How Can You Resolve a Trilemma?
- A Topological Approach -*

Kazuko Takahashi and Tamon Okubo**

Kwansei Gakuin University, 2-1, Gakuen, Sanda, 669-1338, JAPAN
ktaka@kwansei.ac.jp tamonovsky@gmail.com

Abstract. This paper discusses how to escape a state in which argumentation can reach no conclusion, by offering a new argument. We formalize our approach based on Dung's abstract argumentation framework (AF). When an AF has no stable extension, we have no meaningful conclusion. We address the problem of whether it is possible to revise this situation by adding an argument that attacks an existing one. If possible, how many solutions can we generate and at what position should it be added? We discuss this problem using an AF consisting of a trilemma and show conditions depending on the topology of the AF. We also address the point that a specific argument can be accepted or not by this action. We extend the discussion into two possible directions: a general N-lemma case and a set of AFs, each of which consists of several trilemmas. It follows that when a large argumentation becomes stuck in a practical situation, the position to which a counter-argument should be added can be detected by a check of the topology of the AF.

Keywords: abstract argumentation, computational argumentation, revision of argumentation, graph topology

1 Introduction

Argumentation appears in many scenes in our daily life and has been studied from various perspectives. In the field of artificial intelligence and logic programming, the Abstract Argumentation Framework (AF) introduced by Dung [15] has been regarded as a strong framework to handle inconsistency and has generated considerable work on computational argumentation [17].

An AF can be represented as a directed graph in which a node corresponds to an argument, and an edge to an attack relation. When we consider an argumentation as a graph, we find several topological types. One type that attracts our interest is that including a cycle, which means that arguments are attacked by each other.

When two arguments A and B are attacked by each other, we cannot arrive at a unique outcome that each agent can accept. In this case, either A or B is

* This work was supported by JSPS KAKENHI Grant Number JP17H06103.
** Currently, Fuji Soft Incorporated.

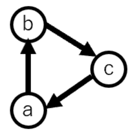


Fig. 1. A trilemma in argumentation.

acceptable. Furthermore, consider what happens when three arguments, A, B, and C, attack in such an order that A attacks B, B attacks C, and C attacks A. We call this a *trilemma*. In this case, either one of A, B or C is acceptable. However, this result is weak in the sense that each single argument does not attack all of the other arguments. According to Dung's semantics, such an AF does not have a stable extension, and no argument is skeptically or credulously accepted. In practical argumentation, the argumentation becomes stuck, and no meaningful result is possible. We can escape from this sticky state by providing a new counter-argument. Moreover, if we want a specific argument to be accepted, we have to choose an appropriate position.

For example, consider the situation in which three agents give their arguments:

a: We should go to Okinawa; it is cold in Hokkaido.

b: We should go to Tokyo because it costs a lot to go to Okinawa.

c: We should go to Hokkaido because we cannot find beautiful scenery in Tokyo.

In this case, these arguments constitute a trilemma (Figure 1). If an agent adds a new argument d, "It is risky to go to Tokyo now because of COVID-19," then the argument b is defeated, and as a result, a and d are accepted. Therefore, if an agent wants her claim a to be accepted, she needs to offer such an argument.

If the entire argumentation is larger, trilemmas may appear in many locations, and they may interact in complicated ways. In such a case, how can one agent find a way to persuade the others?

Changes in argumentation systems have been discussed in several works [14]. In these works, the authors consider the properties depending on the patterns of change in extensions and do not discuss the position to which a new attack is added. Here, we do not address the problem based on a principle of change in extension types but in terms of positions where an argument will be added.

In this paper, we consider AFs consisting of trilemmas that share one or two nodes. We investigate the properties of such an argument graph according to each topology and formalize them. We focus on stable semantics, since this is considered most suitable in a practical situation for drawing a plausible conclusion that is admitted by all agents and that attacks every argument against the conclusion.

More specifically, we discuss the problem of whether we can get an AF with a stable extension by adding one argument and an attack to the AF without it. If it is possible, we show the position to be added. We consider such a change based on the topology. Starting from a simple trilemma, we discuss extensions of the result in two directions: the N-lemma case and the meta-AF case.

Our aim is to resolve a stuck argumentation by offering a counter-argument for a practical situation, rather than to find a general method that can be applied to any argument graph including the ones that seldom appear in practical situations.

This paper is organized as follows. In Section 2, we describe basic concepts. In Section 3 and in Section 4, we consider the stable extension in the case of an AF including a single trilemma and an AF constructed by more than one trilemma, respectively, as well as one including the N-lemma. In Section 5, we consider a case of a meta-AF. In Section 6, we compare our approach with related works. Finally, in Section 7, we present conclusions and directions for future research.

2 Basic Concepts

The abstract argumentation framework (AF), proposed by Dung [15], is a representation of an argumentation structure, ignoring its content.

Definition 1 (argumentation framework (AF)). Argumentation Framework (AF) *is defined as a pair* $\langle \mathcal{A}, \mathcal{R} \rangle$ *where* $\mathcal{A}$ *is a set of arguments and* $\mathcal{R} \subseteq \mathcal{A} \times \mathcal{A}$.

A pair $(A, B) \in \mathcal{R}$ is called *an attack*, and it is said that A *attacks* B.

AF can be represented as a graph in which each node corresponds to an argument, and each edge corresponds to an attack. In this paper, we consider a finite AF.

Definition 2 (sub-AF). *Let* $AF_1 = \langle \mathcal{A}_1, \mathcal{R}_1 \rangle$ *and* $AF_2 = \langle \mathcal{A}_2, \mathcal{R}_2 \rangle$ *be AFs. An* AF_1 *is said to be* a sub-AF *of* AF_2 *if* $\mathcal{A}_1 \subseteq \mathcal{A}_2$ *and* $\mathcal{R}_1 \subseteq \mathcal{R}_2$*, and is denoted by* $AF_1 \subseteq AF_2$.

Semantics is defined either by an extension or labeling, which has a one-to-one relation [2]. In this paper, we consider stable semantics.

Definition 3 (stable extension). *Let* $\langle \mathcal{A}, \mathcal{R} \rangle$ *be an AF.* $S \subseteq \mathcal{A}$ *is said to be a* stable extension *if the following two conditions hold.*

- $\neg \exists A, B \in S; (A, B) \in \mathcal{R}$ *(There is no pair of arguments that attacks each other (conflict-freeness).)*
- $\forall B \in \mathcal{A} \setminus S, \exists A \in S; (A, B) \in \mathcal{R}$ *(Each argument outside the set is attacked by some argument in the set (stability).)*

Definition 4 (labeling, complete labeling). *Let* $\langle \mathcal{A}, \mathcal{R} \rangle$ *be an AF. Labeling is a total function from a set of arguments to a set* $\{\text{in}, \text{out}, \text{undec}\}$*. Labeling* $\mathcal{L}$ *is said to be* complete *if the following conditions are satisfied for any argument* $A \in \mathcal{A}$.

- $\mathcal{L}(A) = \text{in}$ *iff* $\forall B \in \mathcal{A}; (B, A) \in \mathcal{R} \Rightarrow \mathcal{L}(B) = \text{out}$.

– $\mathcal{L}(A) = \text{out}$ *iff* $\exists B \in \mathcal{A}; \mathcal{L}(B) = \text{in} \wedge (B, A) \in \mathcal{R}$.

Definition 5 (stable labeling). *Let* $\langle \mathcal{A}, \mathcal{R} \rangle$ *be an AF. For a complete labeling* $\mathcal{L}$*, if* $\{A | A \in \mathcal{A}, \mathcal{L}(A) = \text{undec}\} = \emptyset$*, then it is called* stable labeling.

