Noname manuscript No.
(will be inserted by the editor)

A distributed argumentation algorithm for mining consistent opinions in weighted Twitter discussions

Teresa Alsinet · Josep Argelich · Ramón Béjar · Joel Cemeli

Received: / Accepted:

Abstract Twitter is one of the most powerful social media platforms, reflecting both support and contrary opinions among people who use it. In a recent work we developed an argumentative approach for analyzing the major opinions accepted and rejected in Twitter discussions. A Twitter discussion is modeled as a weighted argumentation graph where each node denotes a tweet, each edge denotes a relationship between a pair of tweets of the discussion and each node is attached to a weight that denotes the social relevance of the corresponding tweet in the discussion. In the social network Twitter, a tweet always refers to previous tweets in the discussion, and therefore the underlying argument graph obtained is acyclic. However, when in a discussion we group the tweets by author, the graph that we obtain can contain cycles. Based on the structure of graphs, in this work we introduce a distributed algorithm to compute the set of globally accepted opinions of a Twitter discussion based on valued argumentation. To understand the usefulness of our distributed algorithm, we study cases of argumentation graphs that can be solved efficiently with it. Finally, we present an experimental investigation that shows that when solving acyclic argumentation graphs associated with Twitter discussions our algorithm scales at most with linear time with respect to the size of the discussion. For argumentation graphs with cycles, we study tractable cases and we analyze how frequent are these cases in Twitter. Moreover, for the non-tractable cases we analyze how close is the solution of the distributed algorithm with respect to the one computed with the general sequential algorithm, that we have previously developed, that solves any argumentation graph.

Keywords Twitter discussions · Valued argumentation · Probability values · Distributed algorithm · Tractable cases

1 Motivation and related work

Twitter is one of the most powerful social media platforms, given the number of highly focused and influential networks of people who use it. One of the biggest reasons for wide adoption of Twitter has been its swiftness in dissemination of information, apart from its transparency on not restricting content. The openness of the platform makes it easy to monitor, listen and communicate.

In order to ascertain the major opinions accepted and rejected by Twitter users in different domains, in a recent work [2] we have developed a system to analyze discussions in Twitter. The system architecture has two main components: a discussion retrieval and a reasoning system. The discussion retrieval component allows us to move from a discussion in Twitter (a set of tweets) in natural language to a weighted graph which is computed taking into account relationships between tweets and three different attributes of a tweet: the number of followers of the author, the number of retweets and the number of favorites. The reasoning system component maps the weighted graph into a weighted argumentation framework and the set of socially accepted tweets in the discussion is evaluated from the weight assigned to

T. Alsinet
INSPIRES Research Center, University of Lleida, Spain
Tel.: +34-973-702734
E-mail: tracy@diei.udl.cat

J. Argelich
INSPIRES Research Center, University of Lleida, Spain
Tel.: +34-973-003477
E-mail: jargelich@diei.udl.cat

R. Béjar
INSPIRES Research Center, University of Lleida, Spain
Tel.: +34-973-702736
E-mail: ramon@diei.udl.cat

J. Cemeli
Starloop Studios, Lleida, Spain
E-mail: joel.cemeli@starloopstudios.com

each tweet and the relationships between the tweets of the discussion, and it is computed as the ideal extension (output) [16] of a valued abstract argumentation framework [8]. The algorithm to compute the ideal semantics works by computing all the arguments that belong to some admissible extension and it is based on the approach presented in [18], but adapting it to work with valued arguments. Regarding the reasoning system implementation, we developed an approach based on Answer Set Programming (ASP) available in the argumentation system ASPARTIX [23], but we extended it to work with valued arguments, as the current implementation in ASPARTIX only works with non-valued arguments. A detailed description of our manifold ASP computation algorithm can be found in [2].

Argumentation-based reasoning models aim to reflect how human argumentation uses conflicting information to construct and analyze arguments. An argument is an entity that represents some grounds to believe in a certain statement and that can be in conflict with arguments establishing contradictory claims. The most commonly used framework to talk about general issues of argumentation is that of abstract argumentation [15]. In abstract argumentation, an argumentation graph is used to represent a set of arguments and counterarguments. Each node is an argument and each edge denotes an attack between arguments. Several different kinds of semantics for abstract argumentation frameworks have been proposed that highlight different aspects of argumentation (for reviews see [9, 10, 31]). Usually, semantics is given to abstract argumentation frameworks in terms of extensions. For a specific extension the status of an argument is either accepted or rejected and, usually, there is a set of extensions that is consistent with the semantic context.

The exploitation of Twitter by means of argumentation frameworks has also been explored by Grosse et al. [25, 26] with the aim of detecting conflicting elements in an opinion tree to avoid potentially inconsistent information. In a different context, in order to mine arguments from Twitter, Bosc et al. [12] proposed a binary classification mechanism (argument-tweet vs. non argument) and Dusmanu et al. [21] applied supervised classification to identify arguments on Twitter and evaluated facts recognition and source identification for argument mining.

When constructing relationships between tweets from informal descriptions expressed in natural language with other attributes such as emoticons, jargon, onomatopoeia and abbreviations, it is often evident that there is uncertainty about whether some of the criticism relationships hold. In order to deal with such uncertainty, in [3] we have introduced and investigated a natural extension of our previous system [2], in which criticism relationships between tweets are associated with a probability value, indicating the uncertainty that the relationships hold. An important element of the system extended with probability values is the notion of an uncertainty threshold, which characterizes how much uncertainty of probability values we are prepared to tolerate: given an uncertainty threshold α, we would be prepared to disregard the criticism relationships up to α. We therefore obtain a valued abstract argumentation framework where arguments are tweets, argument values are the weights used to model the relative social relevance of tweets from data obtained from Twitter, and attacks between arguments denote criticism relationships between tweets whose probability of fulfillment is greater than or equal to α.

There have been many developments centered on the extension of argumentation frameworks for reasoning with probabilistic information. [29] extend Dung's framework to form a probabilistic argument framework by associating probabilities with arguments and defeats. These probabilities represent the likelihood of existence of a specific argument or defeat. Later, [27] investigates the foundations of probabilistic argument graphs, [32] defines a probabilistic semantics for pure abstract argumentation frameworks and [24] address the fundamental problem of computing the probability that a set of arguments is an extension according to a given semantics. In [28], the author assigns probability values to attacks between arguments and uses them to obtain a probability distribution over the set of spanning subgraphs of an argument graph as a sample space. The probability distribution over the set of spanning subgraphs is used to determine the probability that a set of arguments is admissible or an extension. In the system that we proposed in [3], the probabilities of edges of an argument graph are used for pruning purposes and we use valued abstract argumentation semantics to solve the resulting graph where arguments are tweets, argument values are the weights used to model the relative social relevance of tweets from data obtained from Twitter, and attacks between arguments express criticism relationships between tweets whose probability of fulfillment is greater than or equal to an uncertainty threshold.

In this work we define a distributed approach that allows us to efficiently analyze Twitter discussions with large numbers of tweets and probability valued relationships between them. Since Twitter is a widely used social media platform and the discussions around a particular topic can be quite big, it seems reasonable to consider a distributed approach to compute the status of an argument under a particular semantics with a good scaling performance. As far as we know, the only previous work about solving argumentation frameworks with a distributed approach is the one proposed by Baroni and Giacomin in [5, 6]. The authors proposed a distributed algorithm in which several independent asynchronous processes carry out argumentation activity by exploiting local information only. Their approach devises a general distributed algorithm, which does not rely on any specific notion of defeat between arguments, and a coordination process was explicitly considered by ensuring the property of self-stabilization for the algorithm.

The distributed approach we present here computes the set of accepted tweets (solution) of a Twitter discussion by means of the the ideal extension (output) of a valued abstract argumentation framework. In our framework, one tweet defeats another tweet if the first one criticizes the second one with a sufficient degree of probability and it has at least the same social relevance as the second one. Solving a Twitter discussion by means of the computation of the ideal extension of a valued abstract argumentation framework allows us to unsure both that the set of tweets in the solution is the maximal set of tweets that satisfies that it is consistent, in the sense that there are no defeaters among them, and that the tweets within the solution defeat all the attacks that come from tweets outside of the solution.

In our approach the social relevance of tweets is measured with a weighting scheme that for each tweet considers different information sources from the social network, such as the number of followers of the author, the number of retweets and the number of favorites. The distributed approach can be of special relevance to assess Twitter discussions that involve a large number of tweets and the system can be applied in fields where it is of particular interest to identify groups of tweets that are globally compatible or consistent, but at the same time widely accepted, such as for instance for the assistance and guidance of marketing and policy makers.

In the social network Twitter, a tweet always answers or refers to previous tweets in the discussion, and the underlying argument graph obtained is therefore acyclic. However, when in a discussion we group the tweets by author, the graph that we obtain can contain cycles. Aggregating the information by author allows us to mind the set of authors whose opinions are consistent or in agreement in the discussion, the authors involved in a circular argumentative discussion and the most controversial authors. That is, for instance, the authors who receive the greatest number of effective attacks, the authors who participate in the greatest number of cycles or the authors that generate the longest argumentative chains. In this work we also study how we can use the distributed approach to analyze Twitter discussions from the point of view of authors' opinions (sets of tweets) and not only from the perspective of tweet.

Therefore, our approach takes into account the structure of the argumentation graphs of the Twitter discussions to compute the set of globally accepted opinions based on valued argumentation. Moreover, to understand the usefulness of this new distributed approach, we study cases of argumentation graphs that can be solved efficiently with it. As far as we know, this is the first distributed system developed to solve such a problem.

As a final contribution, we present an experimental investigation that shows that when solving acyclic argumentation graphs associated with Twitter discussions our algorithm scales at most with linear time with respect to the size of the discussion. Moreover, for argumentation graphs with cycles, we study tractable cases and we analyze how frequent are these cases in Twitter. In addition, for the non-tractable cases, we analyze how close is the solution of the distributed algorithm with respect to the one computed with the general sequential algorithm, that we have previously developed in [2], that solves any argumentation graph.

After this introduction, in the next section, we formalize the tweet-based representation model for Twitter discussions and, in Section 3, we define the argumentation reasoning model to compute their solutions. Then, in Section 4, we present our novel distributed algorithm for the implementation of the reasoning model based on the ideal semantics of a valued abstract argumentation framework. In Section 5, we define an author-centered representation model which allows us to perform a different analysis from the one that results from the tweet-based approach and, in Section 6, we show that the distributed algorithm also allows us to solve some of the argumentation graphs for the author-centered representation model. In Section 7 we show that our distributed algorithm can also be used for the case of argumentation graphs without even cycles and, in Section 8, we analyze the frequency of the special cases that can be solved by our algorithm. We end the paper with some conclusions and a discussion of future work.

2 Tweet-based model for a Twitter discussion

In this section, we introduce a simplified computational structure starting from the one proposed in [3] to represent a Twitter discussion with probabilistic valued relationships, which will be called *probabilistic discussion graph*. In such a graph, each node will denote a tweet, each edge will denote an answer relationship between a pair of tweets of the discussion, and each edge will be associated with a probability value, indicating the probability that a criticism relationship between the pair of tweets holds. We provide more formal definitions below.

Definition 1 A *Twitter discussion* Γ is a non-empty set of tweets. A *tweet* $t \in \Gamma$ is a tuple $t = (m, a, fl, r, fv)$, where m is the up to 140 characters long message of the tweet, a is the author's identifier of the tweet, $fl \in \mathbb{N}$ is the number of followers of the author, $r \in \mathbb{N}$ is the number of retweets and $fv \in \mathbb{N}$ is the number of favorites.

Let $t_1 = (m_1, a_1, fl_1, r_1, fv_1)$ and $t_2 = (m_2, a_2, fl_2, r_2, fv_2)$ be tweets from different authors; i.e. $a_1 \neq a_2$. We say that t_1 *answers* t_2 iff t_1 is a reply to tweet t_2 or t_1 mentions (refers to) tweet t_2.

