Computing and Informatics, Vol. 38, 2019, 575–596, doi: 10.31577/cai_2019_3_575

DIALOGUE MODEL USING ARGUMENTS FOR CONSENSUS DECISION MAKING THROUGH COMMON KNOWLEDGE FORMATION

Ayslan Trevizan Possebom

Federal Institute of Paraná – Paranavaí
1400, José Felipe Tequinha St.
87703-536 Paranavaí-PR, Brazil
e-mail: possebom@gmail.com

Mariela Morveli-Espinoza, Cesar Augusto Tacla

Federal University of Technology of Paraná
Graduate Program in Electrical and Computer Engineering
3165, 7 September Ave.
80230-901 Curitiba-PR, Brazil
e-mail: morveli.espinoza@gmail.com, tacla@utfpr.edu.br

Abstract. Argumentation plays an important role in reasoning and allows the justification of opinions, especially when applied to collaborative decision making. Reaching consensus is not a trivial task where arguments exchanged in a dialogue and common knowledge are important for consensus. This paper presents a model of argumentative dialogue to support the formation of common knowledge in a group of agents that communicate by sending arguments, and proposes a semantics for consensus decision making. The output of the model is a weighted argumentation graph in which semantics is used to decide the preference of the group.

Keywords: Structured logical arguments, consensus, dialogue, decision making, common knowledge, possible worlds

Mathematics Subject Classification 2010: 68T42

1 INTRODUCTION

Reaching consensus about an issue through discussion with a group of people is not a trivial task [15, 24]. When applied to a group of intelligent agents, this task becomes even more complex, since agents must reason about logically related instructions [1]. In order for consensus decision making in a group to take place, it is necessary to identify the consensus level (or level of acceptance) of the group with regard to the available decision alternatives. The stronger the justifications for supporting or rejecting a particular decision alternative, and the greater the consensus of the group on these justifications, the closer the situation will be to a decision by consensus [25]. The choice of a decision alternative by consensus does not reflect the optimal decision, but rather the one that is preferred by most of the agents.

Consensus is directly related to common knowledge. Agreement on a decision implies common knowledge, and this becomes a prerequisite when a group of agents try to make decisions together [12, 20]. Common knowledge occurs when all agents know an item of information and also know that the other agents in the group know that information [16].

Several methods for group decision making have been proposed in the literature [15, 24], including majority voting, auctions, Borda, Condorcet, and judgment aggregations, among others. These methods do not assume a dialogue between the group members, and each participant only votes on or gives a preference relation for the set of possible alternatives, expressing neither the reasons for these votes nor the conditions for opinion formation. Thus, dialogue becomes an important step before voting, in which all participants can express their opinions and arguments, defending or attacking the alternatives or the information in other arguments presented by other participants. Furthermore, through this dialogue, agents can change their way of thinking based on the arguments presented.

The use of argumentation in multi-agent systems has received a great deal of attention in the last decade. Building arguments allows the agents to reach a collective agreement that is consistent with their beliefs and goals [6, 22]. For a collective decision to be close to unanimity, we need to identify the consensus level of the group on the information in the arguments sent during the dialogue, analysing both the supporting and rejecting relations in each element of the inner structure of those arguments. The information in an argument that is supported (or accepted) by most agents should be consented to by the other agents in the group who do not know it or who reject it. Thus, the supported information becomes consensually accepted by the group in relation to the issue under discussion.

In this paper, we propose a model of dialogue that uses arguments in the messages sent by the agents. Through the information in these arguments, common knowledge is formed based on the majority knowledge of the group. The innovations and contributions of this work are as follows:

1. development of a process of common knowledge identification;

2. identification of the relation between common knowledge and consensus; and
3. development of semantics for consensus decision making.

Thus, the paper has two goals:

1. To present a dialogue model that can be applied to multi-agent systems, where each argument presented needs to be evaluated by the group in an attempt to identify the consensus level on each piece of information presented in the arguments. As a result of this model, common knowledge is formed about the set of information that was accepted or rejected by the group of agents. This common knowledge formation can be used in several application domains where multi-agent systems are used, such as chatbots, sensor networks, ranking of the importance of web pages, identification of simultaneous actions, or any domain in which there is a need for the formation of group opinion.
2. To generate a weighted argumentation graph [2, 5] for each dialogue, so that the decision alternatives can be analysed based on the arguments presented, resulting in a preference relation for the group. The preference relation draws on computation of the strength of an argument, and uses a semantics that considers all the weights of the arguments to determine the preference level for each alternative.

This paper is structured as follows. Section 2 presents the preliminary concepts of possible worlds, common knowledge and structured arguments. Section 3 describes the proposed model of dialogue, covering the structures of the agents, the construction of arguments, the formation of common knowledge, and the consensus decision-making process, including the argument strength, the weighted argumentation graph and semantics for determining the preference relation among the decision alternatives. A practical example, a discussion of the results, and related work are given in Section 4. Finally, we present the conclusions and an outline of planned future work in Section 5.

2 PRELIMINARIES

In this section, we introduce the fundamental background of knowledge representation related to possible worlds and common knowledge, which we use to represent the possible decision alternatives (or issues) that are the subject of dialogue among a group of agents. We also describe the fundamental concepts of arguments and attack relations between arguments, which form the basic structure used by the agents to send messages to the group during the dialogue.

The model proposed in this work considers a virtual environment where each agent in a group, each one with its own knowledge base, is able to act sending messages to the group in a discussion (dialogue) about any issue using logically structured arguments. We model a dynamic process of dialogue that can be used by the agents for choosing the alternative that is consensually justified or for obtaining the order of preference over the available decision alternatives.

2.1 Possible Worlds and Knowledge Representation

The classical model of reasoning about knowledge, as used by a single agent, is known as the possible world model [17]. Possible worlds represent a possible state of affairs (that is, there may be situations in which a belief holds for one issue under discussion, but does not hold for another issue) [12]. Let $AG = \{ag_1, \ldots, ag_n\}$ with $n > 0$ be a finite set of agents. An agent $ag_i \in AG$ believes f if f is necessarily true for ag_i, i.e. it is true in all possible worlds for that agent. The modal operator K_i represents the knowledge of agent ag_i. The formula $K_1 f$ is read as "agent ag_1 knows f", $K_1 K_2 f$ is read as "agent ag_1 knows that ag_2 knows f" and $\neg K_2 K_1 K_3 f$ is read as "agent ag_2 does not know that ag_1 knows that ag_3 knows f".

When the reasoning involves the knowledge of a set of agents, two modal operators can be defined [13]: E_{AG} (where $E_{AG} f$ represents the situation in which every agent in the group knows f) and C_{AG} (where $C_{AG} f$ represents the situation in which every agent in the group knows f and they all know that every agent in the group knows f, i.e. f is common knowledge in the group).

2.2 Structured Logical Arguments

The basis of the proposed dialogue model is the exchange of arguments among agents. When an agent sends a message to the group containing an argument, this argument represents the agent's opinion of, point of view on or justification for the issue under discussion. In this paper, Σ is a knowledge base with formulae (beliefs) in a propositional language, and the arguments are built based on these formulae [4]. In addition, $\vdash$ is the classical inference, $\equiv$ represents logical equivalence, $\perp$ represents contradiction, $\wedge$ conjunction, $\vee$ disjunction, $\neg$ negation, $\rightarrow$ implication, and $\leftrightarrow$ biconditionality. An argument [3, 4, 21] is formed by a pair $\langle \Phi, \alpha \rangle$ where Φ represents the support (premises) and α the claim of the argument, such that

1. α is a formula;
2. $\Phi \subseteq \Sigma$;
3. $\Phi \nvdash \perp$;
4. $\Phi \vdash \alpha$; and
5. $\nexists \Phi' \subseteq \Phi$ such that $\Phi' \vdash \alpha$.

Arguments are created to justify a position against the decision alternative or other arguments. The most common attack relations between arguments are undercut and rebuttal [4, 21]. Let $arg_1 = \langle \Phi_1, \alpha_1 \rangle$ and $arg_2 = \langle \Phi_2, \alpha_2 \rangle$ be two distinct arguments: arg_1 undercuts arg_2 iff $\exists \varphi \in \Phi_2$ such that $\alpha_1 \equiv \neg \varphi$, and arg_1 rebuts arg_2 iff $\alpha_1 \equiv \neg \alpha_2$.