It has been proven that stable extension and stable labeling coincide, that is, a stable extension corresponds to exactly one stable labeling and vice versa [2].

In addition to these concepts, we introduce several new concepts and terminology.

An AF with three arguments that constitutes a cycle is called a triangular unit.

Definition 6 (triangular unit (TU)). *An AF of the form* $\langle \{A, B, C\}, \{(A, B), (B, C), (C, A)\} \rangle$ *is called* a triangular unit (TU)*, and is denoted by* $t(A, B, C)$*. When a TU is a sub-AF of an AF, then it is said that* the AF includes a triangular unit*.*

Definition 7 (connector, faucet). *Let* $\langle \mathcal{A}, \mathcal{R} \rangle$ *be an AF that includes a triangular unit* $\mathcal{T} = \langle \mathcal{A}_\mathcal{T}, \mathcal{R}_\mathcal{T} \rangle$*. If* $(A, B) \in \mathcal{R}, A \in \mathcal{A} \setminus \mathcal{A}_\mathcal{T}, B \in \mathcal{A}_\mathcal{T}$*, then* B *is said to be* a connector *of* $\mathcal{T}$*; if* $(A, B) \in \mathcal{R}, A \in \mathcal{A}_\mathcal{T}, B \in \mathcal{A} \setminus \mathcal{A}_\mathcal{T}$*, then* A *is said to be* a faucet *of* $\mathcal{T}$*.*

Example 1. In Figure 2, a is the connector, and b and c are faucets of $t(a, b, c)$, respectively.

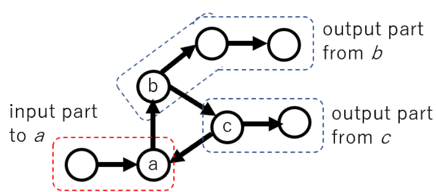


Fig. 2. Example of a triangular unit (TU).

For a TU, a sub-AF connected to C, which includes C itself, from the outside of the TU is called an *input part to* C and that connected from F, which includes F itself, to the outside of the TU is called an *output part from* F (Figure 2).

Definition 8 (input part, output part). *(1) For an AF* $\langle \mathcal{A}, \mathcal{R} \rangle$ *including TU* $\langle \mathcal{A}_\mathcal{T}, \mathcal{R}_\mathcal{T} \rangle$*, its sub-AF* $\langle \mathcal{A}_1, \mathcal{R}_1 \rangle$ *such that* $\mathcal{A}_1 = \{A \mid \exists \sigma = (A_1, \ldots, A_{n-1}); \forall i (1 \leq i \leq n-1)(A_i, A_{i+1}) \in \mathcal{R} \setminus \mathcal{R}_\mathcal{T}, \text{ where } A_1 = A, A_n = C \in \mathcal{A}_\mathcal{T}\} \cup \{C\}$*, and* $\mathcal{R}_1 = \{(A, B) | A, B \in \mathcal{A}_1\}$*, is called an* input part to C*.*
(2) For an AF $\langle \mathcal{A}, \mathcal{R} \rangle$ *including TU* $\langle \mathcal{A}_\mathcal{T}, \mathcal{R}_\mathcal{T} \rangle$*, its sub-AF* $\langle \mathcal{A}_2, \mathcal{R}_2 \rangle$ *such that*

$\mathcal{A}_2 = \{B \mid \exists\sigma = (A_1, \ldots, A_{n-1}); \forall i(1 \le i \le n-1)(A_i, A_{i+1}) \in \mathcal{R}\backslash\mathcal{R}_{\mathcal{T}}$, where $A_1 = F \in \mathcal{A}_{\mathcal{T}}, A_n = B\} \cup \{F\}$, *and* $\mathcal{R}_2 = \{(A,B)|A, B \in \mathcal{A}_2\}$, *is called an* output part from F.

Note that if a node C is not a connector, then *input part to C* consists of only C. An AF may have several input parts or output parts.

Definition 9 (start-TU). *A triangular unit included by an AF without a connector is said to be a* start-TU*.*

Definition 10 (whisker). *For a new argument P and an attack I from P, a pair $\langle P, I\rangle$ is said to be a* whisker*, and P is said to be* a whisker node*.*

Definition 11 (stable AF, repair[1]). *An AF with a stable extension is called a* stable AF*, and one without a stable extension is called an* unstable AF*. For an unstable AF, the act of revising it by adding a single whisker to get a stable AF is called a* repair*.*

The node to which a whisker is added is always labeled out when repaired, from the definition of stable labeling.

Definition 12 (entrance, acceptance set). *If we repair an unstable AF by adding a whisker to a node E, then the node is called an* entrance of AF *and their set is denoted by $ent(AF)$, and the obtained stable extension is called an* acceptance set*.*

3 AF Including One Triangular Unit

We pick up a triangular unit as the simplest odd-length cycle, and consider a finite AF that includes at most three TUs sharing their nodes. We assume that the entire AF has no cycle other than TUs and that it is uncontroversial, that is, there exists no arguments A and B connected by two different paths of even-length and odd-length. From this assumption, we have only one stable extension as a result of repair, and we denote the acceptance set on the entrance E for AF by $acc(AF, E)$.

We address the following problems:

1. When an AF is unstable, is it possible to repair it?
2. If so, how many solutions are possible, and where are the entrances?

In this section, we discuss the case in which an AF includes TUs sharing their nodes.

First, we discuss the case in which an AF includes only one TU.

When the TU has a faucet, the output part can be labeled without using undec if TU can be labeled without using undec. Therefore, we investigate only the case without a faucet.

[1] The meaning of “repair” is not exactly the same as that used in [4].

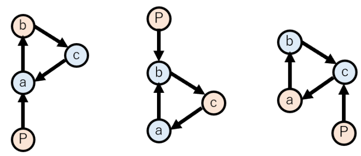


Fig. 3. An AF with one TU.

3.1 No connector

If a TU has no connector, it is a start-TU, and it is trivial that a start-TU is unstable.

Proposition 1. *If AF includes only one start-TU, then it is unstable.*

In this case, we can repair it.

Proposition 2. *If an AF has only one start-TU, then we can repair it by taking any node as an entrance, yielding three solutions.*

Proof. Let AF be a triangular unit $t(a, b, c)$. Assume that we add a whisker node P to a without losing generality. Then, a becomes a connector of this TU. We get a labeling $\mathcal{L}$ such that $\mathcal{L}(P) =$ in, $\mathcal{L}(a) =$ out, $\mathcal{L}(b) =$ in, $\mathcal{L}(c) =$ out, and $acc(AF, a) = \{P, b\}$. □

vin

We show three solutions in Figure 3. Hereafter, in the figures, the pink nodes and blue nodes show the arguments labeled in and out, respectively.

3.2 One connector

When a TU has one connector, the AF has a stable extension depending on the topology. We divide the AF into the TU and the input part to the connector (both of which share the connector), and we consider labeling in each sub-AF. If two labelings can give the same value to the connector, then the AF is stable.

Proposition 3. *Let AF be an AF that includes a TU $t(a, b, c)$ with the unique connector a. Let $\mathcal{L}$ and $\mathcal{L}_C$ be labelings of the AF and the input part to a, respectively. Then, $\mathcal{L}_C(a) =$ in iff AF has no stable extension.*

Proof. ($\Rightarrow$) Assume that $\mathcal{L}_C(a) =$ in. Then, $\mathcal{L}(a)$ should be in, and $\mathcal{L}(b) =$ out, $\mathcal{L}(c) =$ in, which means that (c, a) is an attack from the node labeled in to the node labeled in. This is a contradiction.

