A First Comparison of Abstract Argumentation Systems: A Computational Perspective

Stefano Bistarelli[1,2], Fabio Rossi[1], and Francesco Santini[3]

[1] Department of Mathematics and Computer Science, University of Perugia, Italy
[bista,rossi]@dmi.unipg.it
[2] Institute of Informatics and Telematics (CNR), Pisa, Italy
stefano.bistarelli@iit.cnr.it
[3] EPI Contraintes, INRIA Rocquencourt
francesco.santini@inria.fr

Abstract. In this paper we introduce an initial comparison among three different implementations of Abstract Argumentation Systems: ASPARTIX, ConArg, and Dung-O-Matic. These tools are tested over four different kinds of interaction graphs, corresponding to Erdős-Rényi networks, scale-free Barabasi networks, Watts-Strogatz, and Kleinberg small-world networks. Our final goal is to thoroughly evaluate their performance (in this work we test complete and stable semantics only), and to find the most efficient one, but also, more in general, to better study this kind of problems from the computational point of view.

1 Introduction

An Abstract Argumentation Framework (AF), or System, as introduced in the seminal paper by Dung [5], is simply a pair $\langle \mathcal{A}, R \rangle$ consisting of a set A whose elements are called arguments, and of a binary relation R on A, called "attack" relation. Several different semantics (i.e., extensions) can be obtained over arguments by considering subsets of arguments with specific properties. An AF has an obvious representation as a directed graph where nodes are arguments and edges are drawn from attacking to attacked arguments.

The main aim of this work is to better understand this kind of systems under the perspective of their computational performance, in terms of networks with different properties and size. Indeed, existent and future applications [9] exploiting AFs need to efficiently behave and scale over "large" networks with properties close to real-world ones. In fact, in complex discussions it is not hard to find 50-100 arguments at least, especially if we consider on-line open fora, or discussion groups. When we make these digital tribunes correspond to well-known social networks, as *Twitter* or *Facebook*, but also to more structured debate-friendly tools, as DebateGraph[4], then the number of arguments can further increase due to the a high number of users participating to a discussion. The different intrinsic nature of these graphs leads to find more or less extensions of some kind,

[4] http://debategraph.org

e.g., stable ones [5]. In our tests we adopt two different types of small-world networks (*Kleinberg* [11] and *Watts-Strogatz* [13]), one class of scale-free networks (*Barabasi* [1]), and *Erdős-Rényi* [8] networks (see Sec. 3.2).

Our main goal is to compare three different tools that are more oriented to a straight computation of extensions, than on providing support to negotiation or decision-making. These tools correspond to ASPARTIX, ConArg, and Dung-O-Matic, introduced in Sec. 3.

2 Dung Argumentation

In [5], Dung has proposed an abstract framework for argumentation in which the focus is on the definition of the argument status.

Definition 1. *An Argumentation Framework (AF) is a pair $\langle \mathcal{A}, R \rangle$ of a set $\mathcal{A}$ of arguments and a binary relation R on $\mathcal{A}$ called the attack relation. $\forall a_i, a_j \in \mathcal{A}$, $a_i R\, a_j$ means that a_i attacks a_j. An AF may be represented by a directed graph (the interaction graph) whose nodes are arguments and edges represent the attack relation. A set of arguments $\mathcal{B}$ attacks an argument a if a is attacked by an argument of $\mathcal{B}$. A set of arguments $\mathcal{B}$ attacks a set of arguments $\mathcal{C}$ if there is an argument $b \in \mathcal{B}$ which attacks an argument $c \in \mathcal{C}$.*

Dung gave several semantics to "acceptability". These various semantics produce none, one or several acceptable sets of arguments, called "extensions". In Def. 2 we define the concepts of conflict-free and stable extensions:

Definition 2. *A set $\mathcal{B} \subseteq \mathcal{A}$ is conflict-free in a given $\langle \mathcal{A}, R \rangle$ iff no two arguments a and b in $\mathcal{B}$ exist such that a attacks b. A conflict-free set $\mathcal{B} \subseteq \mathcal{A}$ is a stable extension iff for each argument which is not in $\mathcal{B}$, there exists an argument in $\mathcal{B}$ that attacks it.*

The other semantics for "acceptability" rely upon the concept of defense:

Definition 3. *Given $\langle \mathcal{A}, R \rangle$, an argument b is defended by a set $\mathcal{B} \subseteq \mathcal{A}$ (or $\mathcal{B}$ defends b) iff for any argument $a \in \mathcal{A}$, if a attacks b then $\mathcal{B}$ attacks a.*

An admissible set of arguments according to Dung must be a conflict-free set which defends all its elements. Formally:

Definition 4. *Given $\langle \mathcal{A}, R \rangle$, a conflict-free set $\mathcal{B} \subseteq \mathcal{A}$ is admissible iff each argument in $\mathcal{B}$ is defended by $\mathcal{B}$.*

Besides the stable semantics, three more semantics refining admissibility have been introduced in [5]:

Definition 5. *Given a $\langle \mathcal{A}, R \rangle$, a preferred extension is a maximal (w.r.t. set inclusion) admissible subset of $\mathcal{A}$. An admissible $\mathcal{B} \subseteq \mathcal{A}$ is a complete extension iff each argument which is defended by $\mathcal{B}$ is in $\mathcal{B}$. The least (w.r.t. set inclusion) complete extension is the grounded extension.*

3 Tools and Graphs

In the following of this section we describe our benchmark environment. We organize the content into two subsections: Sec. 3.1 concerns a more detailed description of the three tools we used in our comparison, while Sec. 3.2 describes in detail the random networks we generated as benchmark.