Definition 2 The *Discussion Graph* (DisG) for a Twitter discussion Γ is the directed graph (T, E) such that:

- for every tweet in Γ there is a node in T and
- if t_1 and t_2 are tweets from different authors such that t_1 answers t_2 there is a directed edge (t_1, t_2) in E.

Only the nodes and edges obtained by applying this process belong to T and E, respectively.

Definition 3 A *Weighted Discussion Graph* (WDisG) for a Twitter discussion Γ is a tuple $\langle T, E, R, W\rangle$, where

- (T, E) is the DisG graph for Γ,
- R is the non-empty set of ordered values that models the social relevance values of tweets and
- W is a weighting scheme $W : \mathbb{N}^3 \to R$ for the weight of nodes (tweets). The weighting scheme W evaluates the social relevance of tweets by mapping the triple of values $(\mathit{fl}, r, \mathit{fv}) \in \mathbb{N}^3$ of a tweet $(m, a, \mathit{fl}, r, \mathit{fv}) \in \Gamma$ to a value in R by combining the three sources of information: number of followers, number of retweets and number of favorites.

Regarding the implementation, in order to weight nodes of a WDisG graph considering the social relevance of tweets, we instantiate the set of ordered values R to the natural numbers $\mathbb{N}$ and we consider two weighting schemes $W : \mathbb{N}^3 \to \mathbb{N}$:

(1) $W(\mathit{fl}, r, \mathit{fv}) = \lfloor \log_{10}(\mathit{fl} + 1)\rfloor$ which only considers the number of followers and allows us to quantify the tweets' social relevance from the orders of magnitude of authors' followers, since we want to consider that one tweet is more relevant than another only if the number of followers is at least ten times bigger for the first tweet. We will refer to this weighting scheme as `followers` scheme.

(2) $W(\mathit{fl}, r, \mathit{fv}) = \lfloor \log_{10}(\mathit{fl} + 20 * r + 40 * \mathit{fv} + 1)\rfloor$ which considers not only the number of followers but also the number of retweets and favorites. This function allows us to quantify the orders of magnitude of the social relevance of tweets following the statistics about tweets and retweets defined in [11], trying to give each attribute a weight proportional to its relevance. From the statistics shown in [11], we observe that on weighting with twenty times the value of retweets and forty times the value of favorites, the magnitudes of the three attributes are comparable and one attribute does not dominate the others, since the number of followers is usually much bigger than the number of retweets and favorites. Finally, as for the `followers` scheme, we compute the $\log_{10}$ function of the combined value. We will refer to this weighting scheme as `fl1r20fv40` scheme.

Definition 4 A *Probabilistic Discussion Graph* (PDisG) for a Twitter discussion Γ is a tuple $\langle T, E, R, W, P\rangle$, where

- $\langle T, E, R, W\rangle$ is the WDisG graph for Γ and
- P is a labeling function $P : E \to [0, 1]$ for edges in E. The labeling function P maps an edge (t_1, t_2) to a probability value $p \in [0, 1]$, which expresses the degree of belief that the message of tweet t_1 is a criticism of the message of tweet t_2. Criticism means that the message of tweet t_1 does not agree with the claim expressed in the message of tweet t_2. Thus, $p = 1$ means that we fully believe that tweet t_1 disagrees with the claim expressed in tweet t_2, while $p = 0$ means that we believe that tweet t_1 agrees with the claim expressed in tweet t_2.

The PDisG graph shows discrepancies between a pair of tweets only if there is some (explicit) criticism relationship between them and, thus, indirect criticism relations between tweets have not yet been considered in our model. For instance, consider a Twitter discussion with tweets t_1, t_2 and t_3. Suppose that t_1 criticizes t_2 and t_3 criticizes t_1; i.e. suppose $\{(t_1, t_2), (t_3, t_1)\} \subseteq E$, $P(t_1, t_2) > 0$ and $P(t_3, t_1) > 0$. In our current approach, we restrict ourselves to considering that t_3 criticizes t_2 iff t_3 answers (explicitly replies to or mentions) t_2. The reason is that the information contained in a typical tweet, written in natural language and possibly with other attributes, almost never allows us to consider a sound way to assess an indirect criticism relation between tweets t_3 and t_2 if t_3 does not directly reply to or mention t_2.

Since the social network we are considering in this work is Twitter, every tweet of a discussion can reply at most to one tweet, but can mention many tweets, and all of them are prior in the discussion. Thus, every tweet can answer and, in turn, can criticize many prior tweets, from the same author or from different authors, of the discussion. Given a tweet t_1, we consider the set of tweets $\{t_{1_{a_1}}, \ldots, t_{1_{a_n}}\}$ that t_1 is answering as those tweets including (i) the tweet that t_1 is replying to, and (ii) all the other previous tweets in the discussion by authors mentioned by t_1. Observe that any tweet in a discussion will be answering at least one other tweet, except for the root tweet of the discussion.

To compute the set of tweets $\{t_{1_{a_1}}, \ldots, t_{1_{a_n}}\}$ we access the `in_reply_to_status_id` attribute provided by the Twitter API for the tweet object to which t_1 refers that provides the tweet identifier to which t_1 replies. The set of mentions of t_1 is obtained from all authors' mentions found in the message of t_1. Every mention of an author is stored in the message with a label of the form: `@<author_identifier>`. Thus, t_1 mentions $t_{1_{a_i}}$ whenever the author's identifier of $t_{1_{a_i}}$ is in the set of mentions of t_1.

To check whether a tweet t_1 does not agree with the claim expressed in one of its answered tweets $t_{1_{a_i}}$, the system uses an automatic labeling system based on Support Vector Machines (SVM). The description of the method we used to train the SVM can be found in [3].

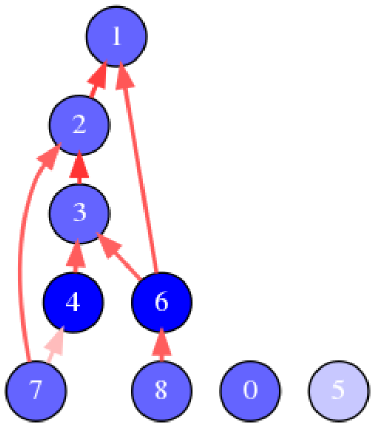

Fig. 1 PDisG graph for a Twitter discussion.

In Figure 1 we show the PDisG graph for a Twitter discussionfrom the political domain obtained by our discussion retrieval system. Each tweet is represented as a node and each criticism relationship between tweets is represented as an edge (answers with criticism probability values greater than zero). Edges are colored in *red scale*, where the darkness of the color is directly proportional to the probability that the answer is a criticism. Unconnected nodes correspond to tweets that have neither generated nor received criticism during the discussion. The root tweet of the discussion is labeled with 0 and the other tweets are labeled with consecutive identifiers according to their generation order. The discussion has a simple structure. The root tweet starts the discussion (Tweet 0) and no tweet in the discussion criticizes it. Then, all the tweets of the discussion, except tweets 0, 5, 7 and 8, are subject to criticism, tweets 1, 2 and 3 being those that concentrate most of them. The discussion contains 9 tweets and 8 criticize relations between tweets. Nodes are colored in *blue scale*, where the darkness of the color is directly proportional to the social relevance of tweets based on the `followers` weighting scheme. Notice that the graph does not contain cycles, since a tweet only answers previous tweets in the discussion.

To understand a little bit what happens in the discussion of Figure 1, we give a overall description of the different arguments presented in it. The root tweet (tweet 0) criticizes *the gag law*, following a previous discussion about a public performance of holograms walking through different streets of Madrid, that it was a way to protest against this Spanish law that restricts some ways of public protest. Then, we have a series of tweets that seem to be also criticizing the gag law: tweet 1, tweet 3, tweet 5, tweet 7 and tweet 8 (they support tweet 0 or they criticize some of the tweets that seem to not reject the gag law) and a series of tweets that seem to not reject the gag law: tweet 2, tweet 4, and tweet 6. Below we show the tweets of this discussion [1](translated from Spanish), and their social relevance (measured by their weight).

Tweet 0 text: *"@europapress The gag law was made while stoned. #THCannabis"*
Weight: 2. This tweet is the **root Tweet**.
URL: https://twitter.com/YaValeCo/status/587048825635897344

Tweet 1 text: *"@Maripac001 @joseletegrillo @europapress when laws are unfair ... they lose legitimacy ... 1/2"*
Weight: 2. This tweet answers to **Tweet 0**.
URL: https://twitter.com/KKI_666/status/587064675021488130

Tweet 2 text: *"@KKI_66 @joseletegrillo @europapress If to protest it was necessary to have the means to send out holograms we would be worse than now."*
Weight: 2. This tweet answers to **Tweet 1**.
URL: https://twitter.com/Maripac001/status/587080070608642049

Tweet 3 text: *"@Maripac001 @joseletegrillo @europapress there is nothing worse than now, they want a submissive majority based on hostilities and sanctions"*.
Weight: 2. This tweet answers to **Tweet 2**.
URL: https://twitter.com/KKI_666/status/587082166389051392

Tweet 4 text: *"@KKI_66 @europapress the law is rejected by those who harass parliaments or want to take the streets when and as they please"*.
Weight: 3. This tweet answers to **Tweet 3**.
URL: https://twitter.com/joseletegrillo/status/587154733598670848

Tweet 5 text: *"@europapress This is the Spain of freedom, equality, with the gag law, and the elite criticizes Venezuela. Are you better????"*.
Weight: 1. This tweet answers to **Tweet 0**.
URL: https://twitter.com/OlivMoon/status/587156829542682624

Tweet 6 text: *"@europapress @KKI_66 Are we holograms? We only needed more phantoms..."*
Weight: 3. This tweet answers to **Tweet 1 and Tweet 3**.
URL: https://twitter.com/olivacamposjara/status/587196628571914240

Tweet 7 text: *"@joseletegrillo @Maripac001 @europapress justice is laughable, you can see imprisoned for 800 eur and free for stealing millions. SHAME"*
Weight: 2. This tweet answers to **Tweet 2 and Tweet 4**.
URL: https://twitter.com/KKI_666/status/587272266657988609

[1] Following the URLs you can access the original tweets. At the time of writing this article the author of the root node was deleted, so the Tweet 0 is not accessible anymore. In the following link you can get the full conversation used for the example in XML format: `http://ia.udl.cat/remository/func-startdown/23/`

Tweet 8 text: *"@olivacamposjara @europapress it's not that we're holograms, it's where the situation goes, it seems to me a way of saying, we're not free"*
Weight: 2. This tweet answers to **Tweet 6**.
URL: https://twitter.com/KKI_666/status/587272876107161601

After labeling each answer (replies and mentions), with the SVM labeling system, we obtain the following results:

- Tweet 2 criticizes Tweet 1 with probability 0.843.
- Tweet 3 criticizes Tweet 2 with probability 0.871.
- Tweet 4 criticizes Tweet 3 with probability 0.768.
- Tweet 6 criticizes Tweet 1 with probability 0.781 and Tweet 3 with probability 0.758.
- Tweet 7 criticizes Tweet 2 with probability 0.761 and Tweet 4 with probability 0.527.
- Tweet 8 criticizes Tweet 6 with probability 0.767.

Notice that tweets 1 and 5 reply to Tweet 0, but they do not criticize its message, the probability of criticism being 0. From our previous discussion about the two groups of arguments we find in this discussion, we observe that the criticism relationships detected by the SVM labeling system are consistent with our identification of criticism between the two groups of tweets.

3 Valued Argumentation reasoning model

Once we have formalized the probabilistic discussion graph structure to represent probabilistic weighted discussions in Twitter, the next key component of our system is the definition of the argumentation reasoning model to obtain the set of socially accepted tweets of a discussion.