($\Leftarrow$) Assume that $\mathcal{L}_C(a) =$ out. Then, $\mathcal{L}(a)$ should be out. $\mathcal{L}(a)$ is out, regardless of the value of $\mathcal{L}(c)$, from the definition of stable labeling. Therefore, there exists a labeling $\mathcal{L}(a) =$ out, $\mathcal{L}(b) =$ in, $\mathcal{L}(c) =$ out, which means that there exists a stable extension. □

If we add a whisker to an arbitrary node a in the TU, then that node becomes a connector. Therefore, if the AF has no stable extension, then we add a whisker to any node in the TU or any node in the input part so that $\mathcal{L}_C(a) = \text{out}$ holds.

Proposition 4. *Let AF be an unstable AF that includes a TU $t(a,b,c)$ with the unique connector a. Let $\mathcal{L}_C$ be labelings of the input part to a. If there is no branch in the input part to a, that is, no node in the input part is attacked by more than one node, then we can repair it if we take a node in $\mathcal{T}$ or any node x in the input part such that $\mathcal{L}_C(x) = \text{in}$ holds as an entrance, and there are at least three solutions.*

Proof. Let $\mathcal{L}'$ be a labeling of the AF obtained by adding a whisker.

Assume that we add a whisker to a. Then, $\mathcal{L}'(a) = \text{out}$, and then, $\mathcal{L}'(b) = \text{in}$, $\mathcal{L}'(c) = \text{out}$, which is stable.

Assume that we add a whisker to an arbitrary node x in the input part such that $\mathcal{L}_C(x) = \text{in}$ holds. Let $\mathcal{L}'_C$ be a label of the input part to a after the whisker is added. Then we get $\mathcal{L}'_C(x) = \text{out}$ and $\mathcal{L}'_C(a) = \text{out}$. Therefore, we have $\mathcal{L}'(a) = \text{out}$, $\mathcal{L}'(b) = \text{in}$, $\mathcal{L}'(c) = \text{out}$, which is stable. Note that if there is a branch, then we need more than one whisker to make $\mathcal{L}_C(a) = \text{out}$, depending on the number of edges from the connector to the branching point, and there is no solution by adding only one whisker to the input part other than the connector.

Assume that we add a whisker to b. Then, $\mathcal{L}'(b) = \text{out}$, and then, $\mathcal{L}'(c) = \text{in}$, $\mathcal{L}'(a) = \text{out}$, which is stable.

Assume that we add a whisker to c. Then, $\mathcal{L}'(c) = \text{out}$, and $\mathcal{L}'(a) = \text{in}$, $\mathcal{L}'(b) = \text{out}$, which is stable.

Conversely, if we add a whisker to the other nodes, then we cannot repair the AF.

Therefore, there are at least three solutions. □

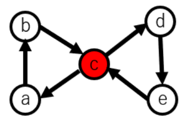
3.3 *k* connectors

We generalize the case in which the number of connectors is k ($k = 0, 1, 2, 3$).

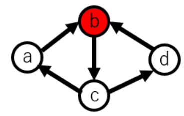
Theorem 1. *Let AF be an AF that includes a TU $t(a,b,c)$. Let $\mathcal{L}_A$, $\mathcal{L}_B$ and $\mathcal{L}_C$ be labelings to input part to a, input part to b, and input part to c, respectively. Then, $\mathcal{L}_A(a) = \mathcal{L}_B(b) = \mathcal{L}_C(c) = \text{in}$ iff AF has no stable extension.*

Proof. ($\Rightarrow$) We show that we cannot define a stable labeling $\mathcal{L}$ to the AF. Assume that $\mathcal{L}(a) = \text{in}$. As such, $\mathcal{L}(b) = \text{out}$, and then, $\mathcal{L}(c) = \text{in}$. This indicates an attack from the node labeled in to the one labeled in, which is a contradiction. Assume that $\mathcal{L}(a) = \text{out}$. Then, $\mathcal{L}(b) = \mathcal{L}_B(b) = \text{in}$, $\mathcal{L}(c) = \text{out}$, and $\mathcal{L}(a) = \mathcal{L}_A(a) = \text{in}$, which is a contradiction. Hence, AF has no stable extension.

($\Leftarrow$) We prove the contraposition by assuming that $\mathcal{L}_A(a) = \text{out}$, without losing generality. Assume that $\mathcal{L}(a) = \text{out}$. Then, $\mathcal{L}(b) = \text{in}$, and then, $\mathcal{L}(c) = \text{out}$ and $\mathcal{L}(a) = \mathcal{L}_A(a) = \text{out}$, which is consistent. Therefore, AF has a stable extension. □



(a) butterfly type



(b) diamond type

Fig. 4. AF with two TUs.

If AF has no stable extension, we can repair it.

Theorem 2. *Let AF be an AF that includes a TU $t(a, b, c)$. Let $\mathcal{L}_A$, $\mathcal{L}_B$, and $\mathcal{L}_C$ be labelings to input part to a, input part to b, and input part to c, respectively. If the AF is unstable and each input part has no branch, then we can repair it iff we take any of the following nodes as an entrance:*

1. *any node x of input part to a such that $\mathcal{L}_A(x) = \text{in}$.*
2. *any node y of input part to b such that $\mathcal{L}_B(y) = \text{in}$.*
3. *any node z of input part to c such that $\mathcal{L}_C(z) = \text{in}$.*

There are at least three solutions.

Proof. This can be proven, similarly to Proposition 4. □

4 Triangular Units Sharing Nodes

4.1 AF including two triangular units

Assume that AF includes two TUs that share their nodes. For simplicity, we assume that each node in AF is included in at least one TU.

There are two topologies, depending on the number of nodes shared with the two TUs. If only one node is shared, we call it the Butterfly type (B-type), and if two nodes are shared, the Diamond type (D-type) (Figure 4). In both types, TUs have only one common connector. In B-type (Figure 4(a)), both TUs have the common connector c, and in D-type (Figure 4(b)), both TUs have the common connector b. The connectors are shown as red nodes in the figures. A node that is not a connector is attacked by exactly one node. Neither of these AFs has a stable extension, and we can repair them. We have three solutions in the case of B-type (Figure 5) and two solutions in the case of D-type (Figure 6).

4.2 AF including three triangular units

Assume that AF includes three TUs that share their nodes. For simplicity, we assume that each node in the AF is included in at least one TU. There are three types of topology: BB-type (Figure 7), BD-type (Figure 10), and DD-type (Figure 14), depending on the types sharing nodes. None of these AFs have a stable extension, and we can repair them.

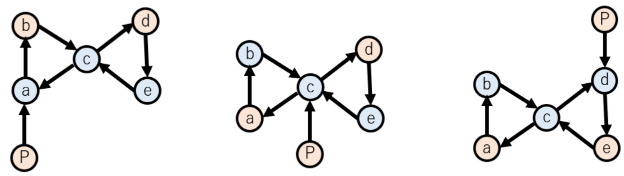


Fig. 5. Solutions of B-type: $ent(AF) = \{a, c, e\}$.

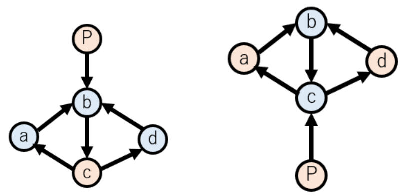
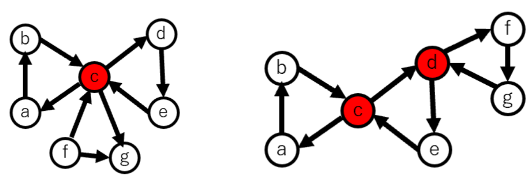


Fig. 6. Solutions of D-type: $ent(AF) = \{b, c\}$.

