A Tool For Ranking Arguments Through Voting-Games Power Indexes

Stefano Bistarelli[1], Francesco Faloci[1], Francesco Santini[1], and Carlo Taticchi[2]

[1] Dipartimento di Matematica e Informatica, Università degli studi di Perugia, Italia
[2] Gran Sasso Science Institute, L'Aquila, Italy

Abstract. Abstract Argumentation Frameworks allow to represent sets of arguments, together with possible relations among them, in form of oriented graphs. This paper gives a short overview of a plug-in function developed for ConArg, a solver of Abstract Argumentation related problems. The web-based tool we present computes a ranking of arguments by applying different voting games power indexes, where the coalitions of individuals are defined by the extensions satisfying Dung's semantics. At this stage of development, the tool can make use of both the Shapley Value and the Banzhaf Index.

1 Introduction

ConArg is a suite of tools that was started to be developed with the purpose to facilitate research in the field of Argumentation in Artificial Intelligence [3], a discipline that copes with uncertainty and defeasible reasoning. In Abstract Argumentation, arguments have no internal structure and the attack relation is not defined; it provides means by which it is possible to distinguish acceptable and not acceptable arguments at an abstract , as its name suggests. In order for a set of arguments to be accepted, it has to be justified according to some criteria, that are called semantics. The sets of collectively-acceptable arguments according to a certain semantics are referred to as "extensions".

Recent works (as the ones presented in [5,6]) have been carried out with the help of ConArg[3]. The project involves a series of components that address different aspects of argumentation, building on a constraint-based solver for argumentation problems [7,9]. The tool has already been extended with two main additional features that allow for handling weighted [6,8] and probabilistic [5] argumentation. While the former relies on algebraic structures (c-semirings) for dealing with weights, the latter makes use of a probabilistic logic programming language.

In this work, we present a new component of the ConArg suite, which integrates the possibility of managing *ranking semantics*. In classical argumentation, arguments can be either accepted or rejected according to their justification status, but no further distinction can be done beyond this division into these two

[3] ConArg Website: `http://www.dmi.unipg.it/conarg/`.

categories.[4] On the other hand, ranking semantics permit to assign an individual score to each argument so that an overall ranking of all arguments can be established by sorting the obtained set of scores. Carrying on the work in [4], we here propose an implementation of a ranking function based on the Shapley Value [13], a very well known concept in cooperative game theory, which we use to distribute the scores among the arguments: the more an argument contributes to the acceptability of an extension, the higher its score. In addition, we also take into account a different valuation scheme, the Banzhaf Index [2], and we implement it in order to study the differences with the results obtained through the Shapley Value. Given an argumentation framework, the tool computes the score of every argument over both the ranking schemes introduced above, and its output is a ranking of the arguments with respect to a given semantics.

As previously introduced, this line of work commenced in [4] with the first theoretical results. This paper is instead dedicated to the description of the underlying tool, which also computes the Banzhaf Index, differently from [4]. In Section 2 we introduce the background information about Abstract Argumentation and Power Indexes. Section 3 describes the tool and its integration in ConArg, while Section 4 presents two examples of application on abstract frameworks. Finally, Section 5 wraps up the paper with final conclusions and ideas about future work.

2 Preliminaries

We introduce our tool by first reporting the necessary background notions on labelling and ranking semantics in Abstract Argumentation, and successively we introduce Power Indexes in cooperative game theory.

2.1 Argumentation

This work takes advantage on notions coming from two different fields: argumentation and cooperative games. In the following, we provide a brief introduction only to the concepts which are most relevant to us. An *Abstract Argumentation Framework* [12] (AF in short) consists of a pair $\langle A, R\rangle$ where A is a set of arguments and $R \subseteq A \times A$ expresses the relations between pairs of arguments. Such relations, which we call "attacks", are interpreted as conflict conditions that allow for determining the arguments in A are acceptable together (i.e., collectively).