Example 1. Let $\Sigma = \{a, \neg b, a \rightarrow \neg b, d \rightarrow b, a \rightarrow d\}$ be a knowledge base of an agent. Let us consider the following three arguments $arg_1 = \langle \{a, a \rightarrow \neg b\}, \neg b \rangle$, $arg_2 = \langle \{\neg b, d \rightarrow b, a \rightarrow d\}, \neg a \rangle$, and $arg_3 = \langle \{a, a \rightarrow d, d \rightarrow b\}, b \rangle$. We have that arg_2 undercuts arg_1 and arg_3 rebuts arg_1. Figure 1 illustrates these attack relations.

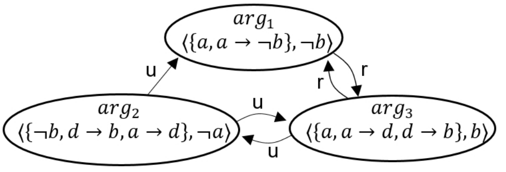


Figure 1. Types of attack relation: undercut (u) and rebuttal (r)

3 COMMON KNOWLEDGE AND CONSENSUS DECISION MAKING

The proposed dialogue model using arguments for common knowledge formation is constructed using agents that play two roles: argumentation and mediation. Argumentative agents are responsible for building arguments and voting, using the beliefs in their respective knowledge bases. Each belief consists of a formula in propositional language. Argumentative agents are also responsible for giving their opinions by voting to support or reject the formulae in arguments sent by the other argumentative agents during the dialogue. The mediator agent is responsible for several other aspects: controlling the message exchange among the argumentative agents, that is, controlling the course of the dialogue; calculating the consensus level on each formula within an argument; informing the group of argumentative agents of which formulae should be accepted during the dialogue; computing the strength of each argument; and informing the group of the outcome of the decision.

For the argumentative and mediator agents, we define a model for common knowledge formation (CKF).

Definition 1. A model for common knowledge formation is a tuple $CKF = \langle AG, EX, ISS, med, t, \sigma\rangle$ where:

- $AG = \{ag_1, \ldots, ag_n\}$ with $n > 1$ is the finite set of argumentative agents;
- $EX = \{ex_1, \ldots, ex_n\}$ with $ex_i \in [0, 1]$ and $\sum_{i=1}^{n} ex_i = 1$ is the set of expertise values for the argumentative agents, such that ag_i has expertise ex_i;
- $ISS = \{iss_1, \ldots, iss_m\}$ with $m > 1$ is the finite set of issues (or decision alternatives) to be discussed;
- med is the mediator agent;
- t is the waiting time used by the mediator to coordinate the message exchange; and
- σ is a threshold value determining when a formula is common knowledge.

3.1 Argumentative Agent

Argumentative agents are responsible for building arguments, and for supporting or rejecting any information used in an argument.

Definition 2. Let AG be the set of argumentative agents. An argumentative agent $ag_i \in AG$ is a tuple $\langle \Sigma_i, S_i, A_i \rangle$, where:

- $\Sigma_i = K_i \cup KO_i$ is the knowledge base, with K_i representing the beliefs that the agent has about the environment, and KO_i representing the beliefs acquired through communication with other agents;
- S_i is the argument base, which is used to store the set of arguments to be sent to the group; and
- A_i is the current argument that is being discussed by the group, which is used to look for counterarguments.

A formula $f \in \Sigma_i$ may be followed by a label, such as $f[iss_i(b), \ldots]$ where $iss_i \in ISS$ is an issue and $b \in [0, 1]$ is the group consensus level on f related to the issue iss_i. Whenever $b > 0$, most of the agents in AG believe f, and this formula is accepted as common knowledge. For $b = 0$, formula f is not accepted by the group as common knowledge since it is rejected or because most agents do not know f.

Example 2. [adapted from [3]] Consider a set of argumentative agents AG deciding whether or not a patient should undergo surgery. Let Bob, $ag_{bob} \in AG$, be an argumentative agent representing a doctor. The formulae in the knowledge base represent the following information: the patient should undergo a surgery (sg), the patient has colonic polyps (a), the patient is at risk of loss of life (b), the patient is experiencing side-effects (c), the patient has cancer (d). Its structure is defined by:

$$\begin{aligned}
K_{bob} &= \{a[sg(1)], \neg c \rightarrow \neg sg[sg(0)], c \rightarrow sg, d \rightarrow b, a \rightarrow d, d \wedge \neg b \rightarrow sg\}, \\
KO_{bob} &= \{\neg b[sg(0.7)], \neg c[sg(0.4)]\}, \\
S_{bob} &= \{\langle \{\neg c, \neg c \rightarrow \neg sg\}, \neg sg \rangle, \langle \{a, a \rightarrow d, d \rightarrow b\}, b \rangle\}, \\
A_{bob} &= \{\langle \{a, a \rightarrow \neg b\}, \neg b \rangle\}.
\end{aligned}$$

The formulae $\neg b$ and $\neg c$ were accepted by the group, and therefore, were considered by ag_{bob} and used to update its KO_{bob} base. Formula a was accepted unanimously, while $\neg c \rightarrow \neg sg$ was not accepted by the group in the dialogue about sg. When ag_{bob} has the opportunity to send arguments, both arguments in S_{bob} will be sent to the group. A_{bob} stores the current argument in the discussion, and this is used as a reference to look for other counterarguments, storing them in S_{bob} when requested.

Let ARG_i be the set of all arguments that can be built from Σ_i and $arg \in ARG_i$ be an argument with $arg = \langle \Phi, \alpha \rangle$. The function $premise(arg)$ returns Φ (a set of formulae in support of arg) and $claim(arg)$ returns α (the formula in the claim of arg). Let F be the set of all formulae in arg obtained from the function $split(arg)$ ($premise(arg) \cup claim(arg)$). Each formula $f \in F$ has a set of atoms obtained from the function $atoms(f)$. These functions are used by the argumentative agents to express support for or rejection of each formula in the argument that is presented to the group in the dialogue.

A formula f in any argument is supported by an argumentative agent when it knows that formula. The argumentative agent rejects f when it knows other formulae with the same atoms, but with no equivalent meaning.

Definition 3. A formula f in an argument arg_1 sent by an agent ag_i is supported by ag_j, with $i \neq j$, iff (i) $\exists arg_2 \in ARG_j | claim(arg_2) \leftrightarrow f$ is a tautology or (ii) $\exists g \in \Sigma_j | atoms(g) = atoms(f)$ and $\mu \leftrightarrow f$ is a tautology. Formula f is rejected iff (i) $\exists arg_2 \in ARG_j | claim(arg_2) \leftrightarrow \neg f$ is a tautology or (ii) $\exists g \in \Sigma_j | atoms(g) = atoms(f)$ and $g \leftrightarrow f$ is not a tautology.

From Definition 3, we can observe that the agent ag_j supports a formula f of an argument sent by ag_i if ag_j has an argument for f (i.e. $\langle \{\Phi\}, f \rangle \in ARG_j$), or if ag_j knows f (i.e. $f \in \Sigma_j$). A rejection occurs when ag_j has an argument for $\neg f$ (i.e. $\langle \{\Phi\}, \neg f \rangle \in ARG_j$), or if ag_j knows a formula g that has the same atoms as f, but g and f are not logically equivalent.

Example 3. [cont. 2] Consider $arg = \langle \{a, a \rightarrow \neg d\}, \neg d \rangle$, which will be analyzed to identify the consensus level on its formulae. Agent ag_{bob} supports a because it believes this formula. Formula $a \rightarrow \neg d$ is rejected because ag_{bob} believes $a \rightarrow d$ and $(a \rightarrow \neg d) \leftrightarrow (a \rightarrow d)$ is not a tautology. Formula $\neg d$ is supported by ag_{bob} because it has an argument for $\neg d$: $\langle \{\neg b, d \rightarrow b\}, \neg d \rangle$ and rejected with argument $\langle \{a, a \rightarrow d\}, d \rangle$.