In this case, there exist one or two connectors. If two connectors exist, the position of the entrance is determined depending on the direction of an attack between the connectors. For each topology, we show that we can repair it by adding a whisker so that all the connectors are labeled out.

In the BB-type, each pair of TUs shares a single node. There are two topologies of the BB-type (Figure 7).



(a) BB1 (b) BB2

Fig. 7. BB-type.

For the BB1-type, there are four solutions: $ent(AF) = \{c, d, a, g\}$ (Figure 8), and for BB2-type, there are two solutions: $ent(AF) = \{d, f\}$ (Figure 9).

In the BD-type, a pair of TUs share a single node, and another pair of TUs share an edge. There are three BD-type topologies (Figure 10).

For the BD1-type, there are three solutions: $ent(AF) = \{b, c, e\}$ (Figure 11), for the BD2-type, there are two solutions: $ent(AF) = \{c, e\}$ (Figure 12), and for the BD3-type, there is one solution: $ent(AF) = \{b\}$ (Figure 13).

In the DD-type, two pairs of TUs share an edge. There are three DD-type topologies (Figure 14).

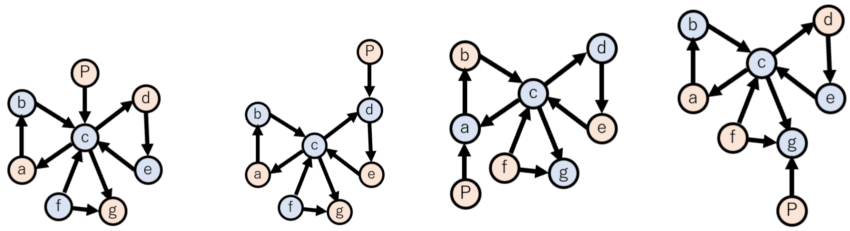


Fig. 8. Solutions of BB1: $ent(AF) = \{c, d, a, g\}$.

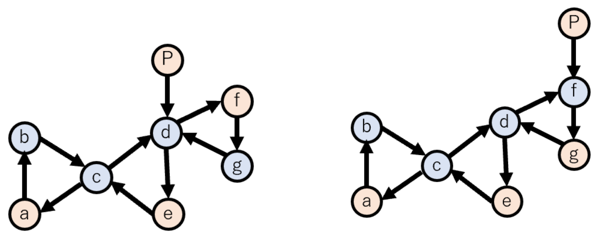


Fig. 9. Solutions of BB2: $ent(AF) = \{d, f\}$.

For DD1-type, there are two solutions: $ent(AF) = \{b, c\}$ (Figure 15), for DD2-type, there is one solution: $ent(AF) = \{b\}$ (Figure 16), and for DD3-type, there is one solution: $ent(AF) = \{c\}$ (Figure 17).

4.3 AF including k triangular units

From the investigation in Subsection 4.1 and Subsection 4.2, we show that the positions of the entrances can be determined generally for any topology presented in these previous sections.

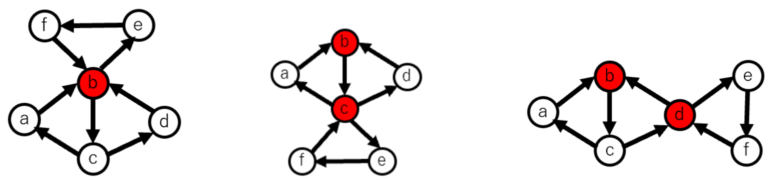
We restrict the target AF to the one that satisfies the following conditions, since we want to clarify the properties of a trilemma itself, avoiding the AF that does not frequently appear in a practical argumentation from our target.

Definition 13 (module). *We call the AF that satisfies the following conditions* **Cond** a module*.*

[Cond]

1. *The AF consists of at most three TUs sharing their nodes.*
2. *Each node in the AF is included in at least one TU.*
3. *It has no cycle other than TUs.*
4. *It is uncontroversial.*

Theorem 3. *1. A module AF has no stable extension.*



(a) BD1 (b) BD2 (c) BD3

Fig. 10. BD-type.

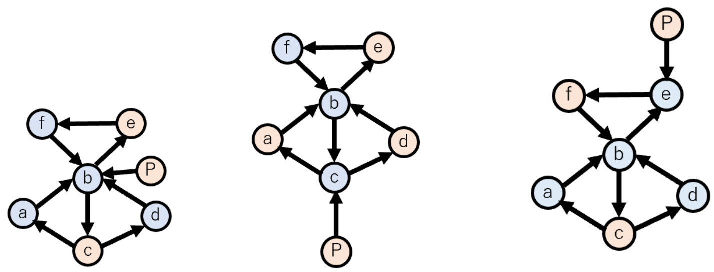


Fig. 11. Solutions of BD1: $ent(AF) = \{b, c, e\}$.

2. *When $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ has one connector C, it can be repaired iff the common connector C or any node A that satisfies $(C, A) \in \mathcal{R}$ is taken as an entrance.*
3. *When $AF = \langle \mathcal{A}, \mathcal{R} \rangle$ has two connectors C_A and C_B such that $(C_A, C_B) \in \mathcal{R}$, it can be repaired iff C_B or any node B that satisfies $(C_B, B) \in \mathcal{R}$ and $(B, C_A) \notin \mathcal{R}$ is taken as an entrance.*

Proof. 1. Since any TU included in AF is unstable, AF is unstable.

2. Assume that AF has one connector C. Let an arbitrary TU in AF be $t(A, B, C)$.
   (a) If we add a whisker to the node C, then C is labeled out. Node A is attacked only by C in AF. Therefore, if C is labeled out, then A is labeled in. Then, B, which is attacked by A, should be labeled out. All TUs can be labeled similarly, since the connector is common. Therefore, AF can be repaired.
   (b) If we add a whisker to the node A such that $(C, A) \in \mathcal{R}$, then A is labeled out, B attacked only by A is labeled in and as a result, C is labeled out. Therefore, AF can be repaired.
   (c) In contrast, assume that we add a whisker to B. Then B is labeled out. Let $t'(C, A', B')$ be another TU that shares the connector C. If C is labeled out, then A' is labeled in, since A' is attacked only by C; B' is labeled out, since B' is attacked only by A'; and C is labeled in, since both its attackers B and B' are labeled out, which is a contradiction. If C is labeled in, then A' is labeled out and B' is labeled in, and then C should be out, which is a contradiction.

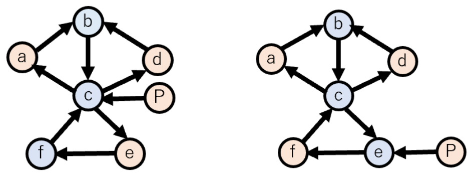


Fig. 12. Solutions of BD2: $ent(AF) = \{c, e\}$.

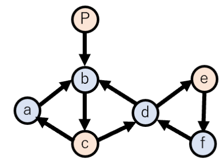
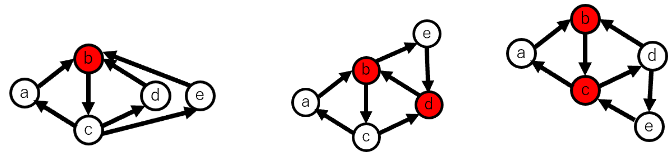


Fig. 13. Solution of BD3: $ent(AF) = \{b\}$.