As we have already pointed out, our approach in this work is based on the argumentation system we already developed in [2] which uses a valued abstract argumentation framework to model the weighted argumentation problem associated with a discussion and ideal semantics to compute its solution (the set of socially accepted tweets).

Valued abstract argumentation is based on the extension of abstract argumentation with a valuation function *Val* on a set of values R for arguments and a (possible partial) preference relation *Valpref* between values in R. In our approach, we use Value-based Abstract Argumentation introduced by [7], also called *Audience-specific Value-based Argumentation* in [8], and we consider an uncertainty threshold α which characterizes how much uncertainty in relation to probability values we are prepared to tolerate.

The approach described in the rest of this section extends the argumentation system we have already developed in [2] to deal with probabilistic weighted relationships, since our previous approach only considers unweighted relationships. Thus, the approach we present here maps a PDisG graph to a valued abstract argumentation framework (VAF) and considers the ideal semantics defined in [16], to compute the (unique) set of consistent tweets of the discussion.

Definition 5 Let $G = \langle T, E, R, W, P\rangle$ be a PDisG graph for a Twitter discussion Γ and let $\alpha \in (0, 1]$ be a threshold on the probability values. The *Valued Argumentation Framework* for G relative to the threshold α, written VAF(G, α), is a tuple

$$\text{VAF}(G, \alpha) = \langle T, \texttt{attacks}_\alpha, R, \mathit{Val}, \mathit{Valpref}\rangle,$$

where

- each node (or tweet) in T results in an argument,
- $\texttt{attacks}_\alpha$ is an irreflexive binary relation on T and is defined according to the threshold α as follows:

$$\texttt{attacks}_\alpha = \{(t_1, t_2) \in E \mid P(t_1, t_2) \geq \alpha\},$$

- R is the non-empty set of ordered values that models the social relevance values of tweets,
- the valuation function $\mathit{Val} : T \to R$ for arguments is defined as the social relevance of tweets; i.e. $\mathit{Val}(t) = W(\mathit{fl}, r, \mathit{fv})$, for any node $t \in T$ with tweet $(m, a, \mathit{fl}, r, \mathit{fv})$, and
- $\mathit{Valpref} \subseteq R \times R$ is the ordering relation over R.

According to this formalization, given a PDisG G for a discussion Γ and a probability threshold $\alpha \in (0, 1]$, we obtain a valued argumentation framework VAF(G, α) for Γ, where arguments are tweets, argument values are the weights used to model the social relevance of tweets from data obtained from Twitter, and attacks between arguments denote answers between tweets with a probability of criticism above threshold α.

Then, given such a VAF(G, α) a *defeat* relation (or effective attack relation) between arguments (tweets) is defined according to the valuation function *Val* and the preference relation *Valpref* as follows:

$$\begin{aligned}\texttt{defeats}_\alpha = \{(t_1, t_2) \in \texttt{attacks}_\alpha \mid \\ (\mathit{Val}(t_2), \mathit{Val}(t_1)) \notin \mathit{Valpref}\}.\end{aligned}$$

The definition of the ideal semantics of a valued argumentation framework is based on the defeat relation between arguments and its ideal extension (solution) guarantees that the set of arguments in the solution is the maximal set of arguments that is consistent, in the sense that there are no defeaters among them, and all the arguments outside the solution are defeated by an argument within the solution. In our framework, since the arguments in a VAF(G, α) are tweets, we have that if a tweet outside the solution defeats a tweet within the solution, it is, in turn, defeated by another tweet within the solution. In other words, the ideal extension of a VAF(G, α) is the biggest consistent set of tweets that defeats any defeater outside the extension. In [16] the authors prove that the ideal extension of an argumentation framework is unique.

Formally, given a VAF(G, α) a set of arguments $S \subseteq T$ is *conflict-free* iff for all $t_1, t_2 \in S, (t_1, t_2) \notin \texttt{defeats}_\alpha$. Moreover, a conflict-free set of arguments $S \subseteq T$ is *maximally admissible* iff

(i) for all $t_1 \notin S$, $S \cup \{t_1\}$ is not conflict-free and
(ii) for all $t_2 \in S$, if $(t_1, t_2) \in \texttt{defeats}_\alpha$ then there exists $t_3 \in S$ such that $(t_3, t_1) \in \texttt{defeats}_\alpha$.

Each maximally admissible conflict-free set of arguments in T is defined as a *preferred extension* of the valued argumentation framework VAF(G, α).

Finally, we define what the solution of a discussion Γ is as follows.

Definition 6 (Solution of a discussion) Given a PDisG graph $G = \langle T, E, R, W, P \rangle$ of a discussion Γ and a threshold on the probability values $\alpha \in (0, 1]$, the *set of socially accepted tweets* of Γ, referred to as the *solution* of Γ relative to the thresholds α, is the ideal extension of VAF(G, α) defined as the largest admissible conflict-free set of arguments (tweets) $S \subseteq T$ in the intersection of all preferred extensions of VAF(G, α).

From the implementation point of view, the defeat relation between arguments of a valued abstract argumentation framework manages three key elements: the non-empty set of ordered values R that models the social relevance values of tweets, the valuation function *Val* for the social relevance of tweets, and the ordering relation *Valpref* over R. Because of our goal in this paper is to define a system for analyzing weighted Twitter discussions, we need to model the defeat relation between tweets in a suitable way. Our approach instantiates these three elements as follows: the social relevance of tweets is evaluated by the natural numbers $\mathbb{N}$ (i.e. the non-empty set of ordered values R is instantiated to the natural numbers $\mathbb{N}$), the valuation function *Val* is computed by means of one of the two weighting schemes, the `followers` and `f1lr20fv40` weighting schemes, and the *Valpref* ordering relation is the ordering relation $>$ over the natural numbers $\mathbb{N}$.

Figure 2 shows the solution for the PDisG graph instance of Figure 1 with probability threshold $\alpha = 0.6$ and the `followers` weighting scheme. The nodes colored in blue are the tweets in the solution and the nodes colored in gray are the rejected tweets, where the darkness of the color is directly proportional to its weight. Moreover, the edges colored in red are defeat relations between tweets (effective attacks) while the edges colored in black have no effect on the solution and can be pruned. Thus, Tweet 0, Tweet 4, Tweet 5, Tweet 6, Tweet 7 and Tweet 8 are accepted, since they do not have any defeater in the discussion. However, Tweet 1, Tweet 2 and Tweet 3 are rejected since all of them have a defeater in the solution. So, observe that the solution includes the root tweet (that was against the Spanish gag law) and other tweets against the Spanish gag law (tweets 1, 5, 7 and 8), but also contains some tweets that seem to not directly reject that Spanish law: tweets 4 and 6, because tweets 4 and 6 are attacked by tweets 7 and 8, but these attacks are not defeats, because of the weight of those arguments. Thus, finally the solution contains tweets 4 and 6 but also tweets 7 and 8.

The fact that the solution for a VAF(G, α) can contain arguments (tweets) with attacks between them is due to the fact that not effective attacks (those not accepted as defeats) are considered as having no effect in the consistency of the set of arguments accepted, and it is a natural property of any argumentation semantics for valued argumentation frameworks that works with the notion of defeats. However, it is true that it seems to be a little bit undesirable to have as solution a set of tweets with some attacks between them, even if we consider those attacks to be not *strong enough* to be taken into account (given the weights of the tweets and the probability of criticism between those tweets). We discuss this issue further in the future work section.

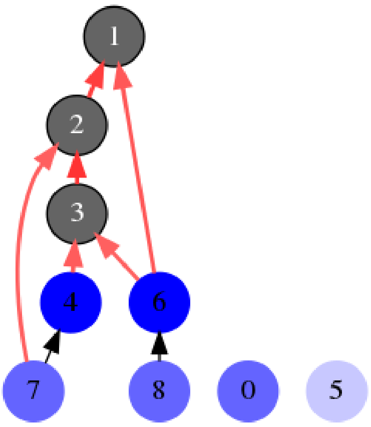

Fig. 2 PDisG graph solution with probability threshold $\alpha = 0.6$.

4 Distributed strategy

In this section we define a distributed strategy to implement the underlying valued argumentation reasoning model to compute the solution of a Twitter discussion, defined in the previous section.

The strategy consists of a *pre-processing step* of a PDisG graph of a discussion Γ and a *distributed computation algorithm* to compute the ideal extension of an argumentation graph.

From now on, we will evaluate the social relevance of tweets by means of a weighting scheme W on the natural numbers $\mathbb{N}$ and will take the ordering relation $>$ over $\mathbb{N}$ to define the defeat relation between tweets.

Given a PDisG graph $G = \langle T, E, \mathbb{N}, W, P \rangle$ of a discussion Γ, the pre-processing step prunes all attacks in the valued argumentation framework VAF(G, α) that are not effective based on the weight of nodes W and the ordering relation $>$ over $\mathbb{N}$. Accordingly, the output of the pre-processing step is an (unweighted) argumentation graph and it represents a set of arguments and counterarguments: each node is an argument (a tweet) and each edge denotes an effective attack (a defeat) between arguments. We will refer to this argumentation graph as *Defeats Graph* for Γ and it is defined from the PDisG graph G and the probability threshold α.

In what follows, given $t_1 = (m_1, a_1, fl_1, r_1, fv_1)$ and $t_2 = (m_2, a_2, fl_2, r_2, fv_2)$ and a probability threshold α, we will say that t_1 *defeats* t_2, written $t_1 \rightsquigarrow_\alpha t_2$, if $(t_1, t_2) \in E$ with $P(t_1, t_2) \geq \alpha$ and $W(fl_1, r_1, fv_1) \geq W(fl_2, r_2, fv_2)$. Based on this notation we formalize the *Defeats Graph* structure as follows.

Definition 7 Let $G = \langle T, E, \mathbb{N}, W, P \rangle$ be a PDisG graph for a Twitter discussion Γ and let $\alpha \in (0, 1]$ be a threshold on the probability values. The *Defeats Graph* of Γ, relative to the threshold α, is the directed graph (T, D_α) where for every tweet in Γ there is a node in T and the set of edges D_α is the defeat relation between tweets in Γ, i.e

$$D_\alpha = \{(t_1, t_2) \in E \mid t_1 \rightsquigarrow_\alpha t_2\}.$$

Based on this definition we get that the edges in D_α correspond to the effective attacks (defeats) between tweets (arguments) in $\texttt{defeats}_\alpha$ of VAF(G, α), considering the ordering relation $>$ over $\mathbb{N}$. Then, the Defeats Graph (T, D_α) of Γ is an argumentation graph where each node (tweet) in T is an argument and each edge in D_α is an effective attack (a defeat) between arguments. Thus, the *solution* of Γ relative to the threshold α corresponds to the ideal extension of the argumentation graph (T, D_α), since the ideal extension of the valued argumentation framework VAF(G, α) of Γ is defined through $\texttt{defeats}_\alpha$.

Formally, given a PDisG graph $G = \langle T, E, \mathbb{N}, W, P \rangle$ of a discussion Γ, the *solution* of Γ relative to a thresholds $\alpha \in (0, 1]$, is the largest admissible conflict-free set of tweets $S \subseteq T$ in the intersection of all preferred extensions of the Defeats Graph (T, D_α).

Figure 3 shows the Defeats Graph of the PDisG graph of Figure 1 with the probability threshold $\alpha = 0.6$. Note that the Defeats Graph is defined by the tweets involved in the discussion and the effective attacks (defeats) between them without reflecting the weights used for their computation. As shown in Figure 2, the solution of the discussion (the set of accepted tweets) is the set of tweets $\{0, 4, 5, 6, 7, 8\}$, while the set $\{1, 2, 3\}$ is the set of rejected tweets.

