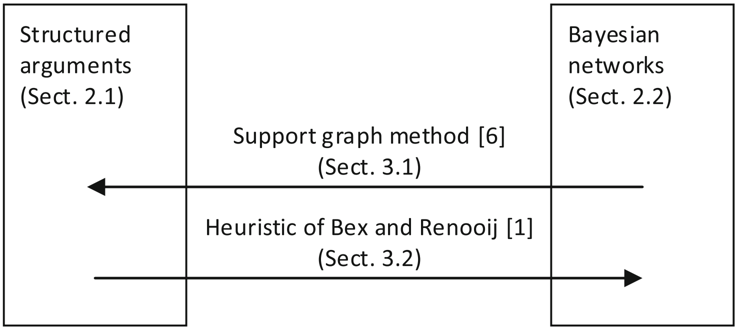

Fig. 1. Outline of Sects. 2 and 3 of this paper.

of a proposition of interest. However, BNs are generally difficult to construct; in fact, they are often constructed by modelers with the relevant mathematical background, called BN engineers, in consultation with a domain expert.

Recently, a heuristic for constructing BNs from structured arguments was proposed by Bex and Renooij [1]; in this paper, the heuristic will be referred to as the BR heuristic. The heuristic helps domain experts who are more accustomed to argumentation to transform their reasoning into a BN (cf. Fig. 1) and subsequently weigh their case evidence in a probabilistic manner. The focus of the BR heuristic lies on obtaining the graphical structure of the BN, called the BN graph, which captures the independence relations between the domain variables. While the underlying undirected graph, or skeleton, of the BN graph can be automatically constructed by following the BR heuristic, the heuristic prescribes that the arc directions should be set manually by a BN engineer in consultation with a domain expert. Although the heuristic further suggests that the commonly used notion of causality be taken as a guiding principle [3], the resulting graph still has to be verified and refined in terms of the independence relations it represents. This type of knowledge elicitation is known to be time-consuming [7], however, and moreover needs to be repeated for every adjustment to the original arguments. As a consequence, letting arc directions be set by a BN engineer is practically infeasible in investigative contexts such as police investigations, where evidence changes dynamically. It is, therefore, of value to investigate whether the process of setting arc directions can be (partly) automated.

Accordingly, in this paper we propose a refinement of the BR heuristic, which specifies the directions in which the arcs should be set in a BN graph under specific conditions on structured arguments. These conditions are identified by applying a method called the support graph method [6]. This method essentially works in the opposite direction of the BR heuristic, in that structured arguments are constructed from BNs (cf. Fig. 1). By applying the support graph method to BN graphs obtained with the BR heuristic, it is determined whether and under which conditions the original arguments are re-obtained. If the original arguments are not re-obtained from the thus constructed BN graph, it may be concluded that this graph represents the original arguments in a different, possibly incorrect, way. Our refinement of the BR heuristic now ensures that BN graphs from which the original arguments are not returned by the support graph method are not constructed.

The paper is structured as follows. Sections 2 and 3 provide some preliminaries on structured argumentation, BNs, the support graph method and the BR heuristic. In Sect. 4, our refinement to the BR heuristic is proposed, based on observations from applying the support graph method. In Sect. 5, our findings are summarized and possible directions for future research are discussed.

2 Preliminaries

In this section, structured argumentation and BNs are briefly reviewed.

2.1 Structured Argumentation

A simplified version of the ASPIC+ framework for structured argumentation [4] is assumed throughout this paper. Let $\mathcal{L}$ be a non-empty propositional literal language with the unary negation symbol $\neg$. Informally, $\mathcal{L}$ contains the basic elements which can be argued about. Given a knowledge base $\mathcal{K} \subseteq \mathcal{L}$ of premises, arguments are constructed by chaining inference rules. These rules are defined over $\mathcal{L}$ and are defeasible, in that the conclusion of a defeasible rule does not universally hold given the premises, in contrast with the strict inferences of classical logic. Let $\mathcal{R}$ be a set of defeasible inference rules of the form $d\colon \phi_1, \ldots, \phi_n \Rightarrow \phi$, where $\phi_1, \ldots, \phi_n$ and ϕ are meta-variables ranging over well-formed formulas in $\mathcal{L}$. An argument A is then either: (1) ϕ if $\phi \in \mathcal{K}$, where the conclusion of the argument A, denoted by $\textsc{Conc}(A)$, is equal to ϕ; or (2) $A_1, \ldots, A_n \Rightarrow \phi$ with $\phi \in \mathcal{L} \setminus \mathcal{K}$, where $A_1, \ldots, A_n$ are arguments such that there exists a rule $\textsc{Conc}(A_1), \ldots, \textsc{Conc}(A_n) \Rightarrow \phi$ in $\mathcal{R}$. In the first case, $\textsc{Conc}$(A) is an element from the knowledge base, while in the second case, $\textsc{Conc}$(A) follows by applying a defeasible rule to the conclusion(s) of arguments $A_1, \ldots, A_n$, which are called the immediate sub-arguments of A. Generally, a sub-argument of an argument A is either A itself or an argument that is (iteratively) used to construct A. The smallest set of finite arguments which can be constructed from $\mathcal{L}$, $\mathcal{K}$ and $\mathcal{R}$ is denoted by $\mathcal{A}$. An argument graph of $\mathcal{A}$ then graphically displays the arguments in $\mathcal{A}$ and their sub-arguments. Figure 3a shows an example of an argument graph.