(a) DD1 (b) DD2 (c) DD3

Fig. 14. DD-type.

3. Assume that AF has two connectors C_A and C_B such that $(C_A, C_B) \in \mathcal{R}$. Let $t(A, C_A, C_B)$ be an arbitrary TU in AF that has two connectors, and $t(C_A, D, E)$ be an arbitrary TU in AF that has one connector C_A.
   (a) If we add a whisker to C_B, then C_B is labeled out. Node A is attacked only by C_B. Therefore, if C_B is labeled out, then A is labeled in. Then, C_A which is attacked by A, should be labeled out. As for $t(C_A, D, E)$, node D is labeled in since it is attacked only by C_A, and node E is labeled out since it is attacked only by D. A TU that shares only C_B can be labeled without a contradiction for the same reason as that in the case of one connector. Therefore, AF can be repaired.
   (b) If we add a whisker to the node B that satisfies $(C_B, B) \in \mathcal{R}$ and $(B, C_A) \notin \mathcal{R}$, then B is labeled out. Let $t(C_B, B, F)$ be a TU that has only one connector C_B. Then B is labeled out, regardless of the label of its other attacker. And F, attacked only by B, is labeled in, and C_B is labeled out. Therefore, AF can be repaired.
   (c) In contrast, assume that we add a whisker to C_A. Then, C_A is labeled out. Let $t(C_B, B, F)$ be another TU that has only one connector C_B. If C_B is labeled in, then B is labeled out, and then F is labeled in. Therefore, C_B is labeled out, since C_B is attacked by F, which is a contradiction. If C_B is labeled out, then B is labeled in, and then F is labeled out. Therefore, C_B should be in, since C_B is attacked by F and C_A, both of which are labeled out, which is a contradiction. When we add a whisker to the other node, a similar discussion follows. $\square$

This theorem shows that we can find an entrance by simply checking the topology of an AF.

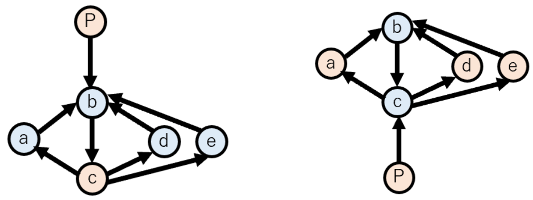


Fig. 15. Solutions of DD1: $ent(AF) = \{b, c\}$.

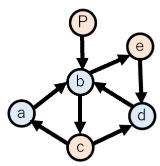


Fig. 16. Solution of DD2: $ent(AF) = \{b\}$.

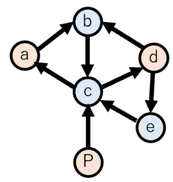


Fig. 17. Solution of DD3: $ent(AF) = \{c\}$.

Example 2. The BB2-type AF $\langle \mathcal{A}, \mathcal{R} \rangle$, shown in Figure 7(b), has connectors c and d such that $(c, d) \in \mathcal{R}$, $(d, f) \in \mathcal{R}$, and $(f, c) \notin \mathcal{R}$ hold. Therefore, d and f are entrances (Figure 9).

In contrast, the BD3-type AF $\langle \mathcal{A}, \mathcal{R} \rangle$, shown in Figure 10(c), has connectors b and d such that $(d, b) \in \mathcal{R}$, $(b, c) \in \mathcal{R}$, and $(c, d) \in \mathcal{R}$ hold. Therefore, b is an entrance, but c is not (Figure 13).

Theorem 3 does not always hold if the AF consists of more than three TUs. For example, the AF shown in Figure 18, which has three connectors, c, e and f, cannot be repaired.

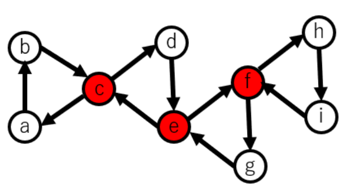


Fig. 18. AF that cannot be repaired.

4.4 Acceptance of a specific argument

We have discussed the entrances to obtain a stable AF. The next question is whether we can find a solution in which a specific argument can be accepted, that is, find an entrance on which an acceptance set includes the argument. The above investigation shows that it is impossible to make the connector be an accepted argument in the case of a trilemma. How about the arguments other than the connector? Unfortunately, it is impossible to ensure that some specific arguments will be accepted, even if any position is selected as an entrance in some topology. For example, see the solutions for BB2-type shown in Figure 9. In this case, $\neg\exists E; b \in acc(AF, E)$, and we have to add more than one whisker to the graph to make b accepted.

4.5 N-lemma

Theorem 3 holds not only for a trilemma but also N-lemma for any finite $N = 2m + 1$.

Definition 14 (odd-unit). *An AF of the form* $\langle\{A_1, \ldots, A_{2m+1}\}, \{(A_1, A_2), (A_2, A_3), \ldots, (A_{2m}, A_{2m+1}), (A_{2m+1}, A_1)\}\rangle$ *is called* an odd-unit *and is denoted by* $t(A_1, \ldots, A_{2m+1})$.

We set the **Cond_N** by replacing the term 'TU' in **Cond** by 'odd-unit', and obtain the following theorem.

Theorem 4. *1. The AF that satisfies* **Cond_N** *has no stable extension.*

2. *When $AF = \langle\mathcal{A}, \mathcal{R}\rangle$ has one connector C, it can be repaired iff the common connector C or any node B such that there exists a sequence of attacks $(B_i, B_{i+1}) \in \mathcal{R}$ ($1 \leq i \leq 2s - 1$) where $B_1 = C$ and $B_{2s} = B$ is taken as an entrance.*
3. *When AF has two connectors C_A and C_B such that there exists a sequence of attacks $(A_i, A_{i+1}) \in \mathcal{R}$ ($1 \leq i \leq 2h - 1$) where $A_1 = C_A$ and $A_{2h} = C_B$. Then, it can be repaired iff a node B that satisfies one of the following conditions is taken as an entrance:*
   - **(i)** $B = C_B$
   - **(ii)** *B is a node of an odd-unit including both C_A and C_B and there exists a sequence of attacks $(B_i, B_{i+1}) \in \mathcal{R}$ ($1 \leq i \leq 2s - 1$) where $B_1 = C_A$ and $B_{2s} = B$.*
   - **(iii)** *B is a node of an odd-unit including C_B but not C_A, and there exists a sequence of attacks $(B_i, B_{i+1}) \in \mathcal{R}$ ($1 \leq i \leq 2s - 1$) where $B_1 = B$ and $B_{2s} = C_B$ and each B_i ($1 \leq i \leq 2s - 1$) is not shared with the other odd-units.*

The theorem can be proved using the properties that all the nodes but for the connectors are attacked only by one node, respectively, and that the label of the entrance is always out.

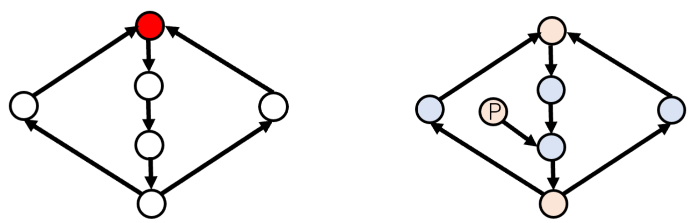
Sketch of Proof.

Due to space constraints, here we discuss the case in which AF has one connector. Let $t(A_1, \ldots, A_{2m+1})$ be an arbitrary odd-unit.