An argumentation *semantics* is a criterion that establishes which are the acceptable arguments by considering the relations among them. Two leading characterisations can be found in the literature, namely *extension-based* [12] and *labelling-based* [11] semantics. While providing the same outcome in terms of accepted arguments, labelling-based semantics can be used to differentiate

[4] More than just two categories have been proposed in the literature, but still from a qualitative point of view.

between three levels of acceptability, by assigning labels to arguments according to the conditions stated in Definition 1.

Definition 1 (Reinstatement Labelling). *Let $F = \langle A, R \rangle$ be an AF and $\mathbb{L} = \{in, out, undec\}$. A labelling of F is a total function $L : A \to \mathbb{L}$. We define $in(L) = \{a \in A \mid L(a) = in\}$, $out(L) = \{a \in A \mid L(a) = out\}$ and $undec(L) = \{a \in A \mid L(a) = undec\}$. We say that L is a reinstatement labelling if and only if it satisfies the following conditions:*

- *$\forall a, b \in A$, if $a \in in(L)$ and $(b, a) \in R$ then $b \in out(L)$;*
- *$\forall a \in A$, if $a \in out(L)$ then $\exists b \in A$ such that $b \in in(L)$ and $(b, a) \in R$.*

The labelling obtained through the function in Definition 1 can be then analysed in terms of Dung's semantics [12].

Definition 2 (Labelling-based semantics). *A labelling-based semantics σ associates with an AF* F *a subset of all the possible labellings for F, denoted as $L_\sigma(F)$. Let L be a labelling of $F = \langle A, R \rangle$, then L is*

- ***conflict-free*** *if and only if for each $a \in A$ it holds that if a is labelled* in *then it does not have an attacker that is labelled* in*, and if a is labelled* out *then it has at least one attacker that is labelled* in*;*
- ***admissible*** *if and only if the attackers of each* in *argument are labelled* out*, and each* out *argument has at least one attacker that is* in*;*
- ***complete*** *if and only if for each $a \in A$, a is labelled* in *if and only if all its attackers are labelled* out*, and a is* out *if and only if it has at least one attacker that is labelled* in*;*
- ***preferred/grounded*** *if L is a complete labelling where the set of arguments labelled* in *is maximal/minimal (with respect to set inclusion) among all complete labellings;*
- ***stable*** *if and only if it is a complete labelling and $undec(L) = \emptyset$.*

The accepted arguments, with respect to a certain semantics σ, are those labelled *in* by σ. In order to further discriminate among arguments, *ranking-based* semantics [1] can be utilised for sorting the arguments from the most to the least preferred.

Definition 3 (Ranking-based semantics). *A ranking-based semantics associates with any $F = \langle A, R \rangle$ a ranking $\succcurlyeq_F$ on A, where $\succcurlyeq_F$ is a pre-order (a reflexive and transitive relation) on A. $a \succcurlyeq_F b$ means that a is at least as acceptable as b ($a \simeq b$ is a shortcut for $a \succcurlyeq_F b$ and $b \succcurlyeq_F a$, and $a \succ_F b$ is a shortcut for $a \succcurlyeq_F b$ and $b \not\succcurlyeq_F b$).*

2.2 Power Indexes

In game theory, cooperative games are games where groups of players (or agents) are competing to maximise their goal, through one or more specific rules. *Voting games* are a particular category of cooperative games in which the profit of coalitions is determined by the contribution of each individual player. In order to identify the "value" brought from a single player to a coalition, *power indexes* are used to define a preference relation between different agents, computed on all the possible coalitions. The most used power indexes for voting games are the Shapley-Shubic Value [13,14] (Shapley Value in the following) and the Banzhaf-Coleman Power Index [2] (Banzhaf Index in the following). Given a set N of players, both indexes rely on a characteristic function $v : 2^N \rightarrow \mathbb{R}$ that associates each coalition $S \subseteq N$ with a real number in such a way that $v(S)$ describes the total gain that agents in S can obtain by cooperating with each other. The Shapley Value of a player $i \in N$ is computed as follows.

$$\Phi_i(v) = \frac{1}{|N|!} \sum_{S \subseteq N \setminus \{i\}} |S|! \ (|N| - |S| - 1)! \ (v(S \cup \{i\}) - v(S)) \tag{1}$$

The formula considers a random ordering of the agents, picked uniformly from the set of all $|N|!$ possible orderings, and exploit the difference of gain between S and $S \cup \{i\}$ for estimating the expected marginal contribution of the player i. The value $|S|! \ (|N| - |S| - 1)!$ expresses the probability that all the agents in S come before i in a random ordering.