The possible actions available to all the argumentative agents $ag_i \in AG$ in an argumentative dialogue for a decision by consensus are as follows:

- *discArg*(arg, y): the current argument arg presented in the dialogue is stored in A_i of the argumentative agents along with a number y that denotes its position in the sequence in which the argument was sent to the group;
- *askSpeak*$()$: when $S_i \neq \emptyset$, ag_i informs med that it has some arguments to be sent;
- *propose*$()$: when ag_i is requested to send its arguments, it sends all the arguments in S_i to med and then S_i is emptied;
- *attack*(t): ag_i looks for arguments attacking the argument in A_i, storing them in S_i, in time t;
- *voteSupport*(f, t): ag_i votes to support formula f at time t;
- *voteRejection*(f, t): ag_i votes to reject formula f at time t;
- *learn*(f, b, iss): ag_i updates Σ_i with formula f and the label containing the consensus level b for issue iss. If $f \in \Sigma_i$, then the action $inform(f, b, iss)$ is executed; otherwise, formula $f[iss(b)]$ should be inserted in KO_i;
- *inform*(f, b, iss): agents with $f \in \Sigma_i$ should update f with the label $f[iss(b)]$ only when issue iss is not yet annotated in f;

- $query(ag_j, at)$: agent ag_i can ask ag_j with $i \neq j$ for formulae containing the atom at. This action does not involve the *med* agent and can be performed at any time by the argumentative agents;
- $answer(ag_i, F)$: when an agent ag_j is queried, it replies to ag_i, returning the set of formulae F containing the atom at.

3.2 Mediator Agent

There is a dialogue for each decision alternative and the mediator agent maintains a dialogue table for each dialogue. This table is used to store the sequence of arguments received during the dialogue, the attacks on the arguments and information about the consensus regarding each argument.

Definition 4. The mediator agent *med* is a tuple $\langle WB, AGENDA, DT, \delta\rangle$, where:

- WB is an ordered list of argumentative agents;
- $AGENDA$ is a list that stores all the arguments sent by one agent when requested;
- $DT = \{dt_1, \ldots, dt_m\}$ is the set of dialogue tables where each $dt_i \in DT$ has the arguments sent during the dialogue on the issue iss_i;
- δ is the knowledge base that stores all the formulae of the arguments within a dialogue, with their respective annotations.

WB is used by *med* as a coordination resource that emulates a face-to-face meeting. When an argumentative agent has arguments to send to the group, it asks to speak (action *askSpeak*()) to *med* and waits for a request to send the arguments. This resource ensures that only agents in WB are granted the right to speak. Only the agent at the top of the list at any given moment is able to send its arguments when requested.

When *med* requests the arguments of an argumentative agent, all the arguments received (action *propose*()) are stored in $AGENDA$. Each argument needs to be checked; that is, *med* checks whether the arguments are admitted and whether an argument has already been presented in the current dialogue.

Definition 5. An argument is admitted iff its formulae in Φ are accepted for the current dialogue. A formula f is accepted for the issue iss_i if it does not mention another issue: $\forall atom(f) \notin ISS \setminus \{iss_i\}$. Furthermore, it must satisfy one of the following conditions:

1. it has not been presented in any other arguments in the current dialogue (formula without a label for iss_i); or
2. there is a consensus on it (label $iss_i(b)$ with $b > 0$).

Example 4. [cont. 2] In a dialogue about sg, the argument $\langle\{a[sg(1)], a \rightarrow d\}, d\rangle$ is admitted. There is consensus on a, and $a \rightarrow d$ has not been presented earlier in any argument about the issue sg. On the other hand, the argument $\langle\{\neg c[sg(0.4)], \neg c \rightarrow \neg sg[sg(0)]\}, \neg sg\rangle$ is not admitted. Although there is consensus on $\neg c$, there is no consensus on $\neg c \rightarrow \neg sg$.

Each admitted argument within the $AGENDA$ occupies a row in the current dialogue table. The dialogue table has the following fields: y (a sequential number indicating the sequence in which the arguments are presented), the issuer agent, the admitted argument, the argument being attacked, the sets of supporting and rejecting agents for each formula of the argument, the set of consensus levels for each formula of the argument, and the intrinsic strength of the argument.

After the support and rejection steps (actions *voteSupport* and *voteRejection*), agent *med* computes the consensus level of f and stores it with the corresponding label in δ, informing the group of argumentative agents of whether or not f should be accepted as common knowledge.

Algorithm 1 shows the dialogue model for CKF executed by *med*. Firstly, the structure for a new dialogue is created (line 2, function *newDialogue*(*iss*)) involving the creation of *WB*, *AGENDA*, and the start point for the dialogue $arg = \langle\{T\}, iss\rangle$ is returned (a structure with only the claim to be discussed). The dialogue table for the issue *iss* is initialized and the current line in the table is returned by *updateDT*(*med*, *arg*) (line 3). Then, *med* sends *arg* to the group (line 4) and asks for support and rejection of formula *iss* at time t (lines 5–6). The dialogue table is updated, including the list of agents that supported and rejected formula *iss*, the consensus level and the intrinsic strength for the start point (lines 7–8, functions buf(*iss*) and *is*(*arg*)). The argumentative agents look for counterarguments at time t (line 9). When *med* requests agents with arguments to send, the responses are stored in *agents* (line 10) and *WB* is updated (line 11, function *updateWB*(*agents*)). For each agent in *WB*, the agent in the first position of the queue sends its arguments (line 13, function *requestArgs*(ag_i)), and all arguments received are stored in the *AGENDA* (line 14, function *updateAGENDA*(*argsList*)). Each argument is checked (line 16, function *check*(arg_k)). Only the admitted arguments are stored in *dt* for the current dialogue (line 17), and the group is informed (line 18). Each formula of the argument undergoes a voting process considering a time t (lines 20–21) and receives a consensus level (line 22); the current dialogue table is updated with the agreement and rejection lists and the consensus level for the formula under analysis (line 23); the knowledge base of *med* is updated with the formula and its related label (line 24); *med* informs the group of whether or not the formula should be accepted (lines 25–28); the intrinsic strength is updated in the dialogue table (line 29); and *med* asks the group to look for counterarguments, waiting a time t before asking which agents have arguments to send (lines 30–32). The current dialogue ends when *WB* and *AGENDA* are empty.

The *buf*(f) function acts as a belief update function. It is responsible for computing the consensus level of the group on formula f in an argument, determining which formulae should be accepted as common knowledge. Equation (1) shows how this function is obtained. We refer to ex_i as the expertise value of the agent that

Algorithm 1 Dialogue model for common knowledge formation

Input: the issue to be discussed (decision alternative)
Output: the dialogue table for the issue

```
procedure DIALOGUE(iss)
    arg ← newDialogue(iss)
    y ← updateDT(med, arg)
    action(discArg(arg, y))
    support ← action(voteSupport(iss, t))
    reject ← action(voteRejection(iss, t))
    b ← buf(iss)
    updateDT(y, iss, support, reject, b, is(arg))
    action(attack(t))
    agents ← action(askSpeak())
    updateWB(agents)
    for all ag_i ∈ WB do
        argsList ← requestArgs(ag_i)
        updateAGENDA(argsList)
        for all arg_k ∈ AGENDA do
            if check(arg_k) then
                y ← updateDT(ag_i, arg_k)
                action(discArg(arg_k, y))
                for all f ∈ arg_k do
                    support ← action(voteSupport(f, t))
                    reject ← action(voteRejection(f, t))
                    b ← buf(f)
                    updateDT(y, f, support, reject, b)
                    if b ≥ σ then
                        action(learn(f, b, iss))
                    else
                        action(inform(f, 0, iss))
                updateDT(y, is(arg_k))
                action(attack(t))
                agents ← action(askSpeak())
                updateWB(agents)
```

sent the current argument, and $Support[f]$ and $Reject[f]$ as the set of agents that voted to support or reject formula f, respectively.

$$buf(f) = ex_i + \sum_{ag_j \in Support[f]} ex_j - \sum_{ag_j \in Reject[f]} ex_j. \tag{1}$$

With $buf(f)$ representing the consensus level on f, $iss_k \in ISS$ the issue under discussion, and $ag_i \in AG$ an argumentative agent:

- $buf(f) \geq \sigma$: formula $f[iss_k(buf(f))]$ should be accepted and considered common knowledge;
- $buf(f) < \sigma$: formula $f[iss_k(0)]$ should be updated only when $f \in \Sigma_i$.

