Enhancing Argument Generation Using Bayesian Networks

Yuan Cao[1,4](✉), Rafael Fuchs[2,3], and Anita Keshmirian[1,2,5]

[1] Fraunhofer-Institut für Kognitive Systeme IKS, Munich, Germany
yuan.cao@mein.gmx
[2] Munich Center for Mathematical Philosophy (MCMP) - LMU, Munich, Germany
Rafael.Fuchs@campus.lmu.de
[3] Graduate School of Systemic Neuroscience (GSN) - LMU, Munich, Germany
[4] Technical University of Munich, Munich, Germany
[5] Forward College, Berlin, Germany

Abstract. In this paper, we examine algorithms that utilize factor graphs from Bayesian Belief Networks to generate and evaluate arguments. We assess their strengths and weaknesses, which leads to the creation of our improved algorithm that rectifies the issues that we identified. Our approach includes applying the original and modified algorithms to previously known networks to pose challenges in generating robust arguments for humans and computers. Our findings reveal significant improvements in the creation of more robust arguments. Moreover, we delve into the dynamics of argument interaction, offering detailed insight into the algorithms' practical efficacy.

Keywords: Argument Strength · Bayesian Belief Network · Argument Generation

1 Introduction

Argumentation is central to collective reasoning, informed decision-making, and decision articulation within collaborative contexts. Yet uncertainty pervades decision-making in real life: in a medical setting, doctors often face uncertain scenarios where they must make critical decisions based on incomplete information. Investment decisions are fraught with uncertainty in finance. In the judiciary, verdicts by judges and juries frequently rely on evidence that lacks absolute certainty. Therefore, realistically applicable argumentation theory must be able to cope with reasoning under uncertainty.

Supported by Deutsche Forschungsgemeinschaft (DFG) - Project number 455912038 and the Bavarian Ministry for Economic Affairs, Regional Development and Energy as part of a project to support the thematic development of the Fraunhofer Institute for Cognitive Systems IKS

Y. Cao, R. Fuchs and A. Keshmirian—Equally contributed to this work.

P. Cimiano et al. (Eds.): RATIO 2024, LNAI 14638, pp. 253–265, 2024.
https://doi.org/10.1007/978-3-031-63536-6_15

The necessity to navigate the complexities of decision-making under uncertainty has sparked significant interest in developing algorithms that facilitate probabilistic reasoning. Such algorithms could enhance the explainability of expert systems, particularly those utilizing Bayesian Belief Networks (BBNs).

BBNs are graphical tools for modeling probabilistic dependencies between variables and facilitating reasoning under uncertainty [4,5]. The use of BBNs to provide explanations in real-world settings faces challenges because of the complexity involved, with many variables and detailed interactions. Developing explanations through the extraction of arguments underscores the need for an argumentation theory that effectively navigates uncertainty and is comprehensible to experts and non-experts.

One method to elucidate BBNs' decisions involves distilling complex arguments into more straightforward, comprehensible segments. However, simplifying arguments presents a dichotomy: while disassembling complex arguments into simpler components enhances transparency and comprehensibility, it risks oversimplification, where the interconnected nature of premises is pivotal. Hence, there is a fundamental trade-off: make the representation of an argument as straightforward as possible while maintaining a sufficient level of accuracy concerning the underlying probabilistic reasoning structure. This balance - streamlining argument representation without compromising the integrity of the underlying probabilistic logic - is at the heart of our paper.

In Sect. 2, we motivate the question of independent arguments and introduce an algorithm by Sevilla [7] that uses factor graphs to extract arguments from BBNs, which also gives a useful criterion for independent arguments.

In Sect. 3, we identify some problems in Sevilla's algorithm using a scenario known as "The Spider" [6] to assess the performance of algorithms in providing explanations. The Spider case is notable for previously testing both human and artificial agents with its complex scenario, frequently uncovering instances of less-than-ideal reasoning. [2]. We propose our own improvements to the algorithm by showing its enhanced reasoning. Finally, we present our improved version results and demonstrate the threshold's merits for independent arguments in the factor graph approach.