The second fair division scheme we use is the Banzhaf Index, which evaluates each player by using the notion of *critical voter*: given a coalition $S \subseteq N \setminus \{i\}$, a critical voter for S is a player i such that $S \cup \{i\}$ is a winning coalition, while S alone is not. In other words, i is a critical voter if it can change the outcome of the coalition it joins in.

$$\beta_i(v) = \frac{1}{2^{|N|-1}} \sum_{S \subseteq N \setminus \{i\}} (v(S \cup \{i\}) - v(S)) \tag{2}$$

On the following section we show how the chosen power indexes are used to evaluate arguments and how the tool computes the corresponding ranking.

3 An Implementation of Ranking Semantics using Power Index

Ranking semantics allow to establish an ordering over arguments in a framework, in a way to discriminate between multiple degrees of acceptance and to identify which arguments are the most preferred ones. Different ranking-based semantics, such as those summarised in [10], use different criteria for evaluating the arguments (for instance, relying on the number and the "strength" of attackers). The approach we propose, instead, takes into account the contribution that an argument brings to the sets of extensions. In particular, arguments that contribute more in forming an extension (e.g., because they defend all the other arguments in that extension) are ranked higher than the others. Each argument is considered a player and the extensions are the coalitions that players want to form. In this way we can deal with the problem of forming the extensions as a cooperative game. Moreover, we can compute the power index of all the arguments in the framework. Note that this kind of ranking semantics is parametric w.r.t. a chosen Dung's semantics, that is the ranking one obtains is different depending on the sets of accepted arguments.

3.1 Computation of Argument's Indexes

The procedure for computing the value of an argument i through the voting power indexes Shapley Value (Equation 1) and Banzhaf Index (Equation 2) using the formal formulas. Each index is a mean of the gains of the argument over the coalitions $S \cup \{i\}$: for the sake of modularity, the shared part of the formula (namely, $[v(S \cup \{i\}) - v(S)]$) is computed once for both schemes. We use $v : 2^N \rightarrow \{0, 1\}$ as the characteristic function for both the indexes we consider: the function outputs 1 if a coalition is an extension according to a given semantics (Definition 2), 0 otherwise. Finally, in terms of computational time, the Shapley Value can be computed in $O(|N|^2)$, while the Banzhaf Index in $O(|N|)$. In order to ease the computation, instead of computing all the possible subsets in the set of arguments, the script only selects the sets of the extensions, since they represent the "winning coalitions" within all the possible subsets. This means that all the subsets S required by the formula consist in the set of extensions received as a parameter.

The script computes the value of the shared part of the formula for each argument, assigning a value of -1 when an argument is not part of an extension and $S \cup \{i\}$ is not included in the semantics; 1 if the argument is part of the extension, and if without its presence the corresponding set is not included in the semantics; 0 otherwise. This computation is repeated on all the given extensions.

The ranking is an arithmetic decreasing ordering of all the values computed for the selected index. In order to ensure a better ranking definition, in addition to the set of *in* arguments (i.e., the extension), also *out* ones are taken into account, that is the power indexes are calculated also on the set of *out*-labelled arguments. This second ranking is used for breaking ties when two arguments receive the same score by the evaluation done with respect to *in* arguments. The script returns first both the ranking of *in* and *out* arguments. When the values of two arguments are equal in the *in* ranking, the script checks the corresponding couple of values of the same arguments in the *out* ranking. The lowest value between them represents the preferred argument of the pair in the final ranking: if also the *out* ranking returns two equal values, then the tie cannot be resolved.