3.3 Consensus Decision Making

Each dialogue table contains a finite set of arguments $ARG = \{arg_1, \ldots, arg_z\}$ with $z > 0$ related to an issue. This set is then mapped to an abstract argumentation framework such as the one proposed by Dung [11], formed of a pair $AF = \langle ARG, R \rangle$ where ARG is the set of arguments and R is a binary relation representing attacks between arguments with $R \subseteq ARG \times ARG$. The notation $R(arg_i, arg_j)$ represents the situation in which arg_i attacks arg_j. During the mapping, an undercut is a single attack relation from the attacking to the attacked (i.e. $R(arg_i, arg_j)$), while a rebuttal is a symmetric relation (i.e. $R(arg_i, arg_j)$ and $R(arg_j, arg_i)$). The abstract argumentation framework is represented as a graph in which the arguments are nodes and the attack relations are edges. During mapping, each line of the dialogue table represents an argument (column *arg*). The attack relation (column *att*) is used to link arguments. Other attacks between arguments can exist and these additional edges need to be identified.

The starting point arg_1 in the dialogue table is the main node in an argumentation graph representing the decision alternative. This node is special since it receives only an undercut representing the arguments against the decision alternative. We refer to $ARGS = ARG \setminus \{arg_1\}$ as the set of all arguments removing the main node.

The consensus decision-making process uses two additional phases to find the decision alternative that is most preferred by the group: computation of the strength of the arguments and determination of the extent to which one alternative is preferred to another.

3.3.1 Computing Argument Strength

Arguments have two types of strength: intrinsic and overall strength [8]. The intrinsic strength is a value obtained using the concept of group majority knowledge, which expresses the extent to which an argument is reliable based on its formulae. This type of strength considers the supporting and rejecting votes in each formula of the structured argument sent during the dialogue. The overall strength is a score representing the importance of the arguments when compared to other arguments in an argumentation graph. This type of strength considers the attack relations between arguments in an abstract argumentation graph.

Equations (2) and (3) show how to compute the intrinsic and overall strengths, respectively. Let $length : ARG \to \mathbb{N}$ be a function that returns the number of formulae of an argument $arg_i \in ARG$ and $attack : ARG \to ATT$ with $ATT \subseteq ARG$ be a function that returns the set of arguments that attack arg_i, that is,

$\{arg_j \in ARG | R(arg_j, arg_i)\}$.

$$is(arg_i) = \left(\frac{\sum_{f \in split(arg_i)} buf(f)}{length(arg_i)} + 1 \right) * 0.5, \tag{2}$$

$$os(arg_i) = \frac{is(arg_i)}{1 + \sum_{arg_j \in attack(arg_i)} os(arg_j)}. \tag{3}$$

To solve the entire system of argumentation, we use an iterative method [10] on the set $ARGS$. Let $time_0$ be the initial overall strength calculation for each argument, and $time_s$ the overall strength calculation obtained after the s^{th} iteration. An iteration at $time_s$ computes a new overall strength at $time_{s+1}$ for all arguments in $ARGS$. We refer to $os(arg_i)^s$ as the process of computing the overall strength of arg_i in iteration s. The iteration terminates when $os(arg_i)^s = os(arg_i)^{s+1}$ for all arguments. The result is independent of the processing order of the arguments.

3.3.2 Computing Preference Relations

To determine the preference relations among the decision alternatives, we propose an adaptation to the argument labelling in which the arguments receive a label of "in", "out" or "undec" according to their iterations with other arguments in the argumentation graph [7]. These labels are used to specify the arguments that are accepted (in) or rejected (out), and those that are neither accepted nor rejected (undec) [23].

In this adaptation, the labelling implies that arguments with greater overall strengths (labelled as "in") are acceptable (or partially acceptable) and undermine those arguments with lower overall strengths (labelled as "out") that are attacked by them. Arguments labelled as "undec" are those with identical overall strengths. In this case, we use the intrinsic strength that represents the consensus level to determine the argument most preferred by the group ("in" or "out"). The arguments are designated as "undec" only when both the overall and intrinsic strengths are the same and there is no attacking argument labelled "in".

We define the following functions to obtain the neighbours of an argument $arg_i \in ARGS$: $getAllNeighbors : ARGS \rightarrow NB$ where NB is the set of all neighbours of arg_i (we consider neighbours to be the attackers and attacked arguments with $NB = \{arg_b \in ARGS | R(arg_b, arg_i) \cup R(arg_i, arg_b)\}$); $getLabeledNeighbors : NB \rightarrow LNB$ where LNB is the set of neighbours with an associated label ("in", "out", or "undec"); and $ACC : AF \rightarrow INARGS$ is the set of acceptable arguments of an argumentation graph with the label "in". The argument labelling is a function that assigns a label to each argument in the graph. Argument labelling uses two sets P and Q representing the set of all labelled neighbours with maximum overall strength and the set of all labelled neighbours with equal overall strength, respectively, where $P = \{arg_j \in LNB | os(arg_j) > os(arg_i)\}$ and $Q = \{arg_j \in LNB | os(arg_j) = os(arg_i)\}$.

Definition 6. Let $arg_i \in ARGS$ be an argument and P and Q be the sets of labelled neighbours with maximum and equal overall strengths, respectively. An argument labelling for an argumentation graph is a total function $L : ARGS \rightarrow \{in, out, undec\}$ such that:

1. if $P = \emptyset$ and $Q = \emptyset$, then $L(arg_i) = in$;
2. if $P = \emptyset$ and $(\exists arg_j \in Q)(is(arg_i) = is(arg_j))$, then $L(arg_i) = undec$;
3. if $P = \emptyset$ and $(\exists arg_j \in Q)(is(arg_i) > is(arg_j))$, then $L(arg_i) = in$;
4. if $P = \emptyset$ and $(\exists arg_j \in Q)(is(arg_i) < is(arg_j))$, then $L(arg_i) = out$;
5. if $(\exists arg_j \in P)(L(arg_j) = in)$, then $L(arg_i) = out$;
6. if $(\forall arg_j \in P)(L(arg_j) \in \{out, undec\})$, then $L(arg_i) = in$.

The labelling of all arguments uses an iterative method to calculate the overall strength of the arguments, since the arguments have links between them and when an argument is labelled, its attacks and attackers need to be reviewed. We refer to $L(arg_i)^w$ as the process of labelling argument arg_i in an iteration w. The last iteration occurs when $L(arg_i)^w = L(arg_i)^{w+1}$ for all arguments in $ARGS$.

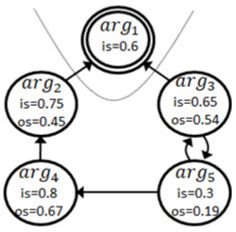
Let $POS : ARGS \rightarrow \{-1, 1\}$ be a function that returns 1 if arg_i is a supporting argument, and -1 otherwise. Supporting arguments are those with even distance over a simple shorter path to the main node in the argumentation graph, while rejecting arguments are those with odd distance over a simple shorter path to the main node in the argumentation graph. To compute the preference level for the decision alternative iss, we need to compute the position of the accepted arguments according to Equation (4). The preferred order relation of two decision alternatives is denoted by the symbols $\succ$ and $\sim$ (preferred or equally preferred, respectively). Whenever $pref(iss_1) > pref(iss_2)$ we have $iss_1 \succ iss_2$; for $pref(iss_1) < pref(iss_2)$ we have $iss_2 \succ iss_1$; and for $pref(iss_1) = pref(iss_2)$ we have $iss_1 \sim iss_2$. In this case, the choice of the preferred decision alternative is random. In the special case when there is no argument in the dialogue, that is, if $ARGS = \emptyset$, then the preference level for the decision alternative is $pref(iss) = is(arg_1)$.

$$pref(iss) = \sum_{arg_y \in INARGS} POS(arg_y) * os(arg_y). \tag{4}$$

Example 5. Consider the argumentation graph and the iterations for labelling in Figure 2. We have $L(arg_2) = out$, $L(arg_3) = in$, $L(arg_4) = in$, $L(arg_5) = out$, $ACC = \{arg_3, arg_4\}$, $POS(arg_3) = -1$, and $POS(arg_4) = 1$. The preference level for the issue is $pref(iss_i) = 0.13$

4 PRACTICAL EXAMPLE

Our example consists of a discussion among three agents that are trying to decide whether a robot should rescue a human being in a disaster situation. The robot has