In the next section we introduce and investigate a distributed implementation of the ideal semantics approach for computing the solution of Twitter discussions based on the Defeats Graphs. In the social network Twitter, a tweet always answers or refers to previous tweets in the discussion and thus, a Defeats Graph of a discussion Γ is always an acyclic argumentation graph. Therefore, the input to the distributed computation algorithm is the (acyclic) argumentation graph (T, D_α) of a PDisG graph $G = \langle T, E, \mathbb{N}, W, P \rangle$ and a threshold on the probability values α based on the ordering relation $>$ over $\mathbb{N}$.

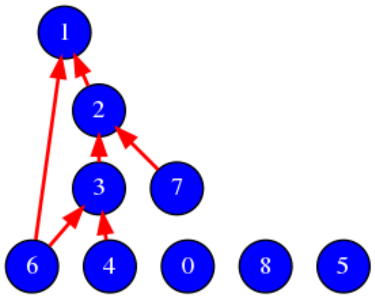

Fig. 3 Defeats Graph with $\alpha = 0.6$ and the `followers` scheme.

4.1 Skeptical approach based on the ideal semantics

We design a distributed strategy to compute the solution of a Defeats Graph of a discussion Γ using the distributed model of computation of Pregel [30]. This model is appropriate for our problem, because the input for a Pregel algorithm is a directed graph, where the nodes can be in different states, and the goal of a distributed algorithm in Pregel is to compute the state of each node based on the state of the nodes' neighbors. Any Pregel algorithm starts initializing each node to some initial state. Then, the distributed computation follows a sequence of *supersteps* separated by global synchronization points until the algorithm finishes a point where every node is satisfied with its current state. This computation model is actually inspired by Valiant's Bulk Synchronous Parallel model [33].

Within each superstep the nodes compute their state in parallel, executing a specific function that computes the new state of the node taking into account the possible messages sent by the nodes' neighbors in the *previous* superstep. Then, as a byproduct of the computation of the node state, some messages may be sent to some of the nodes' neighbors, that will be processed in the next superstep. The idea is that the messages are used by nodes to indicate some change in their state, which could have some influence on the state of their neighbors in the next superstep. The superstep finishes when every node has computed its state and has sent the necessary messages to its neighbors.

The distributed algorithm we present here computes the ideal extension (solution) of a Defeats Graph of a discussion Γ in polynomial time. This is possible because a Defeats Graph is an acyclic argumentation graph and acyclic argumentation graphs are *cohesive* and *coherent*. These two properties allow us to define such tractable computation.

On the one hand, the *cohesive* property of an argumentation graph ensures [19] that its ideal extension coincides with its skeptical extension, where the skeptical extension is the intersection of its preferred extensions. And thus, a cohesive argumentation graph verifies that the intersection of its preferred extensions is a maximally admissible conflict-free set of arguments.

On the other hand, the *coherent* property of an argumentation graph ensures [15] that each preferred extension is a stable extension, where a stable extension is a conflict-free set of arguments S such that every argument not in S is defeated by some argument in S. So, a coherent argumentation graph satisfies that an argument belongs to the skeptical extension if and only if all its defeaters are not admissible.

Finally, as an acyclic argumentation graph has an unique preferred extension, this extension is obviously both the skeptical and ideal extension, since it is cohesive, and any argument not in this extension is not admissible because it is defeated by some argument in the extension, since it is also coherent. And thus, all the work needed to compute the ideal extension (solution) of a discussion Γ is to determine the set of not admissible arguments of the Defeats Graph (T, D_α), given a PDisG graph $G = \langle T, E, \mathbb{N}, W, P\rangle$ of Γ and a probability threshold $\alpha \in (0, 1]$.

In the distributed model of Pregel, we take advantage of these properties to design an algorithm where a node is accepted in the solution if and only if all its defeating nodes are not accepted (or if it has no defeating nodes). This recursive acceptance condition is well defined for the case of an acyclic argumentation graph, as well as for some other case(s) that we will consider later.

In our distributed algorithm, each node can be in three states: undefined, accepted or rejected, but a node also stores an accepted defeaters counter and a rejected defeaters counter, to keep track of the number of *accepted defeaters* and the number of *rejected defeaters* of the node in the current state of the algorithm. Initially, every node starts in the undefined state and with its accepted defeaters and rejected defeaters counters equal to zero.

Algorithm 4.1 shows the pseudocode of the procedure `CompAcceptance` used by each node a to compute its state in a superstep i. [2] The algorithm works as follows. In the superstep i, a node a first checks if it has already reached the state of accepted or rejected, in which case it can soundly finish its computation (lines 3 to 5) as once it reaches one of these two states, that state will be the final one in the solution, and therefore node does not need to process any other incoming messages from its defeater nodes. If the state is still undefined, and there are incoming messages, the node updates its accepted defeaters and rejected defeaters counters, adding all the new accepted defeaters and rejected defeaters that are indicated in the newly received messages (lines 6 to 8).

There are two possible changes in the state of the node that the updating of the counters can produce. On the one hand, if the rejected defeaters counter is equal to its total number of defeaters, that means that all its defeaters are definitively rejected (they are not in the solution), and this makes the node be accepted. Then, the node informs all its defeated nodes about this fact, by sending a new message where it indicates to them that they have a new accepted defeater (lines 9 to 11). On the other hand, if the accepted defeaters counter is greater than zero, this means that the node has at least one accepted defeater, and this makes the node be rejected. Then, the node informs all its defeated nodes about this fact, this time by sending a new message where it indicates to them that they have a new rejected defeater (lines 12 to 14). If the new values of the counters do not produce any change in the undefined state of the node (there are new rejected defeaters, but not accepted defeaters) the node votes for the termination of the distributed execution (lines 15 to 16). Note that in the Pregel computation model, computation continues to the next superstep as long as there is at least one node that sent a message in the previous superstep.

As a running example, consider solving the Defeats Graph shown in Figure 3, whose solution is the set of blue nodes shown in Figure 2 (tweets $\{0, 4, 5, 6, 7, 8\}$):

1. Before the first superstep, all the nodes start in the undefined state, and with its accepted defeaters and rejected defeaters counters equal to zero.
2. In the first superstep, only nodes $0, 4, 5, 6, 7$ and 8 change their state to accepted, since they do not have any defeater, and therefore they send messages to their defeated nodes informing them that they have a new accepted defeater.
3. In the second superstep, nodes $1, 2$ and 3 receive messages (from the first superstep), informing them about accepted defeaters. Therefore, because all of them have at least one accepted defeater, they change their state to rejected. After this change, node 3 sends a message node 2, informing to node 2 that it has a new rejected defeater, and analogously, for nodes 2 and 1.
4. In the third superstep, nodes 2 and 1 receive the messages from superstep 2, informing them about new rejected defeaters. However, since they have already changed their state to rejected (because they have accepted defeaters), they do not change their state, and therefore no node sends any messages in this superstep.

[2] The pseudocode is written using object-oriented notation, as the Pregel API is written in C++. However, our actual implementation is based on the Pregel implementation found in the Spark distributed programming framework, graphX, which is written in Scala.

Algorithm 1 Compute acceptance state for node a

```
procedure $a$.COMPACCEPTANCE($i$)                                  ▷ Update acceptance state of $a$ at superstep i

    if ( a.State() != Undef) then
        a.VoteToHalt()
        return                                                       ▷ Finish if state already fixed
    for all $msg \in$ a.received(i-1) do
        $a$.NumAccepted := $a$.NumAccepted + $msg$.newaccepted
        $a$.NumRejected := $a$.NumRejected + $msg$.newrejected
    if ( a.NumRejected() == a.NumDefeaters()) then                  ▷ all defeaters are rejected, so $a$ is accepted
        a.setAccepted()
        sendToAllNeighbours( msg( (1, 0 ) ) )                      ▷ Tell defeated nodes to add one accepted defeater
    else if (a.NumAcepted() $\geq$ 1) then                           ▷ there is an accepted defeater, $a$ is rejected
        a.setRejected()
        sendToAllNeighbours( msg( (0, 1) ) )                       ▷ Tell defeated nodes to add one rejected defeater
    else
        a.VoteToHalt()                                               ▷ Vote to finish distributed computation
```

5. Because at the end of the third superstep no nodes send any messages, the execution of the distributed algorithm ends after the third superstep.

The next lemma shows that the skeptical (ideal) extension is correctly computed by our distributed algorithm for the case of an acyclic argumentation graph with a number of supersteps that it is linear with respect to the maximum distance of any node in the graph. We define the distance of a node a in an argumentation graph, written $distance(a)$, as the maximum number of nodes in any path between a and the leaf nodes (nodes with an input degree equal to zero) connected with a. For leaf nodes the distance is equal to 1.

Lemma 1 (The set of accepted arguments coincides with the skeptical extension) *At the end of superstep i, any node a with $distance(a) \leq i$ is already correctly labeled (as accepted if it belongs to the skeptical extension and as rejected, otherwise) by Algorithm 4.1.*

Proof: By induction.

- In the first superstep, any node with no defeaters (and so, with distance 1), is accepted. Therefore, all nodes with such a distance are correctly labeled, since if they have no defeaters they belong to the skeptical extension.
- Assume that at the end of superstep $i-1$ any node with distance $\leq i-1$ is correctly labeled. Then, any node a with distance i can be in two situations. By the induction hypothesis, all its defeaters are correctly labeled. Then, either a has an accepted defeater in a superstep $i' \leq i-1$, and therefore its state is set to rejected in superstep $i'+1$, or any defeater of a has been identified as rejected, and therefore a must be accepted in superstep i. In both cases, the label of a is correct (with respect to the skeptical extension) if the labels of its defeaters are correct.

In a recent study, we proposed a similar distributed algorithm for the same case of acyclic argumentation graphs [4]. The differences are that in our previous algorithm a node could change its state several times during the execution of the algorithm, and as a consequence of this difference the total number of messages sent could be higher. In the version presented here, a node changes its state only once and once this happens this node does not send any other messages.

To test the performance of our algorithm we selected a test set of Twitter discussions and solved them in a computer with 4 CPU cores both, with our distributed algorithm, and with the general sequential algorithm we have developed in [2] that solves any kind of discussions, not only acyclic ones. To obtain the Defeats Graph instances for the discussions, we considered the probability threshold $\alpha = 0.5$ and the `f1lr20fv40` weighting scheme. Table 1 shows the results obtained. We show the number of nodes (tweets), the number of attacks in $\mathtt{attacks}_\alpha$ (answers between tweets t_1 and t_2 classified as criticism messages with a probability $P(t_1, t_2) \geq \alpha$), the number of defeats in $\mathtt{defeats}_\alpha$ (attacks with $\mathtt{f1lr20fv40}(t_1) \geq \mathtt{f1lr20fv40}(t_2)$), the percentage of accepted nodes in the solution, as well as the performance of the distributed algorithm in terms of time spent, both the real time (real execution time) and the user time (sum of the time used by all the CPU cores), and the number of messages sent between nodes during the execution of the algorithm.

For the range of sizes tested, assuming the number of edges as the size, the scaling of the real time is at most linear with respect to the size for the distributed algorithm. We also observe that comparing the real time with the user time, the user time is between 2 and 3 times bigger than the real time, meaning that the parallelization exploited by our algorithm is taking advantage of the available computational resources. By contrast, the scaling of the user time for the sequential algorithm seems to be exponential, and working with a cutoff time of 1 hour, from the instance in row 7 to the last one, all of them cannot be solved within that cutoff time.

5 Author-centered model

In this work we also introduce and investigate an author-centered reasoning model for mining consistent authors' opinions of discussions based on the distributed approach presented in the previous section. To this end, starting from the Defeats Graph (T, D_α) of a discussion Γ relative to a threshold α, we built an authors' opinions graph where tweets are grouped by authors and effective attacks or defeat relations between tweets denote controversies at the level of authors.