2 The Question of Independent Arguments

In this section, we look at probabilistic argumentation, explicitly examining how arguments depend on (or are independent of) each other. We focus on extracting arguments from BBNs using factor graphs. First, we briefly introduce factor graphs. Following this, we present a detailed overview of a specific algorithm, as proposed by Sevilla [7], explaining its methodology in the field of probabilistic argumentation. Apart from Sevilla's work, the work on extracting arguments from BBNs is scarce. Other algorithms rely on graphical methods (e.g., [8,9]), but none use factor graphs. Since Sevilla's factor graph approach is novel in this respect, we aim to explore its potential and the power of its criterion for argument independence.

2.1 Factor Graphs

In probabilistic argumentation, it is essential to identify when arguments are independent, as this clarity helps to understand each argument's role in a complex discussion. Factor graphs, which build upon the ideas of BBNs, provide a clear framework for mapping and studying the parts and behavior of arguments. This approach is especially useful when the argumentation process can be simplified into smaller, more manageable functions, each concerning a specific set of variables.

Technically, factor graphs are a type of graphical model used in probability theory and statistical modeling to represent the factorization of a function. Consider a probability distribution $P(X_1, X_2, \ldots, X_n)$ over n random variables. This distribution can be factorized as:

$$P(X_1, X_2, \ldots, X_n) = \prod_{k=1}^{K} f_k(S_k)$$

where $f_k(S_k)$ represents a factor over a subset of variables, and K is the number of factors. Graphically, this factorization is represented as a bipartite graph with variable nodes (X_i) and factor nodes (f_k). An edge is drawn between a variable node and a factor node if the variable is in the subset for that factor. For details, see [1,4].

When using factor graphs for argument extraction, variable nodes can represent components of an argument, such as claims, evidence, counterarguments, and assumptions, whereas factor nodes represent inference rules. Each element plays a distinct role in the structure of the argument. Factors represent the probabilistic relationships between these components, e.g., a factor might represent the strength of evidence supporting a claim or the impact of a counterargument on the overall argument's validity. Using probabilistic models, the factor graph can accommodate uncertainties and variabilities inherent in arguments, including assessing the likelihood of a claim's validity based on the available evidence.

2.2 Overview of the Factor-Graph-Approach Proposed by J. Sevilla

The algorithm constructs a factor graph from a BBN as follows[1]. It creates variable nodes for each variable from the BBN and factors representing the conditional probability tables. Connections between variable nodes and their respective factors are established in these conditional probabilities. To calculate and update joint probability distributions in the factor graph, the message passing algorithm [4] is used.

In preparation for message passing, observation nodes are set to lopsided factors (i.e., zero or one) for the initialization phase, reflecting known states with a probability of one and all other states with zero probability. Other nodes are initialized with constant factors, assuming a uniform distribution. Once the factor

[1] For sources containing a pseudo-code and the technical implementation, see material in Appendix A.

graph is established, the algorithm implements the message-passing algorithm to calculate the flow of messages across the graph.

Effects and Strength of an Argument: This approach represents arguments as directed acyclic graphs over the factor graph. An argument, for example, is shown in Fig. 4. It comprises nodes and factors ranging from observation to the target node. The influence of each inference step in an argument is called *Step Effect* and is defined by how a preceding node impacts the subsequent node. More specifically, the argument's premises (variable nodes) are multiplied with their factor node (inference rule) as per the message-passing algorithm, and the result is normalized by dividing by the factor itself. This division distinguishes new information (Δ) and inherent data in the conditional probability table (ϕ).

The cumulative effect of an argument is calculated by multiplying the effects of all parent factors through the recursive application of the step effect. Finally, the strength of an argument is measured by the logarithmic odds of its effect supporting the outcome. This provides a real-valued metric that indicates the argument's direction (support or opposition) and magnitude (strength).