$time_w$	arg_2	arg_3	arg_4	arg_5
1	$NB = \{arg_4\}$ $LNB = \{\}$ $\Lambda = \{\}$ $\Psi' = \{\}$ $L(arg_2) = in$	$NB = \{arg_5\}$ $LNB = \{\}$ $\Lambda = \{\}$ $\Psi' = \{\}$ $L(arg_3) = in$	$NB = \{arg_2, arg_5\}$ $LNB = \{arg_2\}$ $\Lambda = \{\}$ $\Psi' = \{\}$ $L(arg_4) = in$	$NB = \{arg_3, arg_4\}$ $LNB = \{arg_3, arg_4\}$ $\Lambda = \{arg_3, arg_4\}$ $\Psi' = \{\}$ $L(arg_5) = out$
2	$NB = \{arg_4\}$ $LNB = \{arg_4\}$ $\Lambda = \{arg_4\}$ $\Psi' = \{\}$ $\boldsymbol{L(arg_2) = out}$	$NB = \{arg_5\}$ $LNB = \{arg_5\}$ $\Lambda = \{\}$ $\Psi' = \{\}$ $\boldsymbol{L(arg_3) = in}$	$NB = \{arg_2, arg_5\}$ $LNB = \{arg_2, arg_5\}$ $\Lambda = \{\}$ $\Psi' = \{\}$ $\boldsymbol{L(arg_4) = in}$	$NB = \{arg_3, arg_4\}$ $LNB = \{arg_3, arg_4\}$ $\Lambda = \{arg_3, arg_4\}$ $\Psi' = \{\}$ $\boldsymbol{L(arg_5) = out}$
3	$NB = \{arg_4\}$ $LNB = \{arg_4\}$ $\Lambda = \{arg_4\}$ $\Psi' = \{\}$ $\boldsymbol{L(arg_2) = out}$	$NB = \{arg_5\}$ $LNB = \{arg_5\}$ $\Lambda = \{\}$ $\Psi' = \{\}$ $\boldsymbol{L(arg_3) = in}$	$NB = \{arg_2, arg_5\}$ $LNB = \{arg_2, arg_5\}$ $\Lambda = \{\}$ $\Psi' = \{\}$ $\boldsymbol{L(arg_4) = in}$	$NB = \{arg_3, arg_4\}$ $LNB = \{arg_3, arg_4\}$ $\Lambda = \{arg_3, arg_4\}$ $\Psi' = \{\}$ $\boldsymbol{L(arg_5) = out}$

Figure 2. Labelling of arguments for issue iss_i. The double line represents the main node.

a stretcher that can carry only one person at a time. There are two possible decision alternatives for the robot: recharge its battery, x, or rescue the individual and take him/her to the hospital, y. Let $CKF = \langle\{ag_1, ag_2, ag_3\}, \{0.4, 0.3, 0.3\}, \{x, y\}, med, 10, 0.4\rangle$ and the atoms in the formulae represent the sentences: $a =$ the battery has less than 70 % charge; $b =$ the person is very far from the robot; $c =$ the risk of death is 9 ($[0, 10]$ where 10 means the person is dead); and $d =$ the person is alive. The initial knowledge is:

$$ag_1 = \{\{a, b, \neg c, c \to d, a \wedge b \to x, a \wedge b \to \neg y\}, \{\}\},$$

$$ag_2 = \{\{a, \neg b, c, c \to d, a \wedge b \to x, a \wedge b \to y, a \wedge b \to \neg d\}, \{\}\},$$

$$ag_3 = \{\{b, c, c \to d, d \to \neg x, a \wedge b \to x, a \wedge b \to \neg y, d \to y\}, \{\}\}.$$

Agent med creates the starting point $arg_1 = \langle\{T\}, x\rangle$ and informs the group (arg_1 is stored in the A_i base of all argumentative agents). For voting, we have $Support[x] = \{ag_1\}$ where ag_1 has the argument $\langle\{a, b, a \wedge b \to x\}, x\rangle$ supporting it, and $Reject[x] = \{ag_3\}$ where ag_3 has the argument $\langle\{c, c \to d, d \to \neg x\}, \neg x\rangle$, with $buf(x) = 0.1$. Agent ag_3 has $S_3 = \{\langle\{c, c \to d, d \to \neg x\}, \neg x\rangle\}$ and sends this to med when requested. After being checked by the mediator, this argument is inserted into the dt_x as arg_2. Thus, for voting, we have $Support[c] = \{ag_2\}$, $Reject[c] = \{ag_1\}$, $Support[c \to d] = \{ag_1, ag_2\}$, $Reject[\neg x] = \{ag_1\}$, $buf(c) = 0.2$, $buf(c \to d) = 1$, $buf(d \to \neg x) = 0.3$, and $buf(\neg x) = -0.1$. The group is informed of all formulae and these are updated with the corresponding label.

Agent ag_1 has two arguments arg_3 and arg_4 in $S_1 = \{\langle\{\neg c\}, \neg c\rangle, \langle\{a, b, a \wedge b \to x\}, x\rangle\}$. When requested, ag_1 sends the arguments, med checks them and informs the group. For voting, we have: $Reject[\neg c] = \{ag_2, ag_3\}$ with $buf(\neg c) = -0.2$ (agents ag_2 and ag_3 have the counterargument $\langle\{c\}, c\rangle$ but formula $c[x(0)]$ was presented in a previous argument and was not accepted, meaning that this argument is not admitted); $Support[a] = \{ag_2\}$ and $Reject[a] = \{ag_3\}$ with $buf(a) = 0.4$, $Support[b] = \{ag_3\}$ and $Reject[b] = \{ag_2\}$ with $buf(b) = 0.4$, $Support[a \wedge b \to x] = \{ag_2, ag_3\}$ with $buf(a \wedge b \to x) = 1$ (all agents know this formula), $Reject[x] = \{ag_3\}$

with $buf(x) = 0.1$. Formulae $a[x(0.4)]$, $b[x(0.4)]$, and $a \wedge b \rightarrow x[x(1)]$ are then accepted by the group, becoming common knowledge.

Agent ag_2 has the argument $arg_5 = \langle\{\neg b\}, \neg b\rangle$ in S_2 with $Support[\neg b] = \{ag_3\}$, $Reject[\neg b] = \{ag_1, ag_3\}$ and $buf(\neg b) = -0.1$. Agent ag_3 has the argument $\langle\{b\}, b\rangle$ with $Support[b] = \{ag_1, ag_2\}$, $Reject[b] = \{ag_2\}$ and $buf(b) = 0.7$. Since WB and $AGENDA$ are empty, the dialogue for x is complete. It is important to note that the argumentative agents have other counterarguments during the dialogue, but these are not admitted. Agent *med* starts the dialogue again for the next decision alternative. After all dialogues, the knowledge of the agents is:

$$\begin{aligned} ag_1 = \{\{&a[x(0.4), y(0.4)], b[x(0.4), y(0.4)], \neg c[x(0), y(0.7)], c \rightarrow d[x(1), y(1)],\\ &a \wedge b \rightarrow x[x(1)], a \wedge b \rightarrow \neg y[y(0.4)]\}, \{\}\},\end{aligned}$$

$$\begin{aligned} ag_2 = \{\{&a[x(0.4), y(0.4)], \neg b[x(0), y(0)], c[x(0), y(0)], c \rightarrow d[x(1), y(1)],\\ &a \wedge b \rightarrow x[x(1)], a \wedge b \rightarrow y[y(0)], a \wedge b \rightarrow \neg d[y(0)]\}, \{b[x(0.4), y(0.4)],\\ &a \wedge b \rightarrow \neg y[y(0.4)], \neg c[y(0.7)]\}\},\end{aligned}$$

$$\begin{aligned} ag_3 = \{\{&b[x(0.4), y(0.4)], c[x(0), y(0)], c \rightarrow d[x(1), y(1)], d \rightarrow \neg x[x(0)],\\ &a \wedge b \rightarrow x[x(1)], a \wedge b \rightarrow \neg y[y(0.4)], d \rightarrow y[y(0)]\}, \{a[x(0.4), y(0.4)],\\ &\neg c[y(0.7)]\}\}.\end{aligned}$$

Table 1 shows the dialogues for x and y. The corresponding argumentation graphs are shown in Figure 3 with the arguments and their overall strengths. The steps used in labelling the arguments are presented in Table 2. For a dialogue about x, we have: $L(arg_2) = out$, $L(arg_3) = in$, $L(arg_4) = in$, $L(arg_5) = out$, $L(arg_6) = in$, $ACC = \{arg_3, arg_4, arg_6\}$, $POS(arg_3) = 1$, $POS(arg_4 = 1)$, $POS(arg_6) = 1$ and $pref(x) = 1.52$. For a dialogue about y, we have: $L(arg_2) = in$, $L(arg_3) = out$, $L(arg_4) = out$, $L(arg_5) = out$, $L(arg_6) = in$, $L(arg_7) = in$, $L(arg_8) = in$, $ACC = \{arg_2, arg_6, arg_7, arg_8\}$, $POS(arg_2) = -1$, $POS(arg_6) = -1$, $POS(arg_7) = -1$, $POS(arg_8) = -1$, and $pref(y) = -1.83$. As a result of this dialogue, we have $x \succ y$, where x is the preferred decision alternative for the group.