The author-centered approach allows us to perform a different analysis from the one that results from the tweet-based approach proposed in Section 2. Aggregating the information by author allows us to mind the set of authors whose opinions are consistent or in agreement in the discussion, the authors involved in a circular argumentative discussion and the most controversial authors. That is, for instance, the authors who receive the greatest number of effective attacks, the authors who participate in the greatest number of cycles or the authors that generate the longest argumentative chains.

We consider discussions in which every author's opinion is consistent, discussions in which authors are not self-referenced and do not contradict themselves. That is, for any author a and any pair of tweets from author a $t_1 = (m_1, a, fl_1, r_1, fv_1)$ and $t_2 = (m_2, a, fl_2, r_2, fv_2)$, we assume that messages m_1 and m_2 do not express either conflicting or inconsistent information. Although the consistency of author's opinions can be something that it may not hold always, we believe that in most cases, or at least in discussions where authors think in a rational way when posting their opinions, this is a natural assumption. Next we formalize the notion of an author's opinion.

Definition 8 Let Γ be a Twitter discussion with authors' identifiers $\{a_1, \ldots, a_n\}$. The *Opinion* of an author $a_i \in \{a_1, \ldots, a_n\}$ in the discussion, denoted T_{a_i}, is the set of tweets of a_i in Γ:

$$T_{a_i} = \{(m, a_i, fl, r, fv) \in \Gamma\}.$$

Given a Twitter discussion Γ, in what follows we assume that every author a_i can be represented by his/her opinion T_{a_i}, since for each discussion there is a unique set of authors, and for each author there is a unique opinion in the discussion. Therefore, we shall refer to both terms indistinctly.

In a previous study [1], developed jointly with Francesc Esteva and Lluis Godo, we defined a probabilistic aggregation model to combine probabilistic weighted relationships between tweets which depends on the semantics assumed for the probability that a relationship between two authors' opinions exists from the probabilities of relationships between the tweets of those authors. Now, in this work, we explore and analyze the results provided by the Distributed Algorithm 4.1 to mind consistent sets of authors' opinions. Therefore, we define a new aggregation model, different from the one proposed in [1], to represent author-centered discussions which is based on the Defeats Graph (T, D_α) of a discussion Γ.

Our approach consists of formalizing an argumentation graph to represent Twitter discussions in which each node denotes an author's opinion (argument) and each edge denotes a defeat relationship between two authors' opinions. We will refer to this graph as Author's Defeats Graph of Γ.

In the following, let Γ be a Twitter discussion with authors' identifiers $\{a_1, \ldots, a_n\}$ and let (T, D_α) be the *Defeats Graph* of Γ relative to the threshold α. Suppose further that we have two authors' opinions or sets of authors' tweets T_a and T_b. Next, we define and analyze some semantic patterns for the interpretation of the effective attacks between the authors' opinions T_a and T_b. We propose three semantic patterns ranging from a skeptical semantics to a credulous semantics through an intermediate semantics:

Skeptical semantics: A skeptical notion of defeat between T_a and T_b can be defined as follows: T_a defeats T_b when every tweet in T_b is defeated by some tweet in T_a, i.e. when for all $t \in T_b$, there is $t' \in T_a$ such that $t' \rightsquigarrow_\alpha t$.

Credulous semantics: A credulous notion of defeat between T_a and T_b can be defined as follows: T_a defeats $T_b$when there is at least one tweet $t \in T_b$ that is defeated by a tweet $t' \in T_a$, i.e. when $t' \rightsquigarrow_\alpha t$ for some $t \in T_b$ and $t' \in T_a$.

Intermediate semantics: A more flexible definition of when T_a defeats T_b is to stipulate that this holds when for *most* of the tweets $t \in T_b$ there is a tweet $t' \in T_a$ such that $t \rightsquigarrow_\alpha t'$. The question with the intermediate pattern is how we interpret the quantifier *most*. A first option is to understand *most* as a proportion of at least r, for some $0.5 \leq r < 1$ to be chosen.

At this point we are ready to formalize the Authors' Defeats Graph of Γ, which is relative to the Defeats Graph (T, D_α) and the semantic pattern for the interpretation of effective attacks between the authors' opinions.

Definition 9 Let (T, D_α) be the Defeats Graph of a discussion Γ and let r be a proportion value in the real interval $(0, 1]$. The *Author's Defeats Graph* of Γ relative to the proportion value r is the directed graph $(\mathcal{T}, \mathcal{D}^r_\alpha)$ where:

- The set of nodes $\mathcal{T}$ is the set of authors' opinions:

$$\mathcal{T} = \{T_{a_1}, \ldots, T_{a_n}\}.$$

		# edges			distributed algorithm			sequential algorithm
Discussion ID	# nodes	`attacks`$_\alpha$	`defeats`$_\alpha$	% accepted	real time	user time	# messages	user time
867742319088467968	21	17	2	95%	3.84s	8.56s	2	1.65s
867841549085929472	32	2	2	94%	3.51s	7.64s	2	8.82s
865615862291718144	40	58	23	78%	3.93s	8.90s	19	13.59s
867627523059961856	64	14	5	91%	3.75s	8.46s	6	132.09s
574324656905281538	74	123	69	74%	4.51s	10.98s	38	162.81s
867756958337683457	138	153	22	90%	5.32s	13.59s	19	3134.16s
867651560406478848	153	3	2	99%	3.53s	7.73s	2	> 1 hour
868086232932376580	159	95	17	92%	4.61s	11.21s	16	> 1 hour
867751574436761600	213	106	61	86%	4.43s	10.72s	63	> 1 hour
866895801494228993	380	2520	955	77%	6.06s	15.60s	519	> 1 hour
867494068917608448	387	5234	2757	67%	7.46s	18.94s	1210	> 1 hour
865925019007954944	447	630	214	90%	5.82s	15.03s	198	> 1 hour
867052819878203394	520	2247	1493	66%	7.70s	20.21s	865	> 1 hour
867494287122059264	590	11091	5957	53%	8.90s	21.73s	1743	> 1 hour
866624474568953857	789	6552	4431	60%	9.70s	23.22s	2337	> 1 hour
867693522195034112	898	8158	4788	57%	11.70s	28.75s	1880	> 1 hour
868072573002821635	932	10286	3381	64%	11.21s	26.70s	2364	> 1 hour

Table 1 Performance results of our distributed algorithm when solving Twitter discussions (identified by their root tweet ID) with the probability threshold $\alpha = 0.5$ and the `fl1r20fv40` weighting scheme. For each discussion we show the number of nodes (tweets), the number of edges (attacks and defeats) and the percentage of nodes in the computed solution (accepted tweets). Performance measured by real execution time, user execution time, and number of messages sent. The last column shows the user time for the general sequential algorithm [2].

- The set of edges $\mathcal{D}^r_\alpha$ is the defeat relation between the authors' opinions and is defined as follows:

  For each pair of authors $a_i, a_j \in \{a_1, \ldots, a_n\}$, with $a_i \neq a_j$, we define the set of tweets of a_j defeated by a_i, denoted $T^{a_i}_{a_j}$, as follows:

  $$T^{a_i}_{a_j} = \{t_2 \in T_{a_j} \mid (t_1, t_2) \in D_\alpha \textit{ with } t_1 \in T_{a_i}\}.$$

  Then, there is an edge (T_{a_i}, T_{a_j}) in $\mathcal{D}^r_\alpha$ iff $\frac{|T^{a_i}_{a_j}|}{|T_{a_j}|} \geq r$.

Only the nodes and edges obtained by applying this process belong to $\mathcal{T}$ and $\mathcal{D}^r_\alpha$, respectively.

Note that for $r = 1$ we obtain the Author's Defeats Graph based on the skeptical semantic pattern, while for values close to 0 we obtain the credulous semantic pattern. Moreover, the proportionality parameter r can be used as an element of analysis of discussions between authors, since it allows us to analyze to what extent the defeats relations between tweets determine the status of each author's opinion in the solution of the Twitter discussion. That is, the higher the value of r, the lower the number of defeats between authors and, therefore, the greater the number of accepted opinions in the solution. Meanwhile, for intermediate values, we can expect a higher number of defeats and, thus, a lower number of accepted opinions in the solution.

Figures 4 and 5 show the Author's Defeats Graph for the Defeats Graph of Figure 3, based on the intermediate semantics pattern with proportion value $r = 0.5$ and the credulous semantics, respectively. In the discussion, six authors are involved, with node $a5$ being the author of the root tweet (Tweet 0 in the Defeats Graph of Figure 3), node $a4$ being the author of Tweet 5, node $a3$ being the author of Tweet 4, node $a2$ being the author of the set of tweets $\{1, 3, 7, 8\}$, node $a1$ being the author of Tweet 2 and node $a0$ being the author of Tweet 6. As expected, credulous semantics gives rise to a greater number of defeats between authors' opinions and, therefore, facilitates the appearance of cycles as the graph of Figure 5 shows. Differences between both argumentation graphs are due to the fact that authors $a3$ and $a1$ only defeat one of the four tweets of author $a2$, leading to defeats relations between authors that depend on the semantic pattern that is considered. However, in both Author's Defeats Graphs, the defeats relationships between authors are concentrated around author $a2$ giving rise to the same solution, according to the ideal semantics for argumentation graphs. Thus, in this Twitter discussion, for both semantics of defeats between author's opinions, $\{a0, a1, a3, a4, a5\}$ is the set of accepted authors' opinions (the ideal extension) and only the opinion of author $a2$ is rejected, $a2$ being the author of the set of tweets $\{1, 3, 7, 8\}$. Remember that in the tweet-based model, the set of accepted tweets of the discussion is $\{0, 4, 5, 6, 7, 8\}$, while the set $\{1, 2, 3\}$ is the set of rejected tweets. When we analyze the relationship between the tweet-based and author-centered models, we see that $a2$ is the author of two of the three rejected tweets in the tweet-based model, which causes the rejection of author $a2$ in the author-centered model and, in turn, allows author $a1$ to be accepted since his tweet (Tweet 2) is only defeated by tweets 3 and 7 in the tweet-based model, both by author $a2$.

As with the Defeats Graph of a Twitter discussion, the Author's Defeats Graph also corresponds to an (unweighted) argumentation graph with arguments and counterarguments. Thus, given an Author's Defeats Graph $(\mathcal{T}, \mathcal{D}^r_\alpha)$ of a discussion Γ, the set of authors' opinions $\mathcal{T}$ denotes a set of argu-

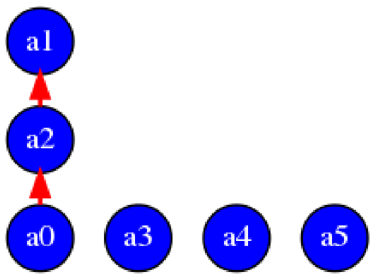

Fig. 4 Author's Defeats Graph with proportion value $r = 0.5$.

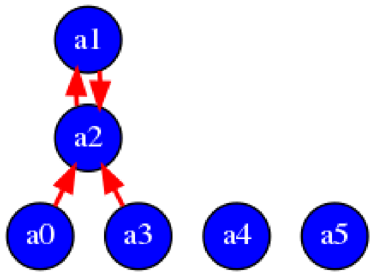

Fig. 5 Author's Defeats Graph based on credulous semantics.

ments and $\mathcal{D}^r_\alpha$ denotes a defeat relation between arguments in $\mathcal{T}$. So, according with our argumentation based reasoning approach, the *set of socially accepted author's opinions* of Γ (i.e. the authors in the solution of Γ) corresponds to the ideal extension of the Author's Defeats Graph $(\mathcal{T}, \mathcal{D}^r_\alpha)$.

Formally, given an Author's Defeats Graph $(\mathcal{T}, \mathcal{D}^r_\alpha)$ of a discussion Γ, the *set of socially accepted author's opinions* is the largest admissible conflict-free set of author's opinions $\mathcal{S} \subseteq \mathcal{T}$ in the intersection of all preferred extensions of the argumentation graph $(\mathcal{T}, \mathcal{D}^r_\alpha)$.