3.2 Tool Description

The visual tool we present takes advantage of all the features offered by ConArg in order to select and to process a particular framework. This tool is composed by a *javascript* (*JS*) and a *PHP* class. The former contains the functions for both the user and the ConArg interface, while in the PHP class it is possible to find all the power index calculation and output formatting. Once a framework

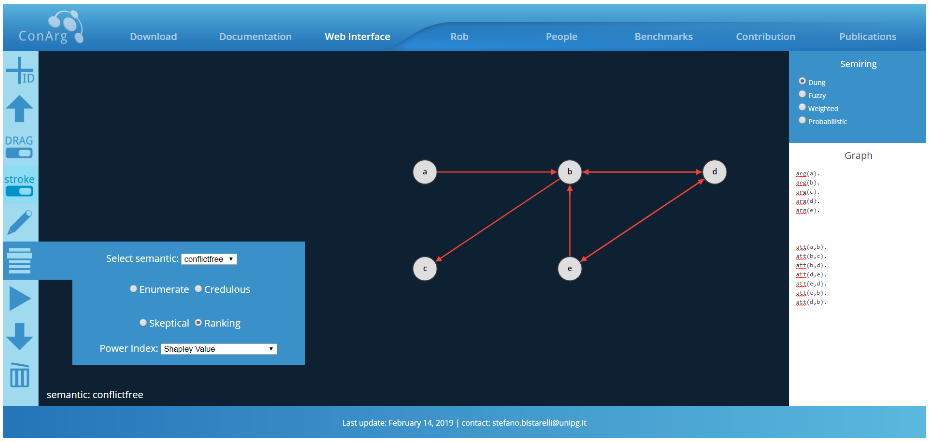

Fig. 1. An example of the tool execution in ConArg. The "Output" frame shows the arguments ordered according to the ranking obtained through the Shapley Value, from the most to the least preferred. Each argument is paired with its corresponding Shapley Value.

is created (or imported) by using ConArg functionalities, the "Ranked" option must be selected from the edit menu, as shown in Figure 1.

In this menu, the tool places different kinds of options to compute semantics. The specific semantics can be chosen in the selection pad above the computation options provided by the menu. The "Enumerate" choice generates the selected semantics. The "Credulous" option identifies if there is an extension in the selected semantics which contains a given argument: the id of an argument is required as a further parameter in the options menu. The "Sceptical" option identifies if all the extensions of a semantics contain the given argument: as for the "Credulous" option, the id of the argument is requested as a further parameter. The last choice is the "Rank" option, which can be used to compute a possible ranking of the framework with the specific semantic. This option asks the user to select the Power Index that is used to produce the ranking. Even if the rankings for both the indexes are provided by the PHP class, the tool shows the selected one in the options menu, together with the computed values.

The tool first computes the set of extensions for the selected semantics, and only at a second stage, it calls the script function that computes the final ranking. All the script functions can be reached by a *post PHP rest call*, which asks for four different parameters: the set of extensions satisfying the requested semantics formatted as ConArg output string (e.g., $\{\{a\},\{a,b\},\{c,d\}\}$), the number of arguments, the attacks between the arguments on the framework and an array of option values (that we plan to use in future implementations). All this information is retrieved from the ConArg toolkit by the JS class. The output of the script is a json file that contains the extensions formatted as ConArg output string, and two lists of arguments ordered according to the Shapley Value and

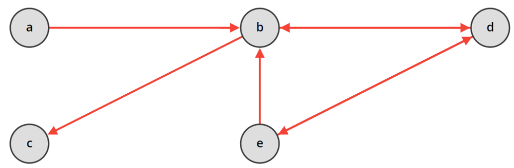

Fig. 2. Example of an AF F. The sets of extensions for the complete, preferred and stable semantics are: $COM = \{\{a, d\}, \{a, c, e\}, \{a, c, d\}\}$, $PRE = \{\{a, c, e\}, \{a, c, d\}\}$, and $STB = \{\{a, c, e\}, \{a, c, d\}\}$, respectively.

the Banzhaf Index, respectively. The JS class shows the final ranking on the Output field. Here the ranking is defined according to the power index specified by the user. The value obtained for each argument is approximated to the nearest fifth decimal digit.

4 Examples

In this section, we present an example based on the framework shown in Figure 2, that is representative of some different features. The considered AF has an initiator (i.e., the argument a, which is not attacked by any other argument), a symmetric attack (between b/d, and d/e) and a cycle (b-d-e).