4.1 Results and Discussion

From the practical example given above, it is possible to observe the relation between common knowledge and consensus about information that is accepted by the group of agents. In this work, we refer to each decision alternative as a possible world. One approach to formalising these possible worlds is the Kripke structure [12]. A Kripke Structure KS for n agents over a set of primitive propositions Γ is a tuple $(P, \pi, \Theta_1, \ldots, \Theta_n)$ where P is a non-empty set of possible worlds; $\pi : (p) \rightarrow \{true, false\}$ is a function that assigns a truth value to the propositions in Γ for each possible world $p \in P$; and Θ_i is a binary accessibility relation between the possible worlds in P. We can represent the practical example in a Kripke Structure using the system S5 [12, 14, 18] for knowledge representation in each agent

γ	dt_x							dt_y						
	ag	**arg**	**att**	**Support**	**Reject**	**buf**	**is**	**ag**	**arg**	**att**	**Support**	**Reject**	**buf**	**is**
1	med	$\langle\{T\}, x\rangle$		$x[ag_1]$	$x[ag_3]$	0.1	0.55	med	$\langle\{T\}, y\rangle$		$y[ag_2, ag_3]$	$y[ag_1, ag_3]$	-0.1	0.45
2	ag_3	$\langle\left\{\begin{matrix} c, \\ c \to d, \\ d \to \neg x \end{matrix}\right\}, \neg x\rangle$	1	$c[ag_2]$ $c \to$ $d[ag_1, ag_2]$ $d \to \neg x[]$ $\neg x[]$	$c[ag_1]$ $c \to d[]$ $d \to \neg x[]$ $\neg x[ag_1]$	0.2 1.0 0.3 -0.1	0.66	ag_1	$\langle\left\{\begin{matrix} a, \\ b, \\ a \wedge b \to !\, y \end{matrix}\right\}, \neg y\rangle$	1	$a[ag_2, ag_3]$ $b[ag_2, ag_3]$ $a \wedge b \to$ $\neg y[ag_3]$ $\neg y[ag_3]$	$a[ag_2, ag_3]$ $b[ag_2, ag_3]$ $a \wedge b \to$ $\neg y[ag_2]$ $\neg y[ag_2, ag_3]$	0.4 0.4 0.4 0.1	0.66
3	ag_1	$\langle\{\neg c\}, \neg c\rangle$	2	$\neg c[]$	$\neg c[ag_2, ag_3]$	-0.2	0.40	ag_3	$\langle\left\{\begin{matrix} b, \\ c, \\ c \to d, \\ d \to y, \\ a \wedge b \to \neg y \end{matrix}\right\}, \neg a\rangle$	2	$b[ag_1, ag_2]$ $c[ag_2]$ $c \to$ $d[ag_1, ag_2]$ $d \to y[]$ $a \wedge b$ $\to \neg y[ag_1, ag_2]$ $\neg a[ag_2]$	$b[ag_2]$ $c[ag_1, ag_2]$ $c \to d[]$ $d \to y[]$ $a \wedge b$ $\to \neg y[ag_2]$ $\neg a[ag_1, ag_2]$	0.7 -0.1 1.0 0.3 0.7 -0.1	0.71
4	ag_1	$\langle\left\{\begin{matrix} a, \\ b, \\ a \wedge b \to x \end{matrix}\right\}, x\rangle$	2	$a[ag_2]$ $b[ag_3]$ $a \wedge b \to$ $x\ [ag_2, ag_3]$ $x[]$	$a[ag_3]$ $b[ag_2]$ $a \wedge b \to x\ []$ $x[ag_3]$	0.4 0.4 1.0 0.1	0.74	ag_2	$\langle\left\{\begin{matrix} b, \\ a \wedge b \to y, \\ a \wedge b \to \neg y \end{matrix}\right\}, \neg a\rangle$	2	$b[ag_1, ag_3]$ $a \wedge b \to y[]$ $a \wedge b \to$ $\neg y[ag_1, ag_3]$ $\neg a[ag_3]$	$b[ag_3]$ $a \wedge b \to$ $y[ag_1, ag_3]$ $a \wedge b \to \neg y[]$ $\neg a[ag_1, ag_3]$	0.7 -0.2 1.0 -0.1	0.68
5	ag_2	$\langle\{\neg b\}, \neg b\rangle$	4	$\neg b[ag_3]$	$\neg b[ag_1, ag_3]$	-0.1	0.45	ag_2	$\langle\{\neg b\}, \neg b\rangle$	2	$\neg b[ag_3]$	$\neg b[ag_1, ag_3]$	-0.1	0.45
6	ag_3	$\langle\{b\}, b\rangle$	5	$b[ag_1, ag_2]$	$b[ag_2]$	0.7	0.85	ag_2	$\langle\left\{\begin{matrix} a, \\ b, \\ a \wedge b \to \neg d, \\ c \to d \end{matrix}\right\}, \neg c\rangle$	3	$a[ag_1, ag_3]$ $b[ag_1, ag_3]$ $a \wedge b \to \neg d[]$ $c \to$ $d[ag_1, ag_3]$ $\neg c[ag_1, ag_3]$	$a[ag_3]$ $b[ag_3]$ $a \wedge b \to$ $\neg d[]$ $c \to d[]$ $\neg c[ag_3]$	0.7 0.7 0.3 1.0 0.7	0.84
7								ag_2	$\langle\{a\}, a\rangle$	3	$a[ag_1, ag_3]$	$a[ag_3]$	0.7	0.85
8								ag_1	$\langle\{b\}, b\rangle$	5	$a[ag_2, ag_3]$	$a[ag_2, ag_3]$	0.4	0.70

Table 1. Arguments, supporting and rejecting votes and intrinsic strengths for x and y

	$time_w$	arg_2	arg_3	arg_4	arg_5	arg_6		
dt_x	1	$L(arg_2) = in$	$L(arg_3) = in$	$L(arg_4) = in$	$L(arg_5) = out$	$L(arg_5) = in$		
	2	$\boldsymbol{L(arg_2) = out}$	$\boldsymbol{L(arg_3) = in}$	$\boldsymbol{L(arg_4) = in}$	$\boldsymbol{L(arg_5) = out}$	$\boldsymbol{L(arg_5) = in}$		
	3	$\boldsymbol{L(arg_2) = out}$	$\boldsymbol{L(arg_3) = in}$	$\boldsymbol{L(arg_4) = in}$	$\boldsymbol{L(arg_5) = out}$	$\boldsymbol{L(arg_5) = in}$		
	$time_w$	arg_2	arg_3	arg_4	arg_5	arg_6	arg_7	arg_8
dt_y	1	$L(arg_2) = in$	$L(arg_3) = out$	$L(arg_4) = in$	$L(arg_5) = out$	$L(arg_5) = in$	$L(arg_5) = in$	$L(arg_5) = in$
	2	$L(arg_2) = out$	$L(arg_3) = out$	$L(arg_4) = out$	$L(arg_5) = out$	$L(arg_5) = in$	$L(arg_5) = in$	$L(arg_5) = in$
	3	$\boldsymbol{L(arg_2) = in}$	$\boldsymbol{L(arg_3) = out}$	$\boldsymbol{L(arg_4) = out}$	$\boldsymbol{L(arg_5) = out}$	$\boldsymbol{L(arg_5) = in}$	$\boldsymbol{L(arg_5) = in}$	$\boldsymbol{L(arg_5) = in}$
	4	$\boldsymbol{L(arg_2) = in}$	$\boldsymbol{L(arg_3) = out}$	$\boldsymbol{L(arg_4) = out}$	$\boldsymbol{L(arg_5) = out}$	$\boldsymbol{L(arg_5) = in}$	$\boldsymbol{L(arg_5) = in}$	$\boldsymbol{L(arg_5) = in}$

Table 2. Labelling arguments from dt_x and dt_y

where the possible worlds are symmetric, transitive and reflexive. Let the sequence $[x, y]$ represent the decision alternatives, with values T for true and F for false.