Argument Independence: Determining argument independence involves assessing if the combined effect of multiple arguments equals the product of their individual effects. Arguments are independent if their effect's discrepancy falls within a predefined threshold. This is measured as the maximum absolute difference in log odds between the factors, represented by the equation:

$$\text{Factor Distance}(\phi_1, \phi_2) = \max \left| \log \frac{(\phi_1/\phi_2(t_0))\,(t_o)}{\text{Average}_{t \neq t_o}\,(\phi_1/\phi_2)\,(t)} \right|,$$

where $(\phi_1/\phi_2)(t_0)$ is the probability ratio $\phi_1(t_0)$ to $\phi_(t_0)$ (i.e. the probability that variable T takes value t_0 given ϕ_1 vs that probability given ϕ_2), which is compared to the average of all values t of T such that $t \neq t_0$ (see pseudo-code in Appendix A).

Finding All Arguments: The algorithm's objective is to identify a set of relevant and independent arguments that elucidate the network's outcome based on given premises and a target. It begins by identifying simple arguments[2]) from each evidence node to the target, excluding paths passing through another evidence node. The algorithm then iteratively combines these simple arguments into more complex ones, checking for their potential breakdown into independent combinations. Two thresholds are set to accommodate larger BBNs: one for the length of simple paths (from one premise node to the query node) and another for the number of these simple paths to be combined. Finally, dependent arguments are amalgamated, and all arguments are ordered by their absolute strength.

[2] an argument is simple if it cannot be broken down into a union of distinct sub-arguments, [7, p. 6].

Explaining Arguments: Natural language explanations of arguments are generated by tracing the nodes each simple argument passes through. The outcome is determined based on the evidence favored by the message-passing algorithm's results.

3 Testing and Improving the Factor Graph Algorithm

In this section, we identify some problems in Sevilla's algorithm that lead to incorrect results in an application scenario ("The Spider") we used to test it. We then propose improvements and show how the improved algorithm yields better outcomes. Finally, we test different threshold levels for independent arguments.

3.1 Overview of the BARD Project and "the Spider" Problem

The BARD project [2,3] sets out to establish an overarching framework leveraging BBNs to advance argumentation. This initiative mainly tailors decision scenarios to underscore the complexities and challenges faced in decision-making endeavors mediated by BBNs, focusing on navigating through evidence conflicts, gauging source reliability, and encapsulating uncertainty to ensure clarity and comprehension.

Our research focuses on the "The Spider" problem presented in the BARD project, as described by Pilditch (2019) [6]. This scenario serves as a testing ground for dealing with misleading information sources.

In this exercise, participants assume the role of intelligence analysts on the hunt for a notorious foreign spy, known as "The Spider," suspected to be hiding in a facility located in a neutral country. The primary objective is to gather additional intelligence to determine the necessity of a covert operation to capture the Spider. Initial reports from agents Emerson and Quinn place the Spider within the facility, with both agents acclaimed for their high reliability (characterized by low false-positive and false-negative rates). However, emerging telephone records cast suspicion on Emerson and Quinn's loyalty, insinuating they might collaborate with the Spider. On the other hand, the records might also be forged: the Spider's true allies might have created them to spread disinformation. If the records turn out to be authentic, it would mean that Emerson and Quinn consistently report the opposite (i.e., if the Spider is in the facility, they report that he is not, and vice versa).

Finally, Winter, a communication analyst known for her meticulousness (almost zero false positives), confirms the Spider's presence through surveillance data. Trustworthy field agent Sawyer and local witness Alpha echo this claim. The structure of this scenario is visualized in the BBN shown in Fig. 1 (all variables are binary).

The decision-making process in this scenario is challenging due to conflicting reports from Emerson, Quinn, and the other members. In particular, uncertainty regarding the authenticity of the telephone records adds another layer of complexity to the conflict. How should we weigh the highly reliable information of one group reporting negatively against the collective inputs of the other members reporting positively? This dilemma underscores the intricacy of the Spider problem and highlights the need for an effective strategy to resolve such conflicts. We will implement the algorithms for this problem to analyze their reliability.