Although both the Defeats Graphs and the Author's Defeats Graphs are interpreted as argumentation graphs and both solutions are defined as the ideal extension, the structure of both graphs is different. A Defeats Graph is always acyclic and an Author's Defeats Graph can contain cycles, since the defeat relation between authors in a discussion can give rise to circular relationships, as occurs with the case of the Author's Defeats Graph based on credulous semantics of Figure 5.

In the next section we show that Algorithm 4.1 can also be used to compute the solution of an Author's Defeats Graph whenever it is a bipartite graph (with and without cycles). [3]

6 Skeptical approach for bipartite graphs

A bipartite Author's Defeats Graph represents a discussion in which authors divide into two opposite sides, with one side attacking (defeating) the arguments of the other side. Therefore, we believe that this particular case can be of relevance in the case of typical discussions where authors defend one of two opposite ideas.

Computing the ideal extension of a bipartite argumentation graph has been shown to be tractable [19].

A bipartite argumentation graph $(\mathcal{Y} \cup \mathcal{Z}, \mathcal{D})$ satisfies, like acyclic ones, that they are cohesive and coherent. Thus, the ideal extension is equal to the skeptical extension and to compute the skeptical extension we have to recognize the set of not admissible arguments.

In [17] the authors present a sequential polynomial-time algorithm, that we show here in Algorithm 2, and they prove that it computes the set of admissible arguments from the partition $\mathcal{Y}$ of $(\mathcal{Y} \cup \mathcal{Z}, \mathcal{D})$, and to compute the admissible ones from $\mathcal{Z}$ we simply swap $\mathcal{Y}$ for $\mathcal{Z}$ in the algorithm.

Algorithm 2 Computing admissible arguments from $\mathcal{Y}$ in $(\mathcal{Y} \cup \mathcal{Z}, \mathcal{D})$

```
1: $k := 0; \mathcal{W}_0 := \mathcal{Y}; \mathcal{D}_0 := \mathcal{D}$
2: repeat
3:     $k := k + 1$
4:     $\mathcal{U}^Y_k := \{y \in \mathcal{W}_{k-1} \mid \exists z, \langle z, y\rangle \in \mathcal{D}_{k-1}$ and
           $\nexists y', \langle y', z\rangle \in \mathcal{D}_{k-1}\}$
5:     $\mathcal{W}_k := \mathcal{W}_{k-1} \setminus U^Y_k$
6:     $\mathcal{D}_k := \mathcal{D}_{k-1} \setminus \{\langle y', z\rangle \mid y' \in U^Y_k\}$
7: until $(\mathcal{W}_k = \mathcal{W}_{k-1})$
```

We prove that Algorithm 2, which stores admissible arguments in the set $\mathcal{W}_k$ and not admissible ones in the sequence of sets $\mathcal{U}^Y_1, \ldots, \mathcal{U}^Y_k$, performs the same computation as the distributed algorithm we presented before, when solving a bipartite argumentation graph, in the sense that the set of rejected arguments in our distributed algorithm coincides with the set of not admissible arguments identified by this sequential algorithm when solving a bipartite argumentation graph. Informally, this is due to the fact that in a bipartite argumentation graph, not admissible arguments y (recognized in line 4 of Algorithm 2) are those defeated by an argument z such that z is in the skeptical extension, as z is either not defeated or defeated only by not admissible arguments.

To compare the set of rejected arguments in our distributed algorithm with the not admissible ones (sets $\mathcal{U}^Y_k$) in Algorithm 2, we observe that the supersteps of Algorithm 1 alternate between computing accepted arguments (in supersteps in odd positions ≥ 1) and computing rejected arguments (in supersteps in even positions ≥ 2). That is, the first superstep computes accepted arguments with distance 1 (the ones without defeaters), then the second superstep computes rejected arguments that are defeated by the accepted ones of the first superstep, and so on. Thus, we can define the sets S_k and R_k as the set of accepted and rejected arguments computed in supersteps $2k - 1$ and $2k$, respectively, with $k \geq 1$. As we now consider a bipartite argumentation graph

[3] A bipartite graph (or bigraph) is a graph whose nodes can be divided into two disjoint sets such that no two graph nodes within the same set are adjacent.

$(\mathcal{Y} \cup \mathcal{Z}, \mathcal{D})$, we also need the filtered (by $\mathcal{Y}$ or by $\mathcal{Z}$) version of these sets:

$$S_k^Y = \{y \mid y \in \mathcal{Y} \text{ and } y \in S_k\}$$

$$R_k^Y = \{y \mid y \in \mathcal{Y} \text{ and } y \in R_k\}$$

and analogously, for the other partition $\mathcal{Z}$.

We can also write the sets $\mathcal{W}_k$ and $\mathcal{D}_k$, iteratively computed by Algorithm 2, in the following form:

$$\mathcal{W}_k = \mathcal{Y} \setminus \mathcal{U}_{\leq k}^Y$$

and

$$\mathcal{D}_k = \mathcal{D} \setminus \{\langle y, z\rangle \mid y \in \mathcal{U}_{\leq k}^Y\}$$

where

$$\mathcal{U}_{\leq k}^Y = \bigcup_{j=1}^{k} \mathcal{U}_j^Y$$

for the following proposition, we will also need the analogous cumulative version of the sets of rejected arguments R_k^Y:

$$R_{\leq k}^Y = \bigcup_{j=1}^{k} R_j^Y$$

Proposition 1 (Correctness of the distributed algorithm for bipartite argumentation graphs) *Given a bipartite argumentation graph $(\mathcal{Y} \cup \mathcal{Z}, \mathcal{D})$, the sets $R_{\leq k}^Y$ and $\mathcal{U}_{\leq k}^Y$ are the same for any $k \geq 1$, as well as the sets $R_{\leq k}^Z$ and $\mathcal{U}_{\leq k}^Z$. Therefore, the set of rejected arguments from our distributed algorithm coincides with the set of not admissible arguments.*

Proof: We show it by induction for the sets $R_{\leq k}^Y$ and $\mathcal{U}_{\leq k}^Y$; the proof is analogous for the other two sets. We first prove that $R_1^Y = \mathcal{U}_1^Y$. First, assume that $y \in R_1^Y$. Then, it follows that $\exists z, \langle z, y\rangle \in \mathcal{D}$ and $z \in S_1^Z$, and therefore z has no defeaters in $\mathcal{D}$. However, because $\mathcal{D}_0 = \mathcal{D}$ and $\mathcal{W}_0 = \mathcal{Y}$, we then have that $y \in \mathcal{W}_0, \exists z, \langle z, y\rangle \in \mathcal{D}_0, \nexists y', \langle y', z\rangle \in \mathcal{D}_0$. This means that y belongs to $\mathcal{U}_1^Y$.

Second, assume that $y \in U_1^Y$. Then, it follows that $y \in \mathcal{W}_0, \exists z, \langle z, y\rangle \in \mathcal{D}_0, \nexists y', \langle y', z\rangle \in \mathcal{D}_0$. However, as we have said, $\mathcal{D}_0 = \mathcal{D}$ and $\mathcal{W}_0 = \mathcal{Y}$. This therefore, means that y is also in R_1^Y because it has a defeater $z \in \mathcal{Z}$ and z has no defeater in $\mathcal{D}$.

For the induction step, assume that $R_{\leq k-1}^Y = U_{\leq k-1}^Y$; we then prove that $R_{\leq k}^Y = \mathcal{U}_{\leq k}^Y$. We first prove that $R_k^Y \subseteq U_k^Y$. Assume $y \in R_k^Y$. Then, it follows that $\exists z, \langle z, y\rangle \in \mathcal{D}$ and $z \in S_k^Z$, and therefore $\forall \langle y', z\rangle \in \mathcal{D}, y' \in R_{\leq k-1}^Y$ and by the induction hypothesis $y' \in U_{\leq k-1}^Y$. Then, $y \in \mathcal{W}_{k-1}$ (because if it is in R_k^Y it was not in $U_{\leq k-1}^Y$) and $\exists z, \langle z, y\rangle \in \mathcal{D}_{k-1}, \nexists y' \langle y', z\rangle \in \mathcal{D}_{k-1}$, as any such y' is in $U_{\leq k-1}^Y$. Therefore, y must be also in U_k^Y.

Next, we prove that $\mathcal{U}_k^Y \subseteq R_k^Y$. Assume $y \in \mathcal{U}_k^Y$. Then, it follows that $\exists z, \langle z, y\rangle \in \mathcal{D}_{k-1}, \nexists \langle y', z\rangle \in \mathcal{D}_{k-1}$, and therefore any such y' is in $U_{\leq k-1}^Y$ and by the induction hypothesis $y' \in R_{\leq k-1}^Y$. Then, z is accepted by our algorithm in superstep $2k - 1$ (because if $y \in \mathcal{U}_k^Y$ then y was not in $R_{\leq k-1}^Y$ and therefore no such z was in $S_{\leq k-1}^Z$); thus, z is in S_k^Z and y is rejected in the next superstep (therefore y is in R_k^Y).

Observe that this proof also shows that every iteration of the sequential algorithm is equivalent to two consecutive supersteps of our distributed algorithm although, in general, our algorithm performs more work since it has to identify both rejected (not admissible) and accepted (in the skeptical extension) arguments. By contrast, the sequential algorithm only identifies explicitly not admissible arguments to eliminate them from the admissible ones. In our algorithm, the number of supersteps to identify a not admissible argument is proportional to their distance to the leaves of the bipartite graph. Thus, the best case is a bipartite graph where every node is part of some even cycle (and so, every argument is admissible), and the skeptical extension is empty. This best case is solved by our distributed algorithm in one superstep, as no node is accepted in the first superstep, and therefore all the nodes remain in the undefined state and without sending any messages at the end of the first superstep; thus, there is no need to perform a second superstep.

To better understand the equivalence of our distributed algorithm with the sequential one, consider solving the *Author's Defeats Graph* based on the credulous semantics of Figure 5. Observe that it is bipartite, as the sets of nodes $\mathcal{Y} = \{a2\}$ and $\mathcal{Z} = \{a0, a1, a3, a4, a5\}$ define a bipartition for the graph, although our distributed algorithm does not need to know an explicit bipartition of the graph to work with it.

In the execution of our distributed algorithm, in the first superstep the undefeated nodes $\{a0, a3, a4, a5\}$ become accepted, and nodes $a0$ and $a3$ send messages to node $a2$ informing it that it has two new accepted defeaters. Then, in the second superstep, node $a2$ receives the messages from nodes $a0$ and $a3$, and therefore it changes its state to rejected and it sends a message to node $a1$, informing it that it has a new rejected defeater. In the third superstep, node $a1$ receives the message from node $a2$, and therefore it changes its state to accepted, as its only defeater (node $a2$) is rejected and it sends a message to node $a2$ informing node $a2$ that it has a new accepted defeater. However, in the next superstep, this message does not change the state of node $a2$, as it was already rejected (therefore the nodes in the even cycle reach a consistent state when one of the nodes is fixed by attacker nodes outside the cycle). Thus, after the four supersteps, all the nodes have reached a stable state, and therefore no node sends any other messages and the execution finishes.

Now consider the execution of the sequential algorithm (Algorithm 2), when determining admissible and not admissible arguments from the partition $\mathcal{Y} = \{a2\}$. In the first iteration, node $a2$ is included in the set of not admissible arguments $\mathcal{U}_1^Y$, as node $a2$ has defeaters (nodes $a0$ and $a3$) such that they do not have any defeaters in the previous set of valid defeats $\mathcal{D}_0$. Therefore, node $a2$ becomes not admissible (observe that in our distributed algorithm this happens in superstep 2), and the defeat edge $(a2, a1)$ is not included in the next set of valid defeats $\mathcal{D}_1$. Then, in the second iteration, there are no new not admissible arguments, and therefore $\mathcal{U}_2^Y$ is empty and the algorithm finishes as the set of admissible arguments becomes stable. If we now consider the execution of the algorithm but to discover admissible and not admissible arguments from the partition $\mathcal{Z} = \{a0, a1, a3, a4, a5\}$, this time with only one iteration the algorithm finds that all the arguments in $\mathcal{Z}$ are admissible, as the set $\mathcal{U}_1^Z$ is empty.