Before the dialogue, agent ag_1 has one argument asserting x ($\langle\{a, b, a \wedge b \to x\}, x\rangle$) and one argument asserting $\neg y$ ($\langle\{a, b, a \wedge b \to \neg y\}, \neg y\rangle$). For this agent, there is only one possible world: $w_1 = [T, F]$. Agent ag_2 does not have an argument for either x or y. In this case, both decision alternatives are accepted with four possible worlds: $w_1 = [T, F]$, $w_2 = [F, T]$, $w_3 = [T, T]$, $w_4 = [F, F]$. Agent ag_3 has one argument for $\neg x$ ($\langle\{c, c \to d, d \to \neg x\}, \neg x\rangle$) and one for y ($\langle\{c, c \to d, d \to$

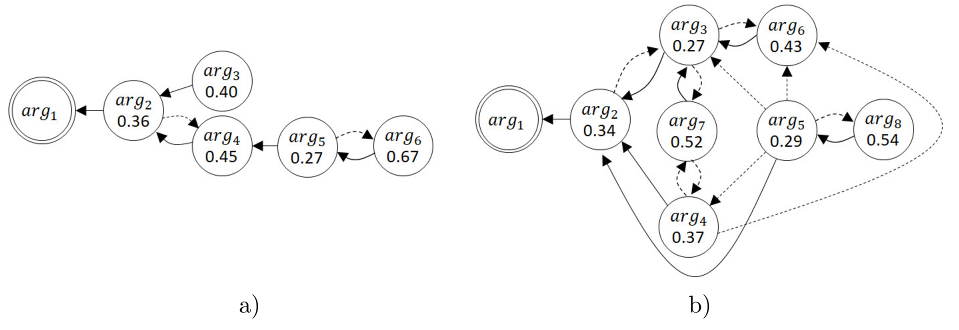


Figure 3. Argumentation graphs mapped from a) dt_x and b) dt_y. Double lines represent the main node, and dotted lines are additional attack relations detected in the mapping.

$y\}, y\rangle)$ with only one possible world: $w_2[F,T]$. In this case, there is no consensus on x or y, as shown in Figure 4 a).

After the dialogue, $ag1$, ag_2 and ag_3 accept only world $w_1 = [T,F]$ (arguments: $\langle\{a, b, a \wedge b \rightarrow x\}, x\rangle$ and $\langle\{a, b, a \wedge b \rightarrow \neg y\}, \neg y\rangle$). Agent ag_2 still has an argument for y ($\langle\{a, b, a \wedge b \rightarrow y\}, y\rangle$) and ag_3 has arguments for $\neg x$ and y ($\langle\{c, c \rightarrow d, d \rightarrow \neg x\}, \neg x\rangle$ and $\langle\{c, c \rightarrow d, d \rightarrow y\}, y\rangle$), but these arguments are not admitted and therefore do not establish a position against $\neg x$ or in favour of y. After the dialogue, we can observe that there is a consensus on world $w_1 = [T,F]$, as shown in Figure 4 b).

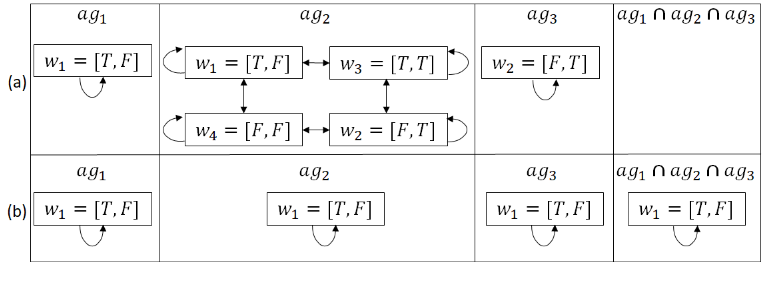


Figure 4. Possible worlds and consensus for agents ag_1, ag_2, and ag_3: a) before the dialogue; and b) after the dialogue, with common knowledge formation

We can also use the modal operators K, E, and C to represent the knowledge of the agents:

- $K_i\varphi$ is equivalent to $\varphi \in \Sigma_i$ (e.g. $K_1\neg c$ implies that $\neg c \in \Sigma_1$). If an agent knows a piece of information, that information is in the agent's knowledge base or can be inferred.
- $\neg K_i K_j \varphi$ (e.g. $K_2 d$ and $\neg K_1 d$ or $\neg K_1 K_2 d$). If an agent does not know about a piece of information, it can query other agents.

- $K_i\varphi$ implies that $K_iK_i\varphi$. If an information is in the agent's knowledge base, the agent knows that information and can use it to build new arguments and vote.
- $\neg K_i\varphi$ implies that $K_i\neg K_i\varphi$. The unknown information cannot be used to build new arguments and vote.
- $E_{AG}\varphi$ is equivalent to $\forall ag_i \in AG : \varphi \in \Sigma_i$. Before the dialogue in the practical example, agents know $c \to d$.
- $C_{AG}\varphi$ is equivalent to $\forall \Sigma_i, \exists \varphi : \varphi[iss(b)]$ for $iss \in ISS$ and $b > 0$. A formula is common knowledge when related to an issue if it has a label with a consensus level greater than zero.

Other characteristics of the model are as follows:

- It is able to represent the different belief states for each formula. Formulae φ and $\neg\varphi$ may be accepted at the same time for the same issue. The agent may not have a well-defined position for that information.
- There are two ways in which an agent can decrease the strength of an argument: voting for rejection of its formulae or sending counterarguments.
- Maximal acceptance of an argument arg results in $is(arg) = 1$, while maximal rejection of the argument results in $is(arg) = ex_i$ (the expertise value of the proponent).
- A small number of strong attacks may be equivalent to or more rigorous than several weak attacks.

4.2 Related Works

Dung [11] proposed some semantics to determine the admissibility of the arguments, that is, a formal framework to identify conflict outcomes, such as preferred or grounded semantics. The idea is to specify sets of acceptable arguments or extensions. An extension is a set of arguments that can be accepted together. These semantics are used to select arguments without considering support for or rejection of a decision alternative or group decision.

Coste-Marquis et al. [9] used an argumentation graph with weights in the attack relations, and applied Dung's semantics in determining the last attacked or best defended extensions. Our proposal deals with strengths in arguments represented as numerical values, applied when a group of agents intends to select the preferred alternative by considering the opinions of all the agents.

Da Costa Pereira et al. [10] use a belief revision based on argumentation that assigns fuzzy labelling to each argument, permitting the agent to change its mind without removing the previous information forever, and allowing for recovery if this information turns out to be wrong. In our work, the evaluation is carried out based on the set of formulae of the argument, all arguments are evaluated, and the agents store all the information that is acceptable to every possible world in their knowledge bases.

The work closest to our approach is probably that of Leite and Martins [19], who extend Dung's framework by applying it to online debate systems, allowing people to vote to support or reject arguments or to send arguments that are not logically structured. They defined a semantics for application to online debating systems (democracy, universality, etc.) and to rank the arguments from the strongest to the weakest, suggesting a preference for the group, although not a definite one. In our work, arguments are logically structured; they are sent by agents within a restricted group; the strengths of the arguments use quantitative (votes) as well as qualitative (attacks) values; there are different dialogues, one for each issue; and the framework still creates the preference order of the possible decision alternatives as a result.

5 CONCLUSION

This work presents an argumentative dialogue model for CKF in a group of agents. This model is generic, and can be applied to a discussion about any issue where there is a need for group opinions. We use propositional logic to represent the information in the knowledge bases, and to build arguments and a voting model for support and rejection, although other logical languages may be used.

The model has four main characteristics:

1. it allows agents to take part in a dialogue by exchanging arguments through attack relations, while supporting or rejecting the arguments by voting on their formulae;
2. based on the expertise of the agents, we can evaluate the arguments in numerical form, representing the extent to which each formula (or argument as a whole) is accepted by the group of agents;
3. the results obtained after the dialogue allow for an approximation of opinions, meaning that the group can apply the model in consensus decision-making problems; and
4. the model presents a direct relation between common knowledge and consensus.