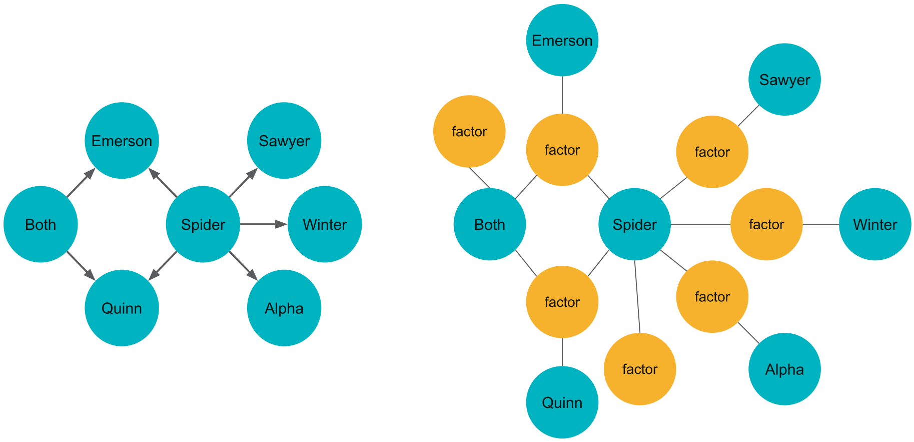

Fig. 1. The structure of the Spider network and its factor graph. Left: the BBN of "The Spider". Right: The factor graph of "The Spider" network. The blue nodes represent the nodes in the BBN, and the orange nodes represent factors. (Color figure online)

3.2 Results with the Original Algorithm

In this section, we apply Sevilla's original algorithm to "The Spider" problem, addressing a fundamental question: based on your evidence, "the Spider is not in the facility" from Emerson and Quinn and "the Spider is in the facility" from Sawyer, what do you believe the probability is of "The Spider" is in the facility? Additionally, we adjust the threshold settings to explore the interactions between different arguments.

Each paragraph in Fig. 2 and Fig. 3 is an argument. For instance, the structure of the first argument in Fig. 2 is from Sawyer to "The Spider" as shown in Fig. 4. The arguments favor "The Spider" being in the facility (Spider is true) or neutral (Spider is true or Spider is false). This means that taken together, the arguments of this algorithm suggest "The Spider" is in the facility when it is known that Emerson and Quinn report the absence of "The Spider" and Sawyer reports the presence of "The Spider".

As depicted in Fig. 2, the default threshold condition results in a clear separation of all arguments. Upon reducing the threshold value, we observe that arguments are identified as being interdependent. Figure 3 illustrates an interaction between the arguments originating from Quinn and Emerson towards Spider, which is a notable deviation from their previously independent status shown in Fig. 2. The threshold deciding the interaction level of arguments is user-defined

```
We have observed that Sawyer is True.
That Sawyer is True is evidence that Spider is True (strong inference).

We have observed that Emersons is False.
That Emersons is False is evidence that Spider is True (weak inference).

We have observed that Quinns is False.
That Quinns is False is evidence that Spider is True (weak inference).

We have observed that Quinns is False.
That Quinns is False is evidence that Both is False or Both is True (certain inference).
That Both is False or Both is True is evidence that Spider is False or Spider is True (certain inference).

We have observed that Emersons is False.
That Emersons is False is evidence that Both is False or Both is True (certain inference).
That Both is False or Both is True is evidence that Spider is False or Spider is True (certain inference).
```

Fig. 2. Results from the original algorithm with default threshold = 0.1.

```
We have observed that Sawyer is True.
That Sawyer is True is evidence that Spider is True (strong inference).

We have observed that Emersons is False and Quinns is False.
That Emersons is False is evidence that Spider is True (weak inference).
That Quinns is False is evidence that Spider is True (weak inference).
All in all, this is evidence that Spider is True (weak inference).

We have observed that Quinns is False.
That Quinns is False is evidence that Both is False or Both is True (certain inference).
That Both is False or Both is True is evidence that Spider is False or Spider is True (certain inference).

We have observed that Emersons is False.
That Emersons is False is evidence that Both is False or Both is True (certain inference).
That Both is False or Both is True is evidence that Spider is False or Spider is True (certain inference).
```

Fig. 3. Results from the original algorithm with threshold $= 2 \times 10^{-16}$.