7 Argumentation graphs with no even cycles

In [20], the authors showed that an argumentation graph with no even length cycles has a unique preferred extension. Moreover, in such cases, if every node of the graph is involved in some odd length cycle, then the unique preferred extension is the empty extension. These facts are used by the authors to show that we can find the preferred extension with a polynomial time algorithm (lemma 11 in [20]).

It turns out that the algorithm defined in [20] is again doing the same work as the distributed algorithm that we presented here to compute the (skeptical) ideal extension of an acyclic or bipartite graph, and therefore our algorithm can also be used for the case of argumentation graphs without even cycles.

This is because their algorithm for the non-even cycles case, works iteratively by:

1. In the first iteration, it first identifies set S of arguments without defeaters. These arguments are obviously part of the preferred extension. Observe that this step corresponds exactly to the first superstep of our distributed algorithm, as set S is the same set of accepted arguments identified in the first superstep. Then, the algorithm eliminates S and set R of all arguments defeated by S, as they cannot be part of the preferred extension, as well as all the defeats that involve R. This set R corresponds to the set of rejected arguments of our distributed algorithm in the second superstep.
2. In any other iteration, this same process is repeated with the remaining argumentation graph. That is, it tries to find a new set S' of non-defeated arguments and a set of arguments R' defeated by S'. Observe that if there is a S' that is not defeated in the remaining argumentation graph it is because S' was defeated by arguments removed in the previous iteration. Therefore, again this corresponds to identifying new accepted arguments in a superstep of our distributed algorithm, after having identified rejected ones in the previous superstep.

This iterative process ends when no more undefeated arguments can be found, and in that case any argument in the remaining argumentation graph appears in an odd cycle, and thus there is nothing else that we have to add to the preferred extension. Therefore, the algorithm to find the unique preferred extension of an argumentation graph with no even length cycles finds the same arguments as our distributed algorithm when working on that case.

Summarizing, with our algorithm we can solve any argumentation graph with no even length cycles and also, any argumentation graph with even length cycles, whenever they are bipartite. When the argumentation graph is a non-bipartite graph with even length cycles, we can compute its solution by means of the general algorithm based on ASP which we developed in [2]. [4] However, finding the solution for a general argumentation graph is intractable for the general case [19], and therefore it is important to recognize particular cases of interest that can be solved in polynomial time.

8 Solving Author's Defeats Graphs

So far, we have identified special cases of Author's Defeats Graphs that our distributed algorithm can solve: graphs with no even cycles and bipartite graphs.

In this section, we use a test set of 28 Twitter conversations, with sizes ranging from 20 to 58 tweets [5] and with their corresponding Author's Defeats Graphs with sizes ranging from 9 to 40 authors, and we solve them with our distributed algorithm and with the algorithm of [2] that can solve any Author's Defeats Graph. Because our algorithm can solve exactly, in principle, only the tractable cases identified, we want to check to what extent we can use our algorithm to solve the real Author's Defeats Graphs that come from Twitter discussions. That is, we want to discover the frequency of the special cases that can be solved by our algorithm. In addition, we want to check whether we can use our algorithm as an *approximation* algorithm for such cases that are in principle not solvable by our algorithm (graphs with even cycles that are not bipartite).

To use it as an approximation algorithm, there is an important property of the output given by our algorithm for a general argumentation graph: it coincides with the so-called *grounded* extension [15].

[4] Since both Defeats Graphs and Author's Defeats Graph are directed graphs, it may occur that an argumentation graph does not contain cycles but it is not bipartite. In this case we can also use the distributed Algorithm 4.1 to compute its solution.

[5] All the conversations were sampled from the results obtained when searching for tweets with the hashtag `#PedroSanchez`.

Table 2 Results obtained when solving Author's Defeats Graphs with our distributed algorithm.

	$r = 0.2$			$r = 0.5$			$r = 0.8$		
	bipartite		avg	bipartite		avg	bipartite		avg
α	acyclic	cyclic	accuracy	acyclic	cyclic	accuracy	acyclic	cyclic	accuracy
0.4	13	3	100 %	18	4	100 %	23	1	100 %
0.5	13	3	100 %	20	2	100 %	23	1	100 %
0.6	20	1	100 %	22	0	100 %	24	0	100 %
0.7	22	1	100 %	23	0	100 %	26	0	100 %

The (unique) grounded extension of an argumentation graph is the minimal (with respect to set inclusion) complete extension and a complete extension is a conflict-free set of arguments that includes all the arguments that are defended by it. We say that an argument A is defended by a set of arguments S if $S \cup A$ is conflict-free and every defeater of A is in turn defeated by some argument in S. So, the grounded extension of an argumentation graph coincides with the output of our algorithm, as the grounded extension is equal to the least fixed point obtained with the following set operator:

$$G(S) = \{A \mid \text{A is defended by S}\}$$

starting with the empty set. Thus, starting from the empty set $S = \emptyset$ and iteratively applying the operator $G(S)$ until no new argument A can be added to S, the set of arguments that we obtain coincides with the ones that we accept in our algorithm.

In [14] it is shown that the grounded extension of an argumentation graph is a subset of the ideal extension. Thus, we can consider the output of our distributed algorithm, for a general argumentation graph, as an approximation of the ideal extension.

To compare the exact solutions with the ones of our algorithm, next we solve our test set of Author's Defeats Graphs with both our distributed algorithm (that solves exactly at least the non-even and bipartite cases) and the exact algorithm we developed in [2]. We generate the Author's Defeats Graph with different values of α and r for all the Twitter discussions of our test set. Then, we counted how many instances, for each combination of α and r, are bipartite (either with cycles or acyclic) [6] and we computed the average number of nodes of the ideal extension (found by the exact algorithm) that also belong to the grounded extension (found by our algorithm). We use this last value as the average accuracy of our distributed algorithm when solving a general Author's Defeats Graph.

Table 2 shows the results obtained with the different combinations of α and r tested. First, observe that for all the values of α and r tested, from the 28 discussions at least 16 discussions have bipartite Author's Defeats Graphs ($\alpha = 0.4$ and $r = 0.2$), and this quantity increases for the more restrictive values of α and r (when less defeat relationships between authors' opinions are present in the Author's Defeats Graphs, it is more likely to have acyclic graphs).

Regarding the accuracy of the grounded extension as an approximation for the ideal extension, the results are very impressive, as for all the cases, the grounded extension always coincides with the ideal extension, even for the case with less bipartite graphs. To analyze why the grounded extension coincides with the ideal extension in such non-bipartite cases (and to discard the possibility of being instances with no even cycles), we have analyzed in detail the smallest instances, and we have found no cases of non-bipartite graphs with no even cycles.

Interestingly, in such non bipartite cases, we have found that the grounded extension coincides with the ideal extension because the part of the graph with cycles is *solved* after computing the grounded extension. That is, after computing the grounded extension, the nodes in cycles become either accepted or rejected, or what remains in the part of the graph not included in the accepted and rejected nodes does not add nothing else to the ideal extension. Consider the two following non-bipartite Author's Defeats Graphs from our test set.

Figure 6 shows the Author's Defeats Graph for the first case, but only the part that involves cycles (apart of two isolated nodes). The nodes colored in blue are the authors in the solution and the nodes colored in black are the rejected ones. This case is solved with our algorithm because after eliminating the accepted and rejected ones by the grounded extension, the remaining part is a simple even cycle between two nodes (authors $a2$ and $a16$), but this bipartite subgraph has an empty ideal extension, so the ideal extension for the whole graph coincides with the grounded extension.

Figure 7 shows the Author's Defeats Graph for the second case (again only the part with cycles). Observe that it contains several even cycles, but it is not bipartite. This time, even if the part with cycles is bigger than before, due to the *grounded* authors $a0$, $a2$, and $a16$ (authors without attackers that are accepted), authors $a6$, $a12$, and $a22$ become rejected in the grounded extension. Then, authors $a7$ and $a20$ become accepted because they are attacked only by rejected authors. So, all the nodes are either accepted or rejected after computing the grounded extension, so the grounded extension coincides again with the ideal extension.

[6] We do not have available a polynomial time algorithm for checking whether a directed graph has even cycles, so we have not checked if there were any no even cycle graphs.

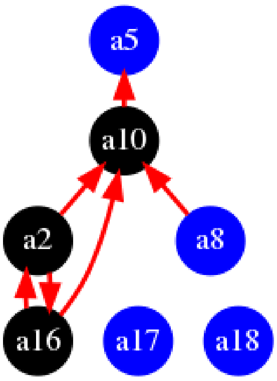

Fig. 6 Part of the Author's Defeats Graph solution for conversation with root Tweet ID 792044727835193344 based on the intermediate semantics using $\alpha = 0.5$ and $r = 0.2$. Author's Defeats Graph size: 22 nodes and 8 edges.

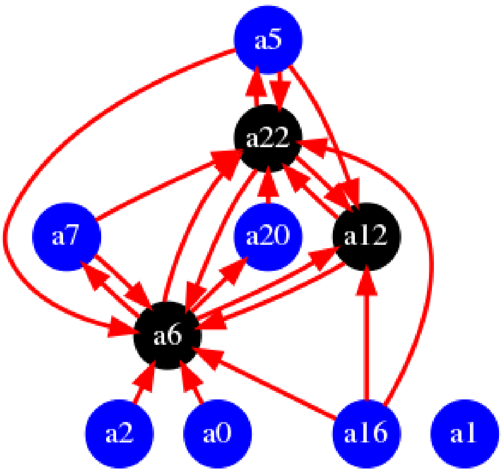

Fig. 7 Part of the Author's Defeats Graph solution for conversation with root Tweet ID 792044003621494784 based on the intermediate semantics using $\alpha = 0.5$ and $r = 0.2$. Author's Defeats Graph size: 23 nodes and 21 edges.

Actually, these two cases can be seen as particular cases for the class of extended tractable cases analyzed in [22]. In that work, the authors analyze tractable argumentation graphs that arise when considering the *reduced* argumentation graph. The reduced argumentation graph is obtained after computing its grounded extension and then subtracting its *range* (accepted and rejected nodes by the grounded extension) from the graph. The authors investigate cases where computing the complete extensions of the reduced argumentation graph can be tractable. In our case, it suffices to satisfy that computing the ideal extension of the reduced argumentation graph is tractable. The two cases we have analyzed above are examples of that particular case, because after eliminating the range of the grounded extension we have obtained a bipartite graph with no leafs (first example) or an empty graph (second example), both trivial cases with empty ideal extension.

Of course, for bigger Author's Defeats Graphs it will be more likely that the reduced argumentation graphs have a non-empty and non-tractable ideal extension. However, with the (exact) general sequential algorithm we cannot expect to solve bigger instances with a reasonable time and thus, it is difficult to study when the grounded extension begins to be a proper subset of the ideal extension.

Another interesting feature we observe in the two non-bipartite graphs of Figure 6 and Figure 7 is that many of the cycles are even. This can be due to the fact that in regular conversations criticism between authors is more likely to be present between two opposite groups, with only a few exceptions. In addition, in the second example (Figure 7), which is the only one with odd cycles, we observe that they all involve author $a22$ and most of them also author $a6$. One possible interpretation of this situation is that both authors have polarized the discussion and the rest of the authors criticize one of them and even both. However, observe that we also have an intermediate situation, like the one we observe between authors $a2$, $a10$ and $a16$ in the first example (Figure 6). In this case, author $a16$ criticizes author $a10$ and author $a2$ criticizes both of them. This situation does not generate an odd cycle, but indicates that opinions should be classified in at least three groups, and not just in two. We leave as future work to investigate more systematically these features that appear in Author's Defeats Graphs of Twitter discussions.