The general ASPIC+ framework further includes the notion of attack. Informally, an argument in $\mathcal{A}$ is attacked on one of its non-premise sub-arguments by another argument in $\mathcal{A}$ with the opposite conclusion of that sub-argument. Due to space limitations, the focus of the current paper lies on argument structures without attack relations.

2.2 Bayesian Networks

BNs [3] are graphical probabilistic models which are being applied in many different fields, including medicine and law [2]. A BN is a compact representation of a joint probability distribution $\Pr(\mathbf{V})$ over a finite set of discrete random variables $\mathbf{V}$. The random variables are represented as nodes in a directed acyclic graph G, where each node[1] can take one of a number of mutually exclusive and exhaustive values; in this paper, we assume all nodes to be Boolean. A node A is a parent of another node B, called the child, in G if G contains an arc from A to B. The BN further includes, for each node, a conditional probability table, or CPT, given its parents; this table specifies the probabilities of the values of the node itself conditioned on the possible joint value combinations of its parents. A node is called instantiated iff it is fixed in a specific value. Given a set of instantiated nodes, conditional probability distributions over the other nodes in the network can be computed using probability calculus [3].

The BN graph captures the independence relations between its variables. Let a chain be defined as a simple path in the underlying undirected graph, or skeleton, of a BN graph. A node V is called a head-to-head node on a chain c if it has two incoming arcs on c. A chain c is blocked iff it includes a node V such that (1) V is an uninstantiated head-to-head node on c without instantiated descendants; (2) V is not a head-to-head node on c and is instantiated. In addition, instantiated end-points of the chain c, that is, instantiated nodes with at most one incoming or outgoing arc on c, serve to block the chain [5]. A chain is inactive if it is blocked; otherwise it is called active. Two nodes $A \neq B$ are called d-separated by a set of nodes $\mathbf{Z}$ if no active chains exist between A and B given instantiations of nodes in $\mathbf{Z}$. If two nodes are d-separated by $\mathbf{Z}$, then they are considered conditionally independent given $\mathbf{Z}$. We note that conditional independence thereby depends on the set of instantiated nodes [8].

An immorality in a BN graph is defined as a triple of nodes (A, B, C), where A and C are parents of B that are not directly connected by an arc. Two BNs are said to be Markov equivalent iff they share the same skeleton and immoralities. Markov equivalent networks constitute an equivalence class, for which Verma and Pearl [10] proved that any two elements represent the same independence relations over the variables involved. Arcs between nodes that are not involved in an immorality can thus be reversed without changing the represented independence relations as long as no new immoralities arise. Immoralities derive their importance from providing for intercausal reasoning [11]. Specifically, if the head-to-head node involved in an immorality is instantiated, an active chain arises between the parents of the node. These parents can be seen as different causes of the same effect modeled by the head-to-head node. If one of the causes is now observed, then the probability of the other cause being present as well can either increase, decrease or stay the same upon updating, depending on the probabilities in the CPT of the head-to-head node.

[1] The terms ‘node’ and ‘variable’ are used interchangeably.

3 Two Methods for Translating Between Structured Arguments and Bayesian Networks

In this section, the support graph method [6] and the BR heuristic [1] are reviewed; the support graph method is used to build structured arguments from BNs, while the BR heuristic is used to construct BN graphs from structured arguments.

3.1 The Support Graph Method

The support graph method, proposed by Timmer and colleagues [6], is a two-phase method for constructing argument structures from BNs. The method allows domain experts who are not familiar with BNs but are accustomed to argumentation to understand the knowledge and reasoning patterns captured by a BN. To this end, the method summarizes all reasoning chains from a set of evidence to a conclusion in a given BN.

In the first phase of the method, a directed graph called the support graph (SG) is constructed from a BN given a variable of interest V^*; in this SG, all reasoning chains in the BN ending in V^* are captured. The SG does not depend on specific instantiations, and can thus be re-used to build argument structures for different evidence. An SG is iteratively constructed, starting with a graph containing only V^*. New parents are added to existing nodes in the SG as new inference steps are identified in the BN. Three types of inference step are distinguished: (1) an inference step along an arc from a parent to a child; (2) an inference step along an arc from a child to a parent; and (3) an inference step between two parents in an immorality. The last type directly accommodates intercausal reasoning steps which occur between the parents of an immorality, and summarizes the inference from one parent of an immorality to another parent via the common child. In the constructed SG, V^* is the only node without children; every other node in the SG is an ancestor of V^*.