We use the tool described in Section 3 for computing the ranking-based semantics of the framework. Table 1 reports the final ranking for the AF in Figure 2, obtained by using the Shapley Value. In Table 2, instead, we consider the Banzhaf Index. In both tables, power indexes are computed for the conflict-free, admissible, complete, preferred and stable semantics, alternating in each row the values of the indexes with respect to the sets of *in* and *out* arguments.

	a	**b**	**c**	**d**	**e**	Semantics	Ranking
IN_{CF}	−0.050	−0.46667	−0.050	−0.21667	−0.21667	SV-CF	$a \succ c \succ e \succ d \succ b$
OUT_{CF}	−0.350	0.06667	−0.26667	−0.18333	−0.26667		
IN_{ADM}	0.050	−0.61667	−0.20	−0.11667	−0.11667	SV-ADM	$a \succ d \simeq e \succ c \succ b$
OUT_{ADM}	−0.31667	0.10	−0.31667	−0.234	−0.234		
IN_{COM}	0.11667	−0.134	0.11667	−0.050	−0.050	SV-COM	$a \simeq c \succ d \simeq e \succ b$
OUT_{COM}	−0.11667	0.3	−0.11667	−0.034	−0.034		
IN_{PRE}	0.06667	−0.1	0.06667	−0.01667	−0.01667	SV-PRE	$a \simeq c \succ d \simeq e \succ b$
OUT_{PRE}	−0.06667	0.1	−0.06667	0.01667	0.01667		
IN_{STB}	0.06667	−0.1	0.06667	−0.01667	−0.01667	SV-STB	$a \simeq c \succ d \simeq e \succ b$
OUT_{STB}	−0.06667	0.1	−0.06667	0.01667	0.01667		

Table 1. Shapley Value for the arguments of the AF in Figure 2.

	a	b	c	d	e	Semantics	Ranking
IN_{CF}	−0.06250	−0.68750	−0.06250	−0.31250	−0.31250	BPI-CF	$a \succ c \simeq e \succ d \succ b$
OUT_{CF}	−0.31250	0.06250	−0.18750	−0.06250	−0.18750		
IN_{ADM}	0.06250	−0.68750	−0.06250	−0.18750	−0.18750	BPI-ADM	$a \succ c \succ d \simeq e \succ b$
OUT_{ADM}	−0.250	0.1250	−0.250	−0.1250	−0.1250		
IN_{COM}	0.18750	−0.18750	0.18750	−0.06250	−0.06250	BPI-COM	$a \simeq c \succ d \simeq e \succ b$
OUT_{COM}	−0.18750	0.18750	−0.18750	−0.06250	−0.06250		
IN_{PRE}	0.125	−0.1250	0.1250	0.0	0.0	BPI-PRE	$a \simeq c \succ d \simeq e \succ b$
OUT_{PRE}	−0.1250	0.1250	−0.1250	0.0	0.0		
IN_{STB}	0.125	−0.1250	0.1250	0.0	0.0	BPI-STB	$a \simeq c \succ d \simeq e \succ b$
OUT_{STB}	−0.1250	0.1250	−0.1250	0.0	0.0		

Table 2. Banzhaf Index for the arguments of the AF in Figure 2.

Except for conflict-free and admissible sets, the obtained rankings are the same for the two indexes. In both cases, argument a is always ranked at the first position, correctly following the principle that unattacked arguments should be ranked before than attacked ones [10].

5 Conclusion

In this paper, we have described a Web-based tool to compute voting games power indexes over the arguments of an AF. What we obtain is a ranking-based semantics for each index and, within the same index, for each Dung's semantics that defines our set of arguments.

In the future, we plan to implement other indexes in the tool, or combinations of them: our aim is to understand which ranking properties (or families of them, i.e., local or global) listed in [10] such indexes can successfully capture. With the comparison of different indexes, we would like to define if there is a link between ties on rankings and the possible resolution of ambiguities. We would like to design indexes or procedures on top of them, which are able to capture global properties instead of local ones. Local properties [10] are local to an argument: they can be checked by inspecting attacked or attacking arguments in the immediate neighbourhood of an argument. Global properties [10] derive instead from the whole framework structure: they depend, for instance, by full attacking or defending paths.