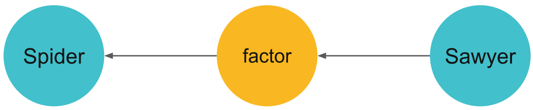

Fig. 4. The first argument in Fig. 2. The direction is from the observation to the query node.

and can be adjusted based on specific situations. The optimal threshold varies depending on the scenario.

3.3 Diagnosis and Solution Proposal

Here, we present our in-depth exploration of the algorithm's technical difficulties and shortcomings. We provide a comprehensive analysis of their causes and effects. Following this analysis, we propose targeted solutions and enhancements to improve the algorithm's accuracy and reliability.

Ignorance of Prior Probability. The initialization of the nodes without information assumes a uniform distribution, which leads to the wrong calculation of the probability marginalization of the outcome. In Fig. 2, Quinn's report of the Spider's absence paradoxically suggests the Spider's presence, contrasting our initial expectations. We anticipate that if Quinn reports the absence of the Spider, it would significantly increase the likelihood of its absence, considering the low propensity to be league with the Spider. To rectify this, we propose changing the initialization of nodes, except for evidence nodes, to reflect their prior probabilities.

Certain Inference. When distinct node states are assigned equivalent probabilities, the algorithm returns a "certain inference." However, this might be misleading about a definitive node's state, which is not the case. To address this semantic inconsistency, we propose renaming this outcome "equal effect inference."

D-Separation Detection Deficiency. The algorithm is unable to identify d-separation structures: two (non-empty) sets of nodes X, Y are d-separated by another (possibly empty) set of nodes Z, if and only if every path from a node $x \in X$ to a node $y \in Y$ is blocked. A path $x_i \rightarrow v \rightarrow ... \rightarrow y$ is blocked by Z iff for every node w on the path one of the following two holds:

1. the path's edges do not meet head-to-head in w and $w \in Z$, or
2. the edges meet head-head in w and $w \notin Z$ **and** none of w's descendant are in Z.

D-separation identifies conditional independence relations between nodes in a Bayes net. Our results indicate that an effect exists between d-separated nodes. We adapted the algorithm to evaluate d-separation between nodes for every step of the argument process. An identification of d-separation signifies that the argument does not affect the target node.

Uncertain Equivalence Between Node Value and Step Effect. In each step, the value of the step target node equals the step effect when moving from parent to child. Conversely, from child to parent, the value equals the step effect times the parent's prior probability. This distinction arises because the step effect represents $P(\text{child}|\text{parent})$. When calculating $P(\text{parent}|\text{child})$, it equals $P(\text{child}|\text{parent}) * P(\text{parent})/P(\text{child})$ according to Bayes rule. By first determining the direction of the effect, we increase the precision of our effect and strength calculations.

Table 1. Conclusion of the improvements

Dimensions	The original algorithm	Our improved algorithm
Initialization of unobserved nodes	uniform distribution	prior probability
Explanation in words	certain inference	equal effect inference
D-separation	Non D-separated detection	D-separated detection addition
Linking the possibility and step effect	Equivalence	Considerations of node relationship

To summarise, our improvements are listed in Table 1.