In the second phase of the support graph method, arguments are constructed from the SG for a given set of node instantiations. Given this evidence, the SG is pruned such that only paths remain that start in an instantiated node. From the thus pruned graph, arguments are constructed as follows. The logical language $\mathcal{L}$ is taken to consist of all literals which correspond to the values of the nodes in the BN; two literals $\phi, \psi \in \mathcal{L}$ negate each other iff ϕ and ψ correspond with the different values of the same node. Given the evidence, the knowledge base $\mathcal{K}$ consists of those literals in $\mathcal{L}$ that correspond with the values of the instantiated nodes. The defeasible rules in $\mathcal{R}$ are of the form $(N_1, o_1), \ldots, (N_k, o_k) \Rightarrow (N, o)$, where $N_1, \ldots, N_k$ are parents of the node N in the pruned SG and $o_1, \ldots, o_k, o$ are values of these nodes. From $\mathcal{L}$, $\mathcal{K}$, and $\mathcal{R}$, a set of arguments $\mathcal{A}$ is then constructed.

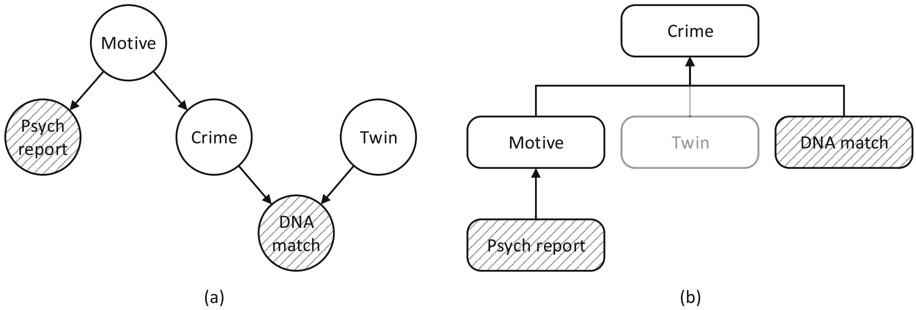

Fig. 2. A BN graph (a) and the corresponding SG for the variable of interest *Crime* (b); *Twin* is pruned from the SG as only *Psych report* and *DNA match* are instantiated.

Example 1. An example by Timmer and colleagues [6] from the legal domain is reviewed to demonstrate the support graph method. In the example, the BN graph from Fig. 2a[2] is constructed for a criminal case, in which we are interested in whether the suspect committed the crime, that is, whether $Crime = true$. Evidence for this possible conclusion would be the existence of a motive, which may be mentioned in a psychological report. A match between the suspect's DNA and DNA found at the crime scene would further support the proposition that the suspect committed the crime. This finding might also be explained, however, if the suspect had an identical twin. For the variable of interest *Crime*, the SG of Fig. 2b is obtained; the node *Twin* is directly added as a parent of *Crime*, as the triplet (*Crime*, *DNA match*, *Twin*) is an immorality in the BN graph. The literals in $\mathcal{L}$ are the possible values of all nodes in the BN graph, that is, $\mathcal{L}$ contains *crime*, $\neg$*crime*, *motive*, $\neg$*motive*, ... Now, if we assume that *Psych report* and *DNA match* are instantiated with the value *true* conform available evidence, and *Twin* is not instantiated, then the path starting at the node *Twin* is pruned from the SG. The knowledge base $\mathcal{K}$ then consists of *psych report* and *dna match*. Among the defeasible rules extracted from the pruned SG are $d_1\colon$ *psych report* $\Rightarrow$ *motive* and $d_2\colon$ *dna match*, *motive* $\Rightarrow$ *crime*. The arguments $A_1\colon$ *psych report*, $A_2\colon$ *dna match*, $A_3\colon A_1 \Rightarrow$ *motive*, and $A_4\colon A_2, A_3 \Rightarrow$ *crime* can then be constructed. Also the rules $d_3\colon$ *psych report* $\Rightarrow \neg$*motive* and $d_4\colon$ *dna match*, $\neg$*motive* $\Rightarrow \neg$*crime* are extracted from the SG, from which arguments $A_5\colon A_1 \Rightarrow \neg$*motive* and $A_6\colon A_2, A_5 \Rightarrow \neg$*crime* are constructed. These arguments have opposite conclusions of A_3 and A_4. □

It should be noted that, when using the support graph method, the reasons pro and con a given conclusion are not distributed over separate arguments, as is usual in argumentation, but are instead encapsulated in a single argument.

[2] In figures in this paper, circles are used in BN graphs, rectangles are used in argument graphs and rounded rectangles are used in SGs. Nodes and propositions corresponding to evidence are shaded. Capital letters are used for the nodes in BN graphs and SGs, and lowercase letters are used for propositions.

That is, all literals that are relevant for a specific proposition are taken as the premises of an argument for that proposition, which reflects the way in which Bayesian networks internally weigh all evidence.

For every argument that is returned from a BN by the support graph method, the method also returns an argument with the same 'structure' but with the opposite conclusion. Timmer and colleagues [6] employ a quantitative step to filter the set of arguments returned. As in the current paper the focus lies on the graphical structures of BNs and not on the modeled probability distribution, this quantitative step is not further discussed here.