References

1. Amgoud, L., Ben-Naim, J.: Ranking-based semantics for argumentation frameworks. In: Proceedings of SUM 2013. Lecture Notes in Computer Science, LNCS, vol. 8078, pp. 134–147. Springer (2013)
2. Banzhaf, J.F.: Weighted voting doesn't work: A mathematical analysis. Rutgers Law Review 19(2), 317–343 (1965)

3. Baroni, P., Gabbay, D.M., Giacomin, M., van der Torre, L.: Handbook of formal argumentation. College Publications (2018)
4. Bistarelli, S., Giuliodori, P., Santini, F., Taticchi, C.: A cooperative-game approach to share acceptability and rank arguments. In: Proceedings of the 2nd Workshop on Advances In Argumentation In Artificial Intelligence, co-located with XVII International Conference of the Italian Association for Artificial Intelligence, AI^3@AI*IA. CEUR Workshop Proceedings, vol. 2296, pp. 86–90. CEUR-WS.org (2018), http://ceur-ws.org/Vol-2296/AI3-2018_paper_8.pdf
5. Bistarelli, S., Mantadelis, T., Santini, F., Taticchi, C.: Probabilistic argumentation frameworks with metaproblog and conarg. In: IEEE 30th International Conference on Tools with Artificial Intelligence, ICTAI 2018, 5-7 November 2018, Volos, Greece. pp. 675–679 (2018)
6. Bistarelli, S., Rossi, F., Santini, F.: Conarg: A tool for classical and weighted argumentation. In: Computational Models of Argument - Proceedings of COMMA 2016, Potsdam, Germany, 12-16 September, 2016. pp. 463–464 (2016)
7. Bistarelli, S., Rossi, F., Santini, F.: A conarg-based library for abstract argumentation. In: 29th IEEE International Conference on Tools with Artificial Intelligence, ICTAI 2017, Boston, MA, USA, November 6-8, 2017. pp. 374–381. IEEE Computer Society (2017), https://doi.org/10.1109/ICTAI.2017.00065
8. Bistarelli, S., Rossi, F., Santini, F.: A novel weighted defence and its relaxation in abstract argumentation. Int. J. Approx. Reasoning 92, 66–86 (2018), https://doi.org/10.1016/j.ijar.2017.10.006
9. Bistarelli, S., Santini, F.: Modeling and solving afs with a constraint-based tool: Conarg. In: Theory and Applications of Formal Argumentation - First International Workshop, TAFA 2011. Barcelona, Spain, July 16-17, 2011, Revised Selected Papers. Lecture Notes in Computer Science, vol. 7132, pp. 99–116. Springer (2011)
10. Bonzon, E., Delobelle, J., Konieczny, S., Maudet, N.: A comparative study of ranking-based semantics for abstract argumentation. In: Schuurmans, D., Wellman, M.P. (eds.) Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence, February 12-17, 2016, Phoenix, Arizona, USA. pp. 914–920. AAAI Press (2016), http://www.aaai.org/ocs/index.php/AAAI/AAAI16/paper/view/12465
11. Caminada, M.: On the issue of reinstatement in argumentation. In: Fisher, M., van der Hoek, W., Konev, B., Lisitsa, A. (eds.) Logics in Artificial Intelligence, 10th European Conference, JELIA 2006, Liverpool, UK, September 13-15, 2006, Proceedings. Lecture Notes in Computer Science, vol. 4160, pp. 111–123. Springer (2006), https://doi.org/10.1007/11853886_11
12. Dung, P.M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artif. Intell. 77(2), 321–358 (1995), https://doi.org/10.1016/0004-3702(94)00041-X
13. Shapley, L.S.: Contributions to the Theory of Games (AM-28), Volume II. Princeton University Press (1953)
14. Winter, E.: The shapley value. In: Aumann, R., Hart, S. (eds.) Handbook of Game Theory with Economic Applications, vol. 3, chap. 53, pp. 2025–2054. Elsevier, 1 edn. (2002)