3.4 Results of the Improved Version

```
We have observed that Quinns is False, Emersons is False and Sawyer is True.
That Quinns is False is evidence that Both is False (strong inference).
That Both is False and Emersons is False is evidence that Spider is False (strong inference).
That Quinns is False is evidence that Spider is False (strong inference).
That Sawyer is True is evidence that Spider is True (strong inference).
All in all, this is evidence that Spider is False (strong inference).

We have observed that Emersons is False, Quinns is False and Sawyer is True.
That Emersons is False is evidence that Both is False (strong inference).
That Emersons is False is evidence that Spider is False (strong inference).
That Both is False and Quinns is False is evidence that Spider is False (strong inference).
That Sawyer is True is evidence that Spider is True (strong inference).
All in all, this is evidence that Spider is False (strong inference).
```

Fig. 5. Results from our updated algorithm with default threshold = 0.1.

After implementing our enhanced algorithm to revisit "The Spider" case, we observed that the outcomes were significantly more plausible than the original results. When Emerson or Quinn reports the argument, arguments are identified as independent with an elevated threshold. This outcome is consistent with their established reliability and the low likelihood of them being allied with the Spider. The outcomes presented in Fig. 5 demonstrate a merging of arguments under the standard threshold. Conversely, Fig. 6 shows that the arguments are identified as independent with an elevated threshold.

```
We have observed that Emersons is False.
That Emersons is False is evidence that Spider is False (strong inference).

We have observed that Quinns is False.
That Quinns is False is evidence that Spider is False (strong inference).

We have observed that Quinns is False.
That Quinns is False is evidence that Both is False (strong inference).
That Both is False is evidence that Spider is False or Spider is True (equal effect inference).
Because Spider and Both are d-separated, this argument alone cannot influence the target node.

We have observed that Emersons is False.
That Emersons is False is evidence that Both is False (strong inference).
That Both is False is evidence that Spider is False or Spider is True (equal effect inference).
Because Spider and Both are d-separated, this argument alone cannot influence the target node.

We have observed that Sawyer is True.
That Sawyer is True is evidence that Spider is True (strong inference).
```

Fig. 6. Results from our updated algorithm with threshold = 6.

Figure 6 further showcases the ability of the algorithm to detect d-separation. Analyzing an individual argument with Quinn leading to Spider via Both, the nodes Both and Spider are d-separated within the Both ← Emerson → Spider collider structure. The impact of Quinn on Spider is interrupted in this sequence. The algorithm detects the d-separation and informs users that this particular type of argument does not influence the target node.

4 Limitation and Future Work

This paper identifies and addresses key areas for enhancement within the factor graph-based approach to the algorithmic generation and evaluation of arguments. We have introduced modifications that considerably bolster reasoning capabilities. Our preliminary research, centered on the exemplary use of a complex and challenging Bayesian Belief Network ("The Spider"), has illuminated promising avenues for refining reasoning strategies. Despite these advancements, there remains substantial scope for future research to validate these algorithmic improvements across a more varied array of scenarios and Bayesian Belief Networks (BBNs), thus underlining their widespread applicability and efficacy.

Through this exploration, we enhance reasoning capabilities and underscore the significance of setting a threshold for independent arguments within the factor graph framework. This work establishes a solid foundation for further investigation into the algorithm's operational effectiveness. Building upon this foundation, we aim to extend our analysis to a wider range of BBNs. This endeavor is motivated by our goal to affirm the universality and practical utility of the proposed algorithmic enhancements.

Moreover, future research is crucial to build upon our findings through empirical evaluation. This subsequent research phase will compare the human understanding and evaluation of the algorithm's arguments against its actual performance. By incorporating a more extensive set of examples and applying quantitative accuracy metrics, we aim to solidify the evidence supporting our claims of improved algorithmic performance. This approach addresses the limitations identified and deepens our comprehension of how these algorithmic enhancements can significantly enhance human reasoning processes in the face of uncertainty.

5 Conclusion

This paper pinpoints and tackles crucial improvement opportunities within the factor-graph-based approach to generating and evaluating arguments using Bayesian Belief Networks (BBNs). We have implemented changes that strengthen the reasoning abilities of an exemplary algorithm that uses factor graphs.