3.2 The BR Heuristic for Constructing Bayesian Networks from Structured Arguments

Bex and Renooij [1] have proposed the BR heuristic for constructing BN graphs from structured arguments. This heuristic allows domain experts who are accustomed to argumentation to translate their reasoning expressed as arguments into a BN graph. This graph is then supplemented with CPTs to arrive at a fully specified BN for probabilistic inference over the original arguments. Focusing on argument structures in which no attack relations are present, from a given set of arguments $\mathcal{A}$ constructed from a logical language $\mathcal{L}$, knowledge base $\mathcal{K}$, and a set of defeasible rules $\mathcal{R}$, BN graphs are constructed as follows:

1. For every proposition $\phi \in \mathcal{L}$ used in $\mathcal{A}$, the BN graph includes a single node V such that $V = \mathit{true}$ corresponds to ϕ and $V = \mathit{false}$ corresponds to $\neg\phi$. For every $e \in \mathcal{K}$, the corresponding node is instantiated at the observed value.
2. For every defeasible rule $d\colon \phi_1, \ldots, \phi_n \Rightarrow \phi \in \mathcal{R}$ used in $\mathcal{A}$, a set of undirected edges between the node associated with ϕ and each of the nodes associated with $\phi_1, \ldots, \phi_n$ is created for inclusion in the BN graph.
3. The direction of the edges from the previous step is decided upon by a BN engineer in consultation with the domain expert, where a causal direction is chosen if possible, and an arbitrary direction otherwise. The resulting arcs are inserted in the BN graph.
4. The BN engineer verifies that the graph is acyclic and that all chains that should be active in the graph indeed are; if the graph does not yet exhibit these properties, appropriate arcs are removed or reversed, once more in consultation with the domain expert.

Example 2. A simple example is introduced to demonstrate the BR heuristic. The logical language, knowledge base and defeasible rules involved are $\mathcal{L} = \{p, \neg p, q, \neg q, r, \neg r\}$, $\mathcal{K} = \{p\}$ and $\mathcal{R} = \{p \Rightarrow q; q \Rightarrow r\}$. The constructed arguments are $\mathcal{A} = \{A_1\colon p;\ A_2\colon A_1 \Rightarrow q;\ A_3\colon A_2 \Rightarrow r\}$; the argument graph of $\mathcal{A}$ is depicted in Fig. 3a. Following steps 1 and 2 of the BR heuristic, the skeleton of the BN graph corresponding to this argument structure consists of nodes P, Q and R, with undirected edges between P and Q and between Q and R. Following step 3, one of the BN graphs of Fig. 3b–e is obtained, depending on how the arc directions are set. □

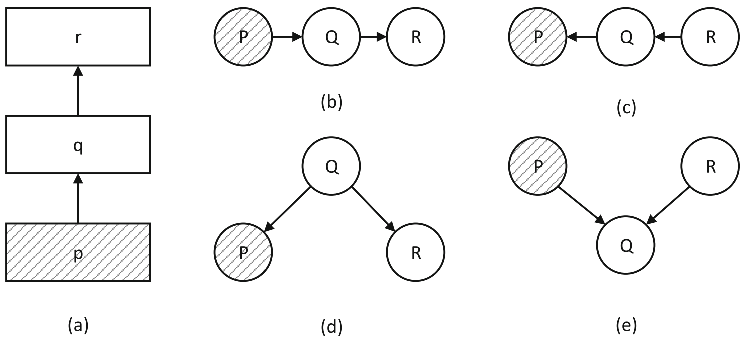

Fig. 3. An argument graph with arguments from p to r via q (a); the four corresponding BN graphs which can be constructed by following the BR heuristic (b–e).

For a given set of arguments $\mathcal{A}$, the skeleton of the BN graph is automatically constructed by following the first two steps of the BR heuristic. Step 3 then prescribes that the directions of the arcs should be set manually by a BN engineer in consultation with the domain expert, using the notion of causality as a guiding principle (see also [3]). For example, if the domain expert indicates for a defeasible rule $d\colon p \Rightarrow q$ that p is a typical cause of q, then the arc is set from node P to node Q. Since immoralities can result from following this guiding principle, the independence relations in the constructed BN graph should be verified manually, as prescribed by step 4 of the BR heuristic. This type of knowledge elicitation and verification is known to be a time-consuming and error-prone process in general [7]. Especially for larger or more densely connected BN graphs, it quickly becomes infeasible to verify all independence relations manually, as all possible chains for all possible combinations of instantiated variables need to be investigated. Moreover, the elicitation and verification needs to be repeated for every adjustment to the original argument graph. As this step is practically infeasible in investigative contexts, such as police investigations, in which the evidence for a case changes dynamically, the arc directions are preferably set (semi-)automatically.