Assume that it shares an odd number of nodes $A_1, A_2, \ldots, A_{2t-1}$ where $(A_i, A_{i+1}) \in \mathcal{R}$ $(\forall i; 1 \leq i \leq 2t, t \leq m)$. Then, $A_1, A_3, \ldots, A_{2t-1}$ have the same label, since each of them is attacked only by one node. Thus, we can consider labeling by reducing $A_1, A_2, \ldots, A_{2t-1}$ to one node. Then, the number of shared nodes can be considered as one. Similarly, since non-shared nodes $A_{2t}, A_{2t+2}, \ldots, A_{2m}$ have the same label, we can consider labeling by reducing $A_{2t}, A_{2t+1}, \ldots, A_{2m}$ to one node. Then, the number of the non-shared nodes can be considered as two. As a result, the problem is reduced to the one in the case of a trilemma.

If an odd-unit $t(A_1, \ldots, A_{2m+1})$ shares an even number of nodes $A_1, A_2, \ldots, A_{2t}$, the problem is reduced to the one of a trilemma consisting of two shared nodes and one non-shared node. $\square$

Note that different from the case of a trilemma, the connector is not necessarily labeled out when repaired (Figure 19).



(a) unstable AF including 5-lemmas (b) labeling when a whisker is added

Fig. 19. Repaired AF in which the connector is not labeled in.

5 Connected Modules

In this section, we consider an AF consisting of modules connected by edges that are not included in any TU.

Definition 15 (meta-AF). *Let $\mathcal{M}$ be a set of modules $\{M_1, \ldots, M_k\}$, where $M_i = \langle \mathcal{A}_i, \mathcal{R}_i \rangle$. Let $AF(\mathcal{M})$ be an AF $\langle \mathcal{A}, \mathcal{R} \rangle$ where $\mathcal{A} = \bigcup_{i=1}^{k} \mathcal{A}_i, \mathcal{R} = \bigcup_{i=1}^{k} \mathcal{R}_i \cup \{(A_i, B_j) | A_i \in \mathcal{A}_i, B_j \in \mathcal{A}_j (1 \leq i \neq j \leq k)\}$. Then, $AF(\mathcal{M})$ is said to be* a meta-AF *of $\mathcal{M}$.*

We define the terms connector, faucet, and start-module for meta-AF, similarly with the case of a single triangular unit.

Definition 16 (connector, faucet, start-module). *Let $\mathcal{M}$ be a set of modules $\{M_1, \ldots, M_k\}$, where $M_i = \langle \mathcal{A}_i, \mathcal{R}_i \rangle$. Let $AF(\mathcal{M})$ be a meta-AF of $\mathcal{M}$.*

If $(A, B) \in \mathcal{R}, A \in \mathcal{A} \setminus \mathcal{A}_i, B \in \mathcal{A}_i$, then B is said to be a connector *of M_i; if $(A, B) \in \mathcal{R}, A \in \mathcal{A}_i, B \in \mathcal{A} \setminus \mathcal{A}_i$, then A is said to be* a faucet *of M_i.*

$conn(M_i)$ and $faucet(M_i)$ denote the set of connectors and that of faucets of M_i, respectively. A module without a connector is said to be a start-module.

For simplicity, we assume that no module is isolated and that each pair of modules is connected by at most one edge. Note that $\{M_1, \ldots, M_k\}$ is not necessarily connected linearly. Each M_i may have more than one connector and/or faucet.

Example 3. Figure 20 shows an AF that consists of four modules. M_1 is a start-module. The connector and faucets of these modules are: $conn(M_1) = \{\}$, $faucet(M_1) = \{d, f\}$, $conn(M_2) = \{h\}$, $faucet(M_2) = \{j\}$, $conn(M_3) = \{p\}$, $faucet(M_3) = \{q\}$, and $conn(M_4) = \{k, m\}$, $faucet(M_4) = \{\}$.

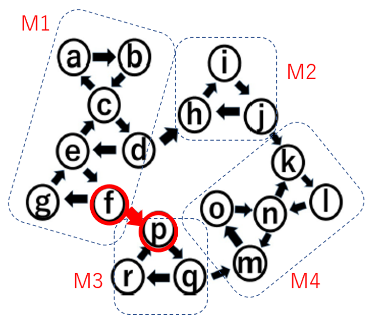


Fig. 20. Connected modules without a meta-cycle.

$AF(\mathcal{M})$ is unstable since each module is unstable.

Proposition 5. *$AF(\mathcal{M})$ is unstable.*

Let $\mathcal{M}$ be a set of modules $\{M_1, \ldots, M_k\}$, where $M_i = \langle \mathcal{A}_i, \mathcal{R}_i \rangle$. Let $AF(\mathcal{M})$ be a meta-AF of $\mathcal{M}$. In the following, we discuss its *repair*.

If there exists more than one start-module, then we have no solution since we have to add a whisker to each start-module so that each of them is stable. If there is one start-module, a configuration of modules should satisfy some condition so that it is possible to be repaired.

Proposition 6. *Let $\mathcal{M} = \{M_1, \ldots, M_k\}$ and $AF(\mathcal{M}) = \langle \mathcal{A}, \mathcal{R} \rangle$. Assume that $AF(\mathcal{M})$ has exactly one start-module.*

We can repair $AF(\mathcal{M})$ by setting the connector E of the start-module as an entrance iff the following three conditions hold for each $M_i \in \mathcal{M}$:

1. *If M_i is a start-module, then $\exists E, F; (E \in ent(M) \wedge F \in acc(M_i, E))$.*

2. *If M_i is not a start-module, let $(F_l, C_i) \in \mathcal{R}$ $(1 \leq l \neq i \leq k)$ where $F_l \in faucet(M_l)$,*
   then $\exists C_i; (C_i \in conn(M_i) \Rightarrow C_i \in ent(M_i) \wedge F_l \in acc(M_l, C_l))$.
3. *If M_i is not a start-module and if $\exists D_i; D_i \neq C_i, D_i \in conn(M_i)$, let $(F_j, C_i) \in \mathcal{R}$ $(1 \leq j \neq i, l \leq k)$ where $F_j \in faucet(M_j)$,*
   then $D_i \in acc(M_i, C_i) \Leftrightarrow F_j \notin acc(M_j, C_j)$.

Sketch of Proof.

First, let M be a start-module. If we add a whisker to an entrance E of M, then we can repair M.

Next, let M not be a start-module. Then, it has a connector.

Assume that M_i $(1 \leq i \leq k)$ has only one connector C_i. From the second condition, C_i is labeled out since the faucet F_l is labeled in, and it is an entrance of M_i. Therefore, M_i has a stable labeling. Let $\mathcal{L}$ be this labeling.

Assume that M_i $(1 \leq i \leq k)$ has a connector D_i different from C_i. From the third condition, $\mathcal{L}(D_i) =$ in iff $\mathcal{L}(F_j) =$ out, and $\mathcal{L}$ is a consistent labeling to M_i.

Therefore, $AF(\mathcal{M})$ can be repaired. □

Example 4. Consider the AF shown in Figure 20.

For a start-module M_1, it is a BB2-type module where $ent(M_1) = \{a, c\}$ and $acc(M_1, c) = \{a, d, f\}$. Since $c \in ent(M_1)$ and $d \in acc(M_1, c)$, the first condition is satisfied.