9 Conclusions and future work

In this paper, we introduce a distributed valued argumentation approach to ascertain the set of globally accepted tweets of a Twitter discussion. We model discussions with a weighted argumentation graph, where each node denotes a tweet, each edge denotes a probability value of criticism relation between a pair of tweets of the discussion and each node is attached to a weight, that denotes the social relevance of the corresponding tweet in the discussion and it is computed from the tweet's attributes, such as the number of followers of the author, the number of retweets and the number of favorites.

The set of globally accepted tweets is defined following a skeptical approach based on the ideal semantics of a valued argumentation framework. The ideal semantics for valued argumentation guarantees that the set of tweets in the solution is the maximal set of tweets that satisfies that it is consistent and admissible, in the sense that there are no defeaters among them, and that all of the tweets outside the solution are defeated by a tweet within the solution.

The distributed implementation of the valued argumentation reasoning model consists of a pre-processing step and a distributed computation algorithm for an argumentation graph. The pre-processing step prunes all attacks that are not effective based on the probability values of edges and the weight of nodes. The distributed computation algorithm is based on the computation model of Pregel which is appropriate for our problem, because the input for a Pregel algorithm is a directed graph, where the nodes can be in different states, and the goal is to compute the state of each node based on the state of the nodes' neighbors.

In our system each node (tweet) can be in a single state, accepted or rejected, and the algorithm allows us to solve acyclic discussions of big size with an efficient polynomial time distributed algorithm, where the state of each node depends only on the state of its defeating nodes. As far as we know, the system is the first distributed implementation of an argumentative reasoning algorithm for social network analysis. Our experimental results show that we can solve real Twitter discussions of significant size with a scaling cost in time that seems to be at most linear, a result that it is consistent with the fact that the number of super steps of our distributed algorithm is, in the worst case, equal to maximum distance of any node to the leafs nodes.

In this work we also analyze author-centered Twitter discussions by means of the distributed implementation of the reasoning model. To this end we define an argumentation graph to represent Twitter discussions based on an authors' opinions model, the Author's Defeats Graph, in which each node denotes an author's opinion (an argument represented by the whole set of author's tweets) and each edge denotes an effective attack (defeat) between two authors' opinions.

From the computational point of view, the main difference between the tweet-based model and the author-centered model, lies in the structure of the underlying graph. For the case of the tweet-based model, it is always a directed acyclic graph; however, for the case of the author-centered model, the graph usually contains some cycles.

Regarding the solving of argumentation graphs with cycles, we show that the distributed algorithm can also solve bipartite graphs or graphs with no even cycles. We also show that for a general argumentation graph, the output provided by our algorithm coincides with the grounded extension, that it is always a subset of the ideal extension (the solution we want to find). Moreover, our experimental results, with a test set of Twitter discussions, show that the typical Author's Defeats Graphs obtained from Twitter discussions have always solutions that coincide with the grounded extension, at least for the range of Twitter discussions sizes we have been able to analyze with both the general sequential algorithm [2], that solves any argumentation graph, and our distributed algorithm.

As future work, we plan to extend both the representation and reasoning models to consider support relationships between tweets and also to explore more credulous defeats semantics for valued argumentation frameworks. In this sense, a possible starting point could be the work of [13], where the authors propose an argumentation system where arguments include a set of labels that allow a finer-grained analysis of argument impact. It would be also interesting to consider acceptance semantics where arguments are accepted with some degree, that depends on some way on the *strength* of its attacking arguments, instead of simply rejecting all the attacks that are not defeats in the VAF associated with the discussion. Also, regarding the issue of the assumed consistency of the set of tweets that form an author's opinion, we plan to analyze to what extent this is true for a given author, and to consider ways to penalize the author's social relevance based on some measure of the author's consistency. Finally, we plan also to study additional tractable cases that could be solved by our algorithm (or by simple modifications of it) by looking at more detail at the results presented in [22] about tractable argumentation problems based on the idea of studying the complexity of solving the reduced argumentation graph, as we have introduced in the experimental results.

Compliance with Ethical Standards

Funding: This work was partially funded by Spanish Project TIN2015-71799-C2-2-P (MINECO/FEDER), by the European Union's Horizon 2020 Research and Innovation Programme under Grant Agreement 723596 and Grant Agreement 768824, and by 2017 SGR 1537. This research article has received a grant for its linguistic revision from the Language Institute of the University of Lleida (2018 call).

Conflict of Interest: Author J. Cemeli has a contract with Company Starloop Studios. Authors T. Alsinet, J. Argelich, and R. Béjar declare that they have no conflict of interest.

Ethical approval: This article does not contain any studies with human participants or animals performed by any of the authors.

References

1. Alsinet, T., Argelich, J., Béjar, R., Esteva, F., Godo, L.: A probabilistic author-centered model for Twitter discussions. In: IJCAI Workshop on Logical Foundations for Uncertainty and Machine Learning, pp. 3–8 (2017)
2. Alsinet, T., Argelich, J., Béjar, R., Fernández, C., Mateu, C., Planes, J.: Weighted argumentation for analysis of discussions in Twitter. Int. J. Approx. Reasoning **85**, 21–35 (2017)
3. Alsinet, T., Argelich, J., Béjar, R., Fernández, C., Mateu, C., Planes, J.: An argumentative approach for discovering relevant opinions in Twitter with probabilistic valued relationships. Pattern Recognition Letters **105**, 191–199 (2018). DOI 10.1016/j.patrec.2017.07.004
4. Alsinet, T., Argelich, J., Béjar, R., Planes, J., Cemeli, J., Sanahuja, C.: A distributed approach for the analysis of discussions in twitter. In: Proceedings of the 3rd International Workshop on Social

Influence Analysis co-located with 26th International Joint Conference on Artificial Intelligence (IJCAI 2017), Melbourne, Australia, August 19, 2017., pp. 45–56 (2017)

5. Baroni, P., Giacomin, M.: A distributed self-stabilizing algorithm for argumentation. In: Proceedings of the 15th International Parallel & Distributed Processing Symposium (IPDPS-01), p. 79. IEEE Computer Society (2001)
6. Baroni, P., Giacomin, M.: Argumentation through a distributed self-stabilizing approach. J. Exp. Theor. Artif. Intell. **14**(4), 273–301 (2002)
7. Bench-Capon, T.J.M.: Value-based argumentation frameworks. In: Proceedings of 9th International Workshop on Non-Monotonic Reasoning, NMR 2002, pp. 443–454 (2002)
8. Bench-Capon, T.J.M.: Persuasion in practical argument using value-based argumentation frameworks. Journal of Logic and Computation **13**(3), 429–448 (2003)
9. Bench-Capon, T.J.M., Dunne, P.E.: Argumentation in artificial intelligence. Artif. Intell. **171**(10-15), 619–641 (2007)
10. Besnard, P., Hunter, A.: A logic-based theory of deductive arguments. Artif. Intell. **128**(1-2), 203–235 (2001)
11. Bild, D.R., Liu, Y., Dick, R.P., Mao, Z.M., Wallach, D.S.: Aggregate characterization of user behavior in Twitter and analysis of the retweet graph. ACM Trans. Internet Techn. **15**(1), 4:1–4:24 (2015)
12. Bosc, T., Cabrio, E., Villata, S.: Tweeties squabbling: Positive and negative results in applying argument mining on social media. In: Computational Models of Argument - Proceedings of COMMA 2016., pp. 21–32 (2016)
13. Budán, M.C.D., Simari, G.I., Simari, G.R.: Using argument features to improve the argumentation process. In: Computational Models of Argument - Proceedings of COMMA 2016, Potsdam, Germany, 12-16 September, 2016., pp. 151–158 (2016)
14. Caminada, M.: Comparing two unique extension semantics for formal argumentation: Ideal and eager. In: Proceedings of 19th Belgian-Dutch Conference on Artificial Intelligence (BNAIC 2007), pp. 81–87 (2007)
15. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence **77**(2), 321 – 357 (1995)
16. Dung, P.M., Mancarella, P., Toni, F.: Computing ideal sceptical argumentation. Artif. Intell. **171**(10-15), 642–674 (2007)
17. Dunne, P.E.: Computational properties of argument systems satisfying graph-theoretic constraints. Artif. Intell. **171**(10-15), 701–729 (2007)
18. Dunne, P.E.: The computational complexity of ideal semantics I: abstract argumentation frameworks. In: Proceedings of Computational Models of Argument, COMMA 2008, Toulouse, France, pp. 147–158 (2008)
19. Dunne, P.E.: The computational complexity of ideal semantics. Artif. Intell. **173**(18), 1559–1591 (2009)
20. Dunne, P.E., Bench-Capon, T.: Complexity and combinatorial properties of argument systems. Tech. rep., University of Liverpool (2001). Available as `http://www.csc.liv.ac.uk/~ped/papers/csd_rep_argument.ps`
21. Dusmanu, M., Cabrio, E., Villata, S.: Argument mining on twitter: Arguments, facts and sources. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017., pp. 2317–2322 (2017)
22. Dvorák, W., Ordyniak, S., Szeider, S.: Augmenting tractable fragments of abstract argumentation. Artif. Intell. **186**, 157–173 (2012). DOI 10.1016/j.artint.2012.03.002
23. Egly, U., Gaggl, S.A., Woltran, S.: Aspartix: Implementing argumentation frameworks using answer-set programming. In: Proceedings of the 24th International Conference on Logic Programming, ICLP 2008, pp. 734–738 (2008)
24. Fazzinga, B., Flesca, S., Parisi, F.: On the complexity of probabilistic abstract argumentation. In: IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, pp. 898–904. IJCAI/AAAI (2013)
25. Grosse, K., Chesñevar, C.I., Maguitman, A.G.: An argument-based approach to mining opinions from Twitter. In: Proceedings of the First International Conference on Agreement Technologies, AT 2012, *CEUR Workshop Proceedings*, vol. 918, pp. 408–422. CEUR-WS.org (2012)
26. Grosse, K., González, M.P., Chesñevar, C.I., Maguitman, A.G.: Integrating argumentation and sentiment analysis for mining opinions from Twitter. AI Commun. **28**(3), 387–401 (2015)
27. Hunter, A.: Some foundations for probabilistic abstract argumentation. In: Computational Models of Argument - Proceedings of COMMA 2012, *Frontiers in Artificial Intelligence and Applications*, vol. 245, pp. 117–128. IOS Press (2012)
28. Hunter, A.: Probabilistic qualification of attack in abstract argumentation. Int. J. Approx. Reasoning **55**(2), 607–638 (2014)
29. Li, H., Oren, N., Norman, T.J.: Probabilistic argumentation frameworks. In: Theory and Applications of Formal Argumentation - First International Workshop, TAFA 2011, *Lecture Notes in Computer Science*, vol. 7132, pp. 1–16. Springer (2011)
30. Malewicz, G., Austern, M.H., Bik, A.J.C., Dehnert, J.C., Horn, I., Leiser, N., Czajkowski, G.: Pregel: a system for large-scale graph processing. In: Proceedings of the ACM SIGMOD International Conference on Management of Data, SIGMOD 2010, pp. 135–146 (2010)
31. Rahwan, I., Simari, G.R.: Argumentation in Artificial Intelligence, 1st edn. Springer Publishing Company (2009)
32. Thimm, M.: A probabilistic semantics for abstract argumentation. In: ECAI 2012 - 20th European Conference on Artificial Intelligence, *Frontiers in Artificial Intelligence and Applications*, vol. 242, pp. 750–755. IOS Press (2012)
33. Valiant, L.G.: A bridging model for multi-core computing. J. Comput. Syst. Sci. **77**(1), 154–166 (2011)