4 Refining the BR Heuristic

We propose a refinement of step 3 of the BR heuristic, which specifies the directions in which arcs should be set in a BN graph under specific conditions on structured arguments. These conditions are identified from applying the support graph method. To this end, the arguments to which the BR heuristic is applied are compared to the arguments returned by the support graph method when applied to a BN graph constructed by steps 1–3 of the BR heuristic. In order to apply the support graph method, a variable of interest has to be chosen. In this paper, we assume that there is a single ultimate conclusion in the input argument graph, that is, a single argument that is not an immediate sub-argument of another argument. The node corresponding to this ultimate conclusion is taken as the node of interest. We further assume that the input arguments for the BR heuristic are linked, in the sense that all premises relevant for a conclusion are encapsulated in a single argument; Fig. 5a shows an example of an argument graph with linked arguments only. Linked argument graphs are similar to the type of argument graphs that are returned by the support graph method.

When applying the support graph method to a BN graph constructed by steps 1–3 of the BR heuristic, a set of arguments is returned. This set may be different from the set of arguments that was used as input for the heuristic. As measures for the differences found, we distinguish between recall and precision, which for a given BN graph respectively measure the proportion of original arguments returned and the proportion of additional arguments returned. Formally, let $\mathcal{A}$ be the set of input arguments for the BR heuristic, let $\mathbf{B}$ be a BN graph constructed from $\mathcal{A}$ by steps 1–3 of the heuristic, and let $\mathcal{A}'$ be the set of arguments returned from $\mathbf{B}$ by the support graph method. We define the recall and precision of $\mathbf{B}$ as follows:

- $\mathit{Recall}(\mathbf{B}) = |\mathcal{A} \cap \mathcal{A}'|/|\mathcal{A}|$
- $\mathit{Precision}(\mathbf{B}) = |\mathcal{A} \cap \mathcal{A}'|/|\mathcal{A}'|$

where $\mathbf{B}$ has maximum recall and precision if these fractions are equal to 1.

In Sect. 4.1, we propose a refinement of the third step of the BR heuristic, which serves to increase the recall of constructed BN graphs. In Sect. 4.2, we address precision. As argued before, Timmer and colleagues [6] propose a quantitative step for filtering the set of arguments returned by the support graph method, which suggests that for improving the precision of constructed BNs, the CPTs need to be taken into account. As in this paper, the focus lies on the graphical structure of a BN, we propose a further refinement of the third step of the BR heuristic based on graphical considerations only.

4.1 Refining the BR Heuristic to Improve Recall

To illustrate how the BR heuristic can be refined such that BN graphs with higher recall are constructed, we revisit Example 2 from Sect. 3.2. By applying steps 1–3 of the heuristic to the argument graph of Fig. 3a, four possible BN graphs over the nodes P, Q and R were constructed, as shown in Figs. 3b–e. These graphs fall into two Markov equivalence classes; the first class consists of the BN graphs of Figs. 3b–d, and the second class consists of the graph of Fig. 3e. Timmer and colleagues [6] proved that for two Markov equivalent BNs and the same node of interest, the same SG is obtained. By applying the support graph method for the node of interest R, we now show that the recall of the original arguments from the BN graph in the second equivalence class is lower than that of the BN graphs in the first class. Since the logical language and knowledge base of the argument structure returned by the support graph method are derived from the BN skeleton, $\mathcal{L}'$[3] $= \{p, \neg p, q, \neg q, r, \neg r\}$ and $\mathcal{K}' = \{p\}$ are the same for all four BN graphs. For the

[3] The prime symbol is used to denote objects which result from applying the support graph method.

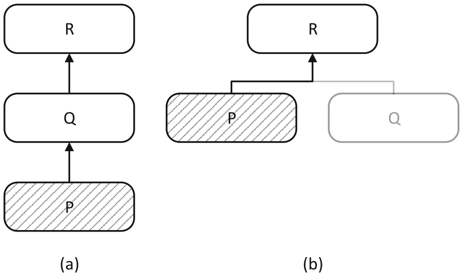

Fig. 4. The (pruned) SG obtained from the BN graphs of Figs. 3b–d (a), and the SG obtained from the BN graph of Fig. 3e (b), where Q is pruned as only P is instantiated.

graphs in the first equivalence class, the SG of Fig. 4a is obtained. The defeasible rules of the returned argument structure correspond to the arcs of this SG, that is, $\mathcal{R}' = \{p \Rightarrow q; p \Rightarrow \neg q; \neg p \Rightarrow q; \neg p \Rightarrow \neg q; q \Rightarrow r; q \Rightarrow \neg r; \neg q \Rightarrow r; \neg q \Rightarrow \neg r\}$. As $\mathcal{L} \subseteq \mathcal{L}'$, $\mathcal{K} = \mathcal{K}'$ and $\mathcal{R} \subseteq \mathcal{R}'$, all original arguments $A_1, A_2, A_3 \in \mathcal{A}$ are re-obtained from the SG. Therefore, the BN graphs of Figs. 3b–d have maximal recall.