For M_2, $ent(M_2) = \{h, i, j\}$ and $acc(M_2, h) = \{i\}$. For the connector h of M_2 where $(d, h) \in \mathcal{R}$, $h \in ent(M_2) \wedge d \in acc(M_1, c)$ holds. For M_3, $ent(M_3) = \{p, q, r\}$ and $acc(M_3, p) = \{q\}$. For the connector p of M_3 where $(f, p) \in \mathcal{R}$, $p \in ent(M_3) \wedge f \in acc(M_1, c)$ holds. For M_4, $ent(M_4) = \{k, m, n\}$ and $acc(M_4, m) = \{k, o\}$. For the connector m of M_4 where $(q, m) \in \mathcal{R}$, $m \in ent(M_4) \wedge q \in acc(M_3, p)$ holds. Therefore, the second condition is satisfied.

M_4 has one more connector k where $(j, k) \in \mathcal{R}, j \in faucet(M_2)$. $j \notin acc(M_2, h) \wedge k \in acc(M_4, m)$ holds. Therefore, the third condition is satisfied.

Hence, we can repair it by adding a whisker node P to c, and the obtained stable extension is $\{P, a, d, f, i, k, o, q\}$.

Example 5. For the AF in Figure 20, assume that an attack (f, p) is replaced by (f, r).

For M_3, $conn(M_3)$ is changed to $\{r\}$, and $acc(M_3, r) = \{p\}$. As a result, in M_4, $q \notin acc(M_3, r)$, which breaks the second condition.

Therefore, we cannot repair it.

Proposition 6 can be extended for an $AF(\mathcal{M})$ that has a meta-cycle. In this case, we must consider the connections between modules.

Example 6. Figure 21 presents an AF consisting of four modules that constitute a meta-cycle. In this case, we regard an arbitrary module as a start-module.

We take M_1 as a start-module where $ent(M_1) = \{a, c\}$ and $acc(M_1, a) = \{b, d, f\}$. Since $a \in ent(M_1) \wedge d \in acc(M_1, a)$, the first condition is satisfied.

It can be checked that the second condition is satisfied for each module.

In this case, it can be considered that M_1 has a connector f in addition to a, where $(p, f) \in \mathcal{R}$, $p \in faucet(M_4)$. And $p \notin acc(M_4, q) \wedge f \in acc(M_1, a)$ holds. Therefore, the third condition is satisfied.

Therefore, we can repair it by adding a whisker node P to a, and the obtained stable extension is $\{P, b, d, f, i, l, m, r\}$.

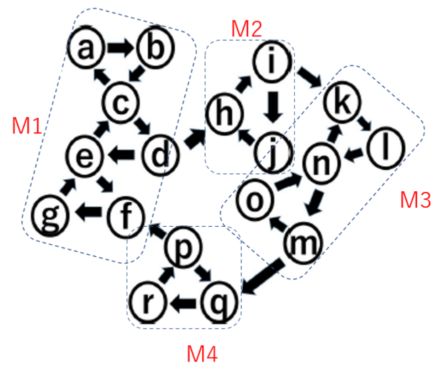


Fig. 21. Connected modules with a meta-cycle.

6 Related Works

In general, the main issue in changing an argumentation framework is the possibility of modification so that a set of arguments becomes a subset of an extension. This issue was introduced as an *enforcing problem* and was first discussed in [4]. Subsequently, considerable work has been done on this problem [14].

Boella et al. discussed the change in grounded semantics if we add or remove an attack relation [8, 9]. They investigated the properties of the grounded extensions, such as expansive change or narrowing change.

Cayrol et al. expanded this discussion to several kinds of semantics including stable semantics. They investigated the properties of the change in extensions with regard to the addition and removal of an argument with an attack. They first investigated a single attack and then extended the procedure to addition and removal of multiple attacks [11, 10, 12]. They showed that some propositions depend on the changing type of extensions, but they did not address the classification of topological features, and not all topological patterns were covered.

Coste-Marquis et al. addressed the revision of extension on changing an attack relation between existing arguments as well as adding an argument with an attack [13].

Alfono et al. developed an efficient algorithm to compute the extension of the revised AF by adding an attack between existing arguments [1].

Baumann et al. showed the minimal change required in an extension to accept a given set of arguments [4], specifically, the change in extensions under several semantics for addition and removal of arguments and attacks [6, 7]. They investigated the change in extensions in various cases. The complexity for the revision was also discussed [5, 19].

These works focused mainly on how to find a solution to realize a minimal change in an extension and the type of properties involved in changes in extensions. In contrast, we did not focus on the properties of changes in extensions. Instead, we investigated the position to which an attack from a new argument is added. Specifically, we considered the AF consisting of TUs that has no stable extension and discussed the problem of how to modify it, depending on the topology. We also attempted to resolve the case of general odd-length cycles.

Some works have utilized the topological features of an argument graph for the treatment of dynamic argumentation frameworks [16, 3]. They used simple topological features such as symmetry and similarity to reduce the complexity of computing changes in extensions, whereas we investigated the relationship of the topological feature and the possibility of repair.

A repair shown in our work can be regarded as an abduction in logic programming, in the sense of finding a minimal change in the knowledge base by adding a fact and a rule. Šefránek described the relationship between a dynamic argumentation framework and revision of logic programming [18]. It would be interesting to relate our approach to an abduction of logic programming.

7 Conclusion

We investigated the conditions under which an unstable AF consisting of a triangular unit can be revised to be stable by adding a new attack from a new argument. We have shown the positions to be added and the number of solutions.

The main contribution of our work is showing a uniform treatment of a trilemma in AFs using its topological features. We also discussed how the result can be extended in two possible directions. One is an extension from the trilemma to N-lemma for any odd-number $N \geq 3$, and the other is the case in which multiple triangular units are connected by edges that are not included in any triangular unit.

The results suggest that we can use topological features, such as connection patterns and the direction of edges, to obtain a stable AF. It follows that when a large argumentation falls into a sticky state, the position to which a counter-argument should be added can be detected by checking the topology of the AF.

Three main problems remain for future research. First, we should investigate the case in which a module includes more than three trilemmas (or N-lemmas). Second, we would like to explore other types of topology, such as those including even-length cycles and other semantics. Third, we plan to discuss the complexity of finding a position. It is not expensive to detect the connectors and the entrances for each topology, but a high computational cost may be incurred to identify its topology.