When a piece of information is taken as common knowledge, there is a consensus of the group accepting that information. The output of the model is not the optimal decision, but rather the decision preferred by the group.

There are several possible ways to extend this work. Some of these future directions involve the application of the model to decision making in which blocking is possible, and the use of a reputation system to assign expertise to the agents, where each value is related to the type of information presented in the argument.

REFERENCES

[1] Amgoud, L.—Belabbès, S.—Prade, H.: A Formal General Setting for Dialogue Protocols. In: Euzenat, J., Domingue, J. (Eds.): Artificial Intelligence: Methodology,

Systems, and Applications (AIMSA 2016). International Workshop on Argumentation in Multi-Agent Systems, Varna, Bulgaria, September 2006. Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 4183, 2006, pp. 13–23, doi: 10.1007/11861461_4.

[2] Amgoud, L.—Ben-Naim, J.—Doder, D.—Vesic, S.: Acceptability Semantics for Weighted Argumentation Frameworks. Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, Vol. 1, 2017, pp. 56–62, doi: 10.24963/ijcai.2017/9.

[3] Amgoud, L.—Prade, H.: Using Arguments for Making and Explaining Decisions. Journal of Artificial Intelligence, Vol. 173, 2009, No. 3, pp. 413–436, doi: 10.1016/j.artint.2008.11.006.

[4] Besnard, P.—Hunter, A.: Constructing Argument Graphs with Deductive Arguments: A Tutorial. Argument and Computation, Vol. 5, 2014, No. 1, pp. 5–30, doi: 10.1080/19462166.2013.869765.

[5] Bistarelli, S.—Pirolandi, D.—Santini, F.: Solving Weighted Argumentation Frameworks with Soft Constraints. In: Larrosa, J., O'Sullivan, B. (Eds.): Recent Advances in Constraints (CSCLP 2009). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 6384, 2011, pp. 1–18, doi: 10.1007/978-3-642-19486-3_1.

[6] Bodanza, G.—Tohmé, F.—Auday, M.: Collective Argumentation: A Survey of Aggregation Issues Around Argumentation Frameworks. Argument and Computation, Vol. 8, 2017, No. 1, pp. 1–34, doi: 10.3233/aac-160014.

[7] Caminada, M.: On the Issue of Reinstatement in Argumentation. In: Fisher, M., van der Hoek, W., Konev, B., Lisitsa, A. (Eds.): Logics in Artificial Intelligence (JELIA 2006). Springer, Berlin, Heidelberg, Lecture Notes in Computer Science, Vol. 4160, 2006, pp. 111–123, doi: 10.1007/11853886_11.

[8] Cayrol, C.—Lagasquie-Schiex, M. C.: Graduality in Argumentation. Journal of Artificial Intelligence Research, Vol. 23, 2005, pp. 245–297, doi: 10.1613/jair.1411.

[9] Coste-Marquis, S.—Konieczny, S.—Marquis, P.—Ouali, M. A.: Selecting Extensions in Weighted Argumentation Frameworks. In: Verheij, B., Szeider, S., Woltran, S. (Eds.): Computational Models of Argument. IOS Press, Frontiers in Artificial Intelligence and Applications, Vol. 245, 2012, pp. 342–349, doi: 10.3233/978-1-61499-111-3-342.

[10] Da Costa Pereira, C.—Tettamanzi, A. G. B.—Villata, S.: Changing One's Mind: Erase or Rewind? Possibilistic Belief Revision with Fuzzy Argumentation Based on Trust. Proceedings of the 22$^{\text{nd}}$ International Joint Conference on Artificial Intelligence, Barcelona, Catalonia, Spain, 2011, pp. 164–171.

[11] Dung, P. M.: On the Acceptability of Arguments and Its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. Artificial Intelligence, Vol. 77, 1995, No. 2, pp. 321–357, doi: 10.1016/0004-3702(94)00041-x.

[12] Fagin, R.—Halpern, J. Y.—Moses, Y.—Vardi, M. Y.: Reasoning about Knowledge. MIT Press, London, England, 2004.

[13] Fagin, R.—Halpern, J. Y.: Reasoning about Knowledge and Probability. Journal of the ACM, Vol. 41, 1994, No. 2, pp. 340–367, doi: 10.1145/174652.174658.

[14] Girle, R.: Possible Worlds. Acumen Publishing, Durham, 2013, doi: 10.1017/upo9781844653454.

[15] Grossi, D.—Pigozzi, G.: Judgment Aggregation: A Primer. Synthesis Lectures on Artificial Intelligence and Machine Learning, Morgan and Claypool, Vol. 8, 2014, No. 2, pp. 1–151, doi: 10.2200/s00559ed1v01y201312aim027.

[16] Halpern, J. Y.—Moses, Y.: Knowledge and Common Knowledge in a Distributed Environment. Journal of the ACM, Vol. 37, 1990, No. 3, pp. 549–587, doi: 10.1145/79147.79161.

[17] Halpern, J. Y.: Reasoning about Knowledge: An Overview. Proceedings of the 1986 Conference on Theoretical Aspects of Reasoning about Knowledge, Monterey, California, 1986, pp. 1–17, doi: 10.1016/b978-0-934613-04-0.50004-1.

[18] van der Hoek, W.: Systems for Knowledge and Belief. Journal of Logic and Computation, Vol. 3, 1993, No. 2, pp. 173–195, doi: 10.1093/logcom/3.2.173.

[19] Leite, J.—Martins, J.: Social Abstract Argumentation. Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence (IJCAI '11), Barcelona, Spain, Vol. 3, 2011, pp. 2287–2292, doi: 10.5591/978-1-57735-516-8/IJCAI11-381.

[20] Ménager, L.: Consensus and Common Knowledge of an Aggregate of Decisions. Games and Economic Behaviour, Vol. 62, 2008, No. 2, pp. 722–731, doi: 10.1016/j.geb.2007.04.007.

[21] Parsons, S.—McBurney, P.: Argumentation-Based Dialogues for Agent Co-Ordination. Group Decision and Negotiation, Vol. 12, 2003, No. 5, pp. 415–439, doi: 10.1023/b:grup.0000003742.50038.d3.

[22] Rahwan, I.—Simari, G. R. (Eds.): Argumentation in Artificial Intelligence. Springer US, 2009, doi: 10.1007/978-0-387-98197-0.

[23] Rahwan, I.—Tohmé, F.: Collective Argument Evaluation as Judgement Aggregation. Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems (AAMAS '10), Toronto, Canada, Vol. 1, 2010, pp. 417–424.

[24] Rome, A. G. L.: Unanimity Rule and Organizational Decision Making: A Simulation Model. Organization Science, Vol. 15, 2004, No. 6, pp. 704–718, doi: 10.1287/orsc.1040.0090.

[25] Singh, R.—Benyoucef, L.: A Consensus Based Group Decision Making Methodology for Strategic Selection Problems of Supply Chain Coordination. Engineering Applications of Artificial Intelligence, Vol 26, 2013, No. 1, pp. 122–134, doi: 10.1016/j.engappai.2012.03.013.

Ayslan Trevizan Possebom received his B.Sc. in information systems in 2002 from Paranaense University and his M.Sc. in computer science from the State University of Maringa. He is Full Professor of Information Technology at the Federal Institute of Paraná, and a Ph.D. candidate in the electrical and computer engineering program at the Federal University of Technology of Parana. His research interests include multi-agent systems, argumentation and decision making.

Mariela Morveli-Espinoza received her B.Sc. in systems engineering in 2001 from the San Agustin National University of Arequipa and her M.Sc. in systems engineering and computer science from the Federal University of Rio de Janeiro. She is currently a Ph.D. candidate in the electrical and computer engineering program at the Federal University of Technology of Parana. Her research interests include intelligent agents, multi-agent systems and argumentation.

Cesar Augusto Tacla received his B.Sc. in software engineering in 1989 and his M.Sc. in electrical and computer engineering in 1993 from the Federal University of Technology of Parana. In 2003, he was awarded his Ph.D. degree in informatics from the University of Technology of Compiègne. Currently, he is Full Professor within the electrical and computer engineering program at the Federal University of Technology of Parana. His research interests include multi-agent systems, argumentation, ontologies and cooperative systems.