For the BN graph in the second equivalence class, the SG of Fig. 4b is constructed. In this SG, node P is a direct parent of R and not of Q, as (P, Q, R) is an immorality. We recall that an SG is meant for constructing arguments for different sets of evidence. In the example, where just P is instantiated, node Q is pruned from the SG. The defeasible rules corresponding to this pruned SG are $\mathcal{R}' = \{p \Rightarrow r; p \Rightarrow \neg r; \neg p \Rightarrow r; \neg p \Rightarrow \neg r\}$ and the arguments which can be constructed are $A_1 : p$, $A_2' : A_1 \Rightarrow r$ and $A_2'' : A_1 \Rightarrow \neg r$. Timmer and colleagues [6] employ a quantitative step using the CPTs from the original BN to filter the set of constructed arguments; by this step, arguments A_2' and A_2'' are filtered out, as P and R are independent given that Q is not instantiated. The original arguments A_2 are A_3 are not returned by the support graph method. The recall of the BN graph from Fig. 3e is $\frac{1}{3}$, which is lower than that of the BN graphs in the first equivalence class. It therefore seems desirable to prohibit construction of this BN graph when using the BR heuristic.

Generalizing from the example, let $A_1, \ldots, A_n \in \mathcal{A}$, where A_i is an immediate sub-argument of A_{i+1} for all $i \in \{1, \ldots, n-1\}$, let $\textsc{Conc}(A_i) = p_i$, $p_1 \in \mathcal{K}$, and let p_n be the ultimate conclusion of the argument graph of $\mathcal{A}$. Further assume that no immorality (P_{i-1}, P_i, P_{i+1}) is formed for $i \in \{2, \ldots, n-1\}$ by steps 1–3 of the BR heuristic. As no immoralities (P_{i-1}, P_i, P_{i+1}) are present for $i \in \{2, \ldots, n-1\}$, upon constructing the SG for the node of interest P_n parents are added iteratively, that is, P_{n-1} is added as a parent of P_n, ..., P_1 is added as a parent of P_2. As P_1 corresponds to an instantiated variable, the path starting in P_1 is not pruned from the SG. The support graph method, therefore, returns the arguments $A_1, \ldots, A_n$, and the recall is maximal. On the other hand, if for a given $i \in \{2, \ldots, n-1\}$ an immorality (P_{i-1}, P_i, P_{i+1}) would be formed by steps 1–3 of the BR heuristic, then an SG would result in which P_{i+1} is an ancestor of P_n. As P_{i-1} is directly added as a parent of P_{i+1}, the argument A_i would not be returned, and the recall would not be maximal.

Based on the above observations, the following refinement of step 3 of the BR heuristic is proposed:

3′. Let $A_1, \ldots, A_n \in \mathcal{A}$, where A_i is an immediate sub-argument of A_{i+1} for any $i \in \{1, \ldots, n-1\}$ and where $\text{Conc}(A_i) = p_i$. Then, the directions of the arcs are set such that no immoralities (P_{i-1}, P_i, P_{i+1}) are formed for any $i \in \{2, \ldots, n-1\}$. Taking this constraint into account, the directions of the (remaining) arcs are set by a BN engineer in consultation with the domain expert, where a causal direction is chosen if possible.

4.2 A Further Refinement of the BR Heuristic

While in the previous section, simple chains in an argument structure were shown to be best translated in the BN graph by a chain without any immoralities, we now focus on argument structures that do enforce immoralities in the BN graph and propose a further refinement of the refined third step of the heuristic.

Example 3. We consider the linked argument graph of Fig. 5a. The logical language, knowledge base and defeasible rules involved are $\mathcal{L} = \{p, \neg p, q, \neg q, r, \neg r, s, \neg s, t, \neg t\}, \mathcal{K} = \{p, q\}$, and $\mathcal{R} = \{p \Rightarrow r; p, q \Rightarrow s; r, s \Rightarrow t\}$; the constructed arguments are $\mathcal{A} = \{A_1 \colon p; A_2 \colon A_1 \Rightarrow r; A_3 \colon q; A_4 \colon A_1, A_3 \Rightarrow s; A_5 \colon A_2, A_4 \Rightarrow t\}$. Steps 1 and 2 of the BR heuristic result in the BN skeleton of Fig. 5b. In order to

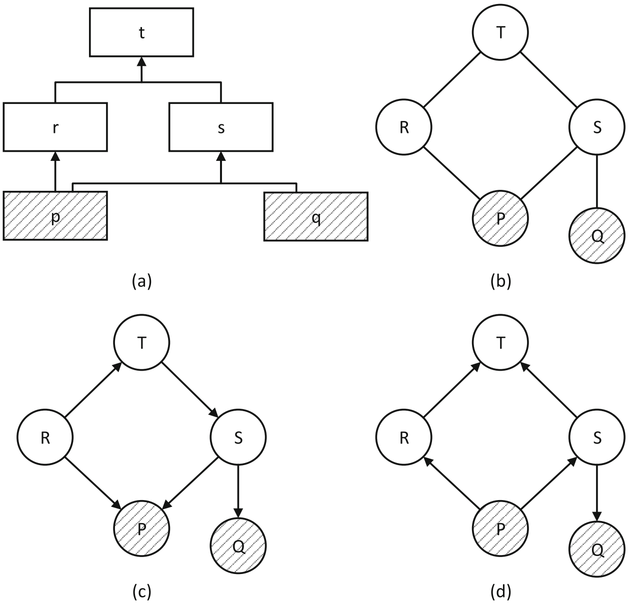

Fig. 5. An argument graph (a) and the corresponding BN skeleton that is constructed by the BR heuristic (b); a corresponding BN graph with the immorality (R, P, S) (c) and a BN graph with the immorality (R, T, S) (d).

obtain an acyclic directed graph from this skeleton, at least one immorality has to be created in the subgraph induced by the nodes P, R, S and T.