References

1. Alfaso, G.; Greco, S.; and Parisi, F. 2021. Incremental Computation in Dynamic Argumentation Frameworks. *IEEE Intell. Syst.* 36(2): 6–12.
2. Baroni, P.; Caminada, M.; and Giacomin, M. 2011. An Introduction to Argumentation Semantics. *Knowl. Eng. Rev.* 26(4): 365–410.
3. Baroni, P.; Giacomin, M.; and Liao, B. 2014. On Topology-Related Properties of Abstract Argumentation Semantics. A Correction and Extension to Dynamics of Argumentation Systems: A Division-Based Method. *Artificial Intelligence* 104–115.
4. Baumann, R.; and Brewka, G. 2010. Expanding Argumentation Frameworks: Enforcing and Monotonicity Results. In *COMMA 2010*, 75–86.
5. Baumann, R.; and Ulbricht, M. 2018. If Nothing Is Accepted - Repairing Argumentation Frameworks. In *KR 18*, 108–117.
6. Baumann, R.; and Brewka, G. 2019. Extension Removal in Abstract Argumentation - An Axiomatic Approach. In *AAAI 2019*, 2670–2677.
7. Baumann, R.; Gabbay, D. M.; and Rodrigues, O. 2020. Forgetting an Argument. In *AAAI 2020*, 2750–2757.
8. Boella, G.; Kaci, S.; and van der Torre, L. W. N. 2009a. Dynamics in Argumentation with Single Extensions: Abstraction Principles and the Grounded Extension. In *ECSQARU 2009*, 107–118.
9. Boella, G.; Kaci, S.; and van der Torre, L. W. N. 2009b. Dynamics in Argumentation with Single Extensions: Attack Refinement and the Grounded Extension. In *AAMAS 2009*, 1213–1214.
10. Cayrol, C.; de Saint-Cyr, F. D.; and Lagasquie-Schiex, M.-C. 2010. Change in Abstract Argumentation Frameworks: Adding an Argument. *J. Artif. Intell. Res.* 38: 49–84.
11. Cayrol, C.; de Saint-Cyr, F. D.; and Lagasquie-Schiex, M.-C. 2011. Revision of an Argumentation System. In *KR 2008*, 124–134.
12. Cayrol, C.; and Lagasquie-Schiex, M.-C. 2011. Weighted Argumentation Systems: A Tool for Merging Argumentation Systems. In *ICAI 2011*, 629–632.
13. Coste-Marquis, S.; Konieczny, S; Mailly, J.-G.; and Marquis, P. 2015. Extension Enforcement in Abstract Argumentation as an Optimization Problem. In *IJCAI 2015*, 2876–2882.
14. Doutre, S.; and Mailly, J.-G. 2018. Constraints and Changes: A Survey of Abstract Argumentation Dynamics. *Argument and Computation* 9(3): 223–248.
15. Dung, P. M. 1995. On the Acceptability of Arguments and Its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and N-Person Games. *Artificial Intelligence* 77: 321–357.
16. Liao, B.; Jin, L.; and Koons, R. C. 2011. Dynamics of Argumentation Systems: A Division-Based Method. *Artificial Intelligence* 175(11): 1790–1814.
17. Rahwan, I.; and Simari, G. R., eds. 2009. *Argumentation in Artificial Intelligence*. Springer.
18. Šefránek, J. 2012. Updates of Argumentation Frameworks. In *NMR 2012*.
19. Wallner, J. P., Niskanen, A. and Järvisalo, M. 2017. Complexity Results and Algorithms for Extension Enforcement in Abstract Argumentation. *J. Artif. Intell. Res.* 60: 1–40.

Appendix

Proof of Theorem 4.

1. **[Unstablility]**
   Since any odd-unit included in AF is unstable, AF is unstable.
2. **[One connector]**
   Assume that AF has one connector C (Figure 22(a)).
   If we add a whisker to the node C, then C is labeled out. Then, it is trivial that AF has stable labeling, since C is a common connector.
   Let $t(A_1, \ldots, A_{2m+1})$ be an arbitrary odd-unit. If we add a whisker to one of $A_2, A_4, \ldots, A_{2m}$, then A_{2m+1} is labeled in. Then, C, which is attacked by A_{2m+1}, is labeled out. As a result, $t(A_1, \ldots, A_{2m+1})$ has stable labeling. The other odd-units have stable labelings, since the connector is common.
3. **[Two connectors]**
   Assume that AF has two connectors C_A and C_B (Figure 22(b)).

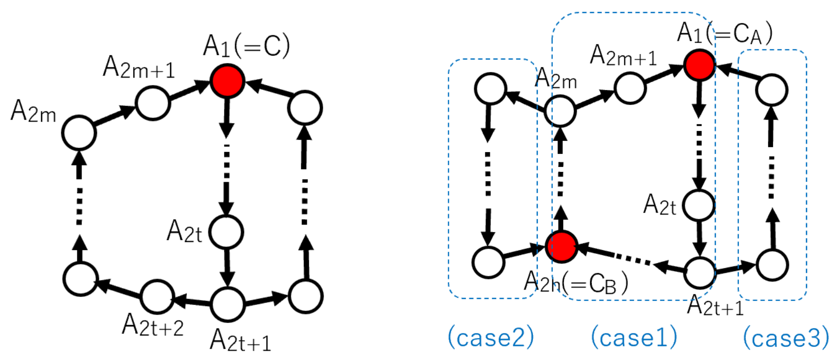
   (case1) Adding a whisker to the odd-unit including two connectors. Let $t(A_1, \ldots, A_{2m+1})$ be the odd-unit that includes two connectors.
   If we add a whisker to C_B, then $C_B(= A_{2h})$ is labeled out. Then, A_{2m+1} is labeled in, since each of $A_{2h+1}, A_{2h+2}, \ldots, A_{2m+1}$ is attacked only by one node. Therefore, $C_A(= A_1)$ attacked by A_{2m+1} is labeled out. Then, A_{2h-1} is labeled in, since $A_2, A_3, \ldots, A_{2h-1}$ is attacked only by one node. Therefore, C_B attacked by A_{2h-1} is labeled out, which is consistent. Moreover, C_B is labeled out, regardless of the label of its other attacker, since it is attacked by A_{2h-1} which is labeled in. Therefore, the odd-unit that includes only one connector C_B has a stable labeling. Similarly, the odd-unit that includes only one connector C_A has a stable labeling. Thus, the entire AF has stable labeling.
   If we add a whisker to one of $A_2, A_4, \ldots, A_{2m}$, then $C_B(= A_{2h})$ is labeled out, since A_{2h-1} is labeled in. Then, C_A is also labeled out, since A_{2m+1} is labeled in. In this case, the other odd-units that share only one connector have stable labelings.
   In contrast, if we add a whisker to one of $A_3, A_5, \ldots, A_{2m+1}$, then A_{2m+1} is labeled out. If $C_A(= A_1)$ is labeled out, then $C_B(= A_{2h})$ should be labeled in. But it is impossible to give a stable labeling to the odd-unit that includes only one connector C_B. If A_{2m+1} is labeled out, then another attacking node to C_A should be labeled in. But it is impossible to give a stable labeling to the odd-unit that includes only one connector C_A.

   (case2) Adding a whisker to the odd-unit including only one connector C_B. Let such an odd-unit be $t(D_1 \ldots, D_{2u+1})$, where $D_1 = C_B$. If we add a whisker to one of $D_2, D_4, \ldots, D_{2u}$, which are not shared with the other odd-units, then D_{2u} is labeled out, since each of these nodes is attacked only by one node. Therefore, D_{2u+1} is labeled in, and then $D_1(= C_B)$, attacked by D_{2u+1}, is labeled out regardless of the label of its other attackers. Then, C_A is labeled out. Thus, AF has a stable labeling.

In contrast, if we add a whisker to one of $D_3, D_5, \ldots, D_{2u+1}$, which are not shared with the other odd-units, D_{2u+1} that attacks C_B is labeled out. If C_B is labeled in, then A_{2m+1} that attacks $A_1(= C_A)$ is labeled out. In this case, it is impossible to give a stable labeling to the odd-unit that includes only one connector C_A. If C_B is labeled out, then A_{2m+1} that attacks $A_1(= C_A)$ is labeled in, and C_A is labeled out, regardless of the label of the other attacker, and A_{2h-1} that is the other attacker of C_B should be labeled out. Then $A_{2h}(= C_B)$ should be in, since both its attackers are labeled out, which is a contradiction.

(case3) Adding a whisker to the node in the odd-unit including only one connector C_A.
If we add a whisker to the node so that C_A is labeled out, then A_{2h-1} that attacks C_B is labeled out. In this case, the other unit that has only one connector C_B does not have a stable labeling.
If we add a whisker to the node so that C_A is labeled in, then C_B is labeled out. Then, A_{2m+1} that attacks $A_1(= C_A)$ is labeled in. Then, C_A should be out, which is a contradiction.

From (case1)–(case3), Theorem 4 holds. □



(a) one connector (b) two connectors

Fig. 22. Repair of N-lemma.