Refining Sevilla's algorithm, we demonstrated that meaningful argument extractions from BBNs are possible within this approach. We especially noted the utility of establishing a threshold for independent arguments. This feature, in particular, showcases the potential for more precise and nuanced argumentation within complex probabilistic models.

Acknowledgements. We thank the Deutsche Forschungsgemeinschaft (DFG) and the Fraunhofer Institute for Cognitive Systems IKS for supporting our research. We thank Dr. Narges Ahmidi and Alireza Zamanian for their insightful suggestions and valuable feedback on this paper. We would also like to thank anonymous reviewers for their useful comments.

A Appendix

Our enhancements to this algorithm, based on Sevilla's packages and the PGMPy open-source software for computing Bayesian networks, can be viewed in detail at this link[3]. To help you understand our enhancements better, we have attached pseudo-code for the relevant parts.

For the initialization of the unobserved nodes, they are under a uniform distribution initialization:

```
for node in model:
    if node in evidence:
    node[observed_state] = 1
    node[other_states] = 0
    else:
    node[states] = uniform distribution
```

[3] https://github.com/yuancao-git/factor_graph_algorithm.git.

We modify it to make use of prior probabilities:

```
for node in model:
    if node in evidence:
    node[observed_state] = 1
    node[other_states] = 0
    else:
    prior = model.prior_calculation(node)
    node[states] = prior
```

We change how we calculate the argument's strength to make the explanations more precise. The outcomes are altered from "certain inference":

```
query_node = [v1, v2, ..., vn] (set V)
if query_node has one maximum vm:
    argument_strength = vm/sum(V out of vm)
if query_node has several maximums vm1,...,vmm:
    choose all these states vm1,...,vmm (set Vm)
    argument_strength=sum(vm1,...,vmm)/sum(V out of Vm)
```

to the "equal effect inference":

```
query_node = [v1, v2, ..., vn] (set V)
if query_node has one maximum vm:
    argument_strength = vm/sum(V out of vm)
if query_node has several maximums vm1,...,vmm:
    choose all these states vm1,...,vmm (set Vm)
    argument_strength = vm1 / sum (V out of Vm)
    display ``equal effect inference''
```

Before calculating the argument strength, we add D-separation detection to check if there is influence between nodes:

```
for argument in all_arguments:
    for step in argument:
    sub_BBN = to_BBN(argument,step)
    d_separation =
        d_separated(sub_BBN,evidence_in_argument)
    if d_separation == True:
        ``no influence''
    if d_separation == False:
        calculate argument_strength
```

References

1. Acar, U.A., Ihler, A., Mettu, R., Sümer, O.: Adaptive Bayesian inference. Neural Information Processing Systems (NIPS), vol. 10, pp. 2981562–2981743 (2007)
2. Cruz, N., et al.: Widening access to Bayesian problem-solving. Front. Psychol. **11**, 660 (2020)
3. Dewitt, S., Lagnado, D., Fenton, N.: Updating prior beliefs based on ambiguous evidence. In: COGSCI2018: Changing Minds, pp. 2047-2052 (2018)
4. Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles And Techniques. MIT Press, Cambridge, MA (2009)
5. Lauritzen, S.L.: Graphical Models, vol. 17. Clarendon Press, Oxford (1996)
6. Pilditch, T.D., Fries, A., Lagnado, D.A.: Deception in evidential reasoning: Willful deceit or honest mistake? In Proceedings of the CogSci, pp. 931-937 (2019)
7. Sevilla, J.: Finding, Scoring, and Explaining Arguments in Bayesian Networks. arXiv preprint arXiv:2112.00799. https://arxiv.org/abs/2112.00799 (2021)
8. 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. Approximate Reasoning **80**, 475–494 (2017)
9. Keppens, J.: Argument diagram extraction from evidential Bayesian networks. Artif. Intell. Law **20**(2), 109–143 (2012)