According to the refined third step of the BR heuristic, an immorality (T, R, P) should not be formed, as $A_1\colon p$ is an immediate sub-argument of $A_2\colon A_1 \Rightarrow r$, which in turn is an immediate sub-argument of $A_5\colon A_2, A_4 \Rightarrow t$. Similarly, the immorality (T, S, P) should not be formed. Now, the equivalence class of BN graphs is considered which includes just the immorality (R, P, S); the BN graph depicted in Fig. 5c is an element of this class. With T as the node of interest, the SG of Fig. 6 is obtained from this graph. The logical language and knowledge base corresponding to this SG are $\mathcal{L}' = \{p, \neg p, q, \neg q, r, \neg r, s, \neg s, t, \neg t\}$ and $\mathcal{K}' = \{p, q\}$, matching those of the original argument graph. The set of defeasible rules $\mathcal{R}$ corresponding to the SG includes the rules $p, q \Rightarrow s$; $p, s \Rightarrow r$; $p \Rightarrow r$; $p, q, r \Rightarrow s$ and $r, s \Rightarrow t$. Among the arguments which can be constructed from the SG are $A_1\colon p$, $A_2\colon A_1 \Rightarrow r$, $A_3\colon q$, $A_4\colon A_1, A_3 \Rightarrow s$, $A_5\colon A_2, A_4 \Rightarrow t$, $A'_2\colon A_1, A_4 \Rightarrow r$, $A'_4\colon A_1, A_2, A_3 \Rightarrow s$, and $A'_5\colon A'_2, A'_4 \Rightarrow t$. While the recall of the BN graphs from Fig. 5c is maximal, the precision is not; more specifically, the returned arguments A'_2, A'_4 and A'_5 were not in the original argument set $\mathcal{A}$.

Now, the equivalence class of BN graphs with just the immorality (R, T, S) is addressed; the BN graph depicted in Fig. 5d is an element of this class. From this BN graph, again the SG of Fig. 6 is constructed for the node of interest T, and thus the same arguments as above are returned. While the precision of the BN graph of Fig. 5d is equal to that of the BN graph from Fig. 5c, we note that the nodes R and S are conditionally independent given the evidence for $\mathbf{Z} = \{P, Q\}$ in the former graph, that is, in the BN graph with just the immorality (R, T, S). The immediate sub-argument A_4 of A'_2 and the immediate sub-argument A_2 of A'_4, therefore, appear to be irrelevant, as the associated reasoning is non-existent in this BN graph. As noted before, Timmer and colleagues [6] employ a quantitative step to filter the set of arguments returned by

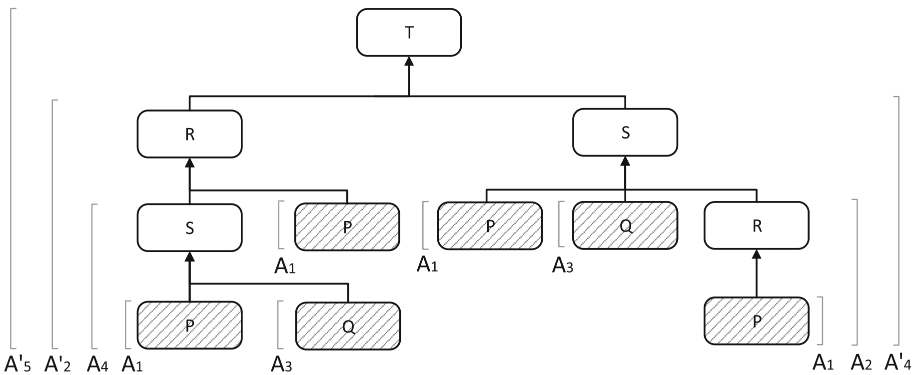

Fig. 6. The SG corresponding to the BN graph of Figs. 5c and d, with T as the node of interest; the SG is annotated with some of the possible arguments which can be extracted from it.

the support graph method; specifically, as the nodes R and S are conditionally independent given the evidence in the BN graph in Fig. 5d, A_4 and A_2 are filtered out as immediate sub-arguments of A'_2 and A'_4 respectively. Building on the conditional independence relations that can be inferred from the BN graph given the set of instantiated nodes, however, irrelevance of A_4 and A_2 as immediate sub-arguments of A'_2 and A'_4 can be decided upon by graphical considerations only, without involving the CPTs of the nodes. □

Based on the above example, we propose to set the directions of arcs in a BN skeleton such that no instantiated head-to-head nodes or head-to-head nodes with instantiated descendants are formed, as such head-to-head nodes may introduce unwarranted dependence relations. More specifically, the following refinement of step $3'$ of the BR heuristic is proposed, which fully specifies the directions of the arcs in a BN graph corresponding to a set of arguments $\mathcal{A}$:

$3''$. The directions of the arcs in a BN graph are set in the same direction as the arcs in the argument graph, that is, if A is an immediate sub-argument of B, then an arc should be drawn from the node corresponding to $\textsc{Conc}(A)$ to the node corresponding to $\textsc{Conc}(B)$.

We note that step $3''$ is a further refinement of step $3'$, as none of the immoralities (P_{i-1}, P_i, P_{i+1}) mentioned in that step are formed if arcs are set in the same direction as in the argument graph. By step $3''$, arcs are guaranteed to be set such that head-to-head nodes are not instantiated and do not have instantiated descendants, as the premise arguments in the argument graph, and hence the instantiated nodes in the BN graph, only have outgoing arcs. Finally, we note that step $3''$ is not a strict specification of the directions of the arcs in a BN graph; directions can possibly be reversed, given that an element from the same Markov equivalence class as specified by step $3''$ is obtained.

5 Conclusion and Future Research

In this paper, we have proposed a refinement of the heuristic of Bex and Renooij [1] for constructing BN graphs from structured arguments. This heuristic is aimed at aiding domain experts who are accustomed to argumentation to transform their reasoning into BNs and subsequently weigh their case evidence in a probabilistic manner. Our refinement consists of fully specifying the directions in which arcs should be set in a BN graph for a given argument structure without attack relations; more specifically, when employing the refined heuristic for a set of arguments $\mathcal{A}$, the directions of the arcs in the BN graph are set in the same direction as the arcs in the original argument graph of $\mathcal{A}$. By our refined heuristic, BN graphs with maximal recall are constructed, that is, the original arguments are returned by applying the support graph method to the constructed BN graphs. Furthermore, our refined heuristic prevents the creation of direct intercausal dependence relations between variables in the BN graph that did not exist between the corresponding propositions in the original argument graph. In the near future, we will evaluate the heuristic in practice by establishing, for example, the extent to which the automatically derived arc directions match the perceived real-world causality or the judgments of domain experts.

In this paper, we focused on improving the recall of BN graphs constructed by the BR heuristic. In our future research, we will address the construction of BN graphs with increased precision. Furthermore, we will extend our research to a more general framework of argumentation [4], not restricting ourselves to linked argument graphs without attack relations.

References

1. Bex, F., Renooij, S.: From arguments to constraints on a Bayesian network. In: Baroni, P., Gordon, T.F., Scheffler, T., Stede, M. (eds.) Computational Models of Argument: Proceedings of COMMA 2016, pp. 95–106. IOS Press, The Netherlands (2016)
2. Fenton, N., Neil, M.: Risk Assessment and Decision Analysis with Bayesian Networks. CRC Press, Boca Raton (2012)
3. Jensen, F.V., Nielsen, T.D.: Bayesian Networks and Decision Graphs, 2nd edn. Springer Verlag, Berlin (2007)
4. Prakken, H.: An abstract framework for argumentation with structured arguments. Argum. Comput. **1**(2), 93–124 (2010)
5. Shachter, R.D.: A graph-based inference method for conditional independence. In: D'Ambrosio, B.D., Smets, P., Bonissone, P.P. (eds.) Proceedings of the Seventh Conference on Uncertainty in Artificial Intelligence, pp. 353–360. Morgan Kaufmann Publishers Inc., San Mateo (1991)
6. Timmer, S.T., Meyer, J.-J.C., Prakken, H., Renooij, S., Verheij, B.: A two-phase method for extracting explanatory arguments from Bayesian networks. Int. J. Approx. Reason. **80**, 475–494 (2017)
7. van der Gaag, L.C., Helsper, E.M.: Experiences with modelling issues in building probabilistic networks. In: Gómez-Pérez, A., Benjamins, V.R. (eds.) EKAW 2002. LNCS (LNAI), vol. 2473, pp. 21–26. Springer, Heidelberg (2002). https://doi.org/10.1007/3-540-45810-7_4
8. van der Gaag, L.C., Meyer, J.-J.C.: The dynamics of probabilistic structural relevance. In: van der Gaag, L.C., Meyer, J.-J.C. (eds.) Proceedings of the Eighth Dutch Conference on Artificial Intelligence (NAIC'96), pp. 145–156. Utrecht University, Utrecht (1996)
9. Verheij, B., Bex, F., Timmer, S.T., Vlek, C.S., Meyer, J.-J.C., Renooij, S., Prakken, H.: Arguments, scenarios and probabilities: connections between three normative frameworks for evidential reasoning. Law Probab. Risk **15**(1), 35–70 (2015)
10. Verma, T., Pearl, J.: Equivalence and synthesis of causal models. In: Bonissone, P.P., Henrion, M., Kanal, L.N., Lemmer, J.F. (eds.) Proceedings of the Sixth Conference on Uncertainty in Artificial Intelligence, pp. 255–270. Elsevier Science Inc., New York (1991)
11. Wellman, M.P., Henrion, M.: Explaining "explaining away". IEEE Trans. Pattern Anal. Mach. Intell. **15**(3), 287–292 (1993)