3.1 Tools

ASPARTIX. The *ASPARTIX*[5] system [7, 6] is an ASP-based tool for computing acceptable extensions for a broad range of formalizations of Dung's AFs and generalisations, e.g., value-based AFs [2]. *ASPARTIX* relies on a fixed disjunctive *Datalog* program (ASP also includes default negation) which takes an instance of an argumentation framework as input, and uses an ASP solver (as DLV) for computing the type of extension specified by the user. ASPARTIX is able to solve admissible, stable, complete, grounded, preferred, semi-stable, ideal, stage, cf2, resolution-based grounded and stage2 extensions.

ConArg. ConArg[6] [4, 3] is a constraint-based tool based on the *Java Constraint Programming* solver[7] (*JaCoP*), a Java library that provides the Java user with a *Finite Domain Constraint Programming* paradigm [12]. All the properties describing conflict-free, admissible, complete, stable, grounded and preferred extensions have been modeled and implemented with constraints. In this paper we consider a (faster) command-line version of ConArg. To model all the semantics presented in Sec. 2 we used MiniZinc[8], which is a medium-level constraint modelling language that can be mapped onto different existing solvers.

Dung-O-Matic. Dung-O-Matic[9] is an abstract argument computation engine implemented by the *javaDungAF* Java class. Dung-O-Matic supports Dung's AFs and several of their semantics, namely admissible, complete, eager, grounded, ideal, preferred, semi-stable, and finally stable. In order to find each of the proposed extensions, this tool implements a different algorithm presented in literature; for instance, the grounded semantics is computed with the original algorithms presented by Dung in [5].

3.2 Networks

Due to the lack of well-established benchmarks using real interaction graphs, we decided to randomly generate our test networks. To generate random graphs we adopted two different tools. The first one is *Java Universal Network/Graph Framework* (*JUNG*, `http://jung.sourceforge.net`), which is a Java software library for the modeling, generation, analysis and visualization of graphs. With

[5] `http://www.dbai.tuwien.ac.at/proj/argumentation/systempage/`

[6] `https://sites.google.com/site/santinifrancesco/tools/ConArg.zip`

[7] `http://www.jacop.eu`

[8] `http://www.minizinc.org`

[9] `http://www.arg.dundee.ac.uk/?page_id=279`

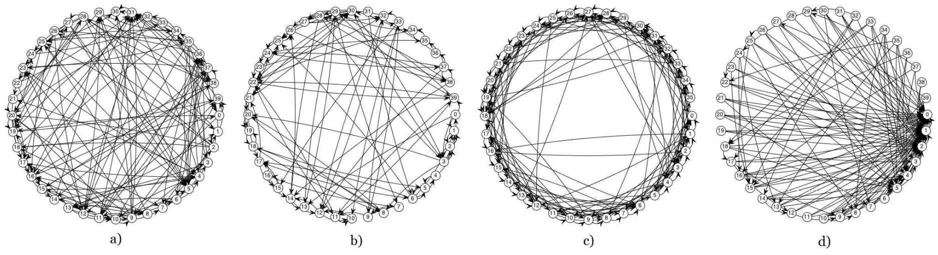

Fig. 1: An example of a *a)* Erdős-Rényi, *b)* Watts-Strogatz, *c)* Kleinberg, and *d)* a Barabasi graph, all with around 40 nodes.

JUNG we generated *Barabasi* [1] and *Kleinberg* [11] networks. The second library we used is named *NetworkX* (`http://networkx.github.io`), and it consists of a Python software package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks. With this tool we were able to generate *Erdős-Rényi* [8] and *Watts-Strogatz* [13]. We use two different libraries because of their different capabilities: with JUNG we are able to randomly generate directed Barabasi and Kleinberg networks, while NetworkX does not cover Kleinberg networks at all, and only provides undirected Barabasi ones. On the other side, NetworkX offers both Watts-Strogatz and Erdős-Rényi networks (not present in JUNG). Four examples of these networks are represented in Fig. 1. Our attention is more focused on testing small-world networks because we consider real large interaction-graphs as coming from social networks [10, 9].

4 Preliminary Tests and Conclusion

In Fig. 2 and Fig. 3 we show a first comparison among the three systems presented in Sec. 3.1, testing complete and stable extensions respectively. On the x-axis we report the exact number of nodes of each set of networks we use: in each set, we considered 50 random networks for each of the four classes reported in Sec. 3.2, for a total of 200 networks. On the y-axis we report the average CPU time to compute all the complete/stable extensions on that given set of networks. These first tests show that ConArg performs better than the other two systems, even if ASPARTIX has comparable results.

Performance has been collected using an Intel(R) Core(TM) i7 2.4Ghz processor, with 16Gb of RAM. To run ASPARTIX, we have used the last version of DLV (December 2012) with *Gringo* v3.0.5 and *ClaspD* v1.1.4 as extensions.

In this work we have commenced a study on the computational efficiency of nowadays tools dealing with AFs [5]. Our will is to extend this study under several lines. First, we will show tests on more computationally-demanding semantics (e.g., preferred extensions), we will test hard problems in Argumentation (e.g., deciding if a set is a preferred extension is CO-NP-complete), and, finally, we will better define the link with small-world networks, by finding the most appropriate random-network model. At last, we will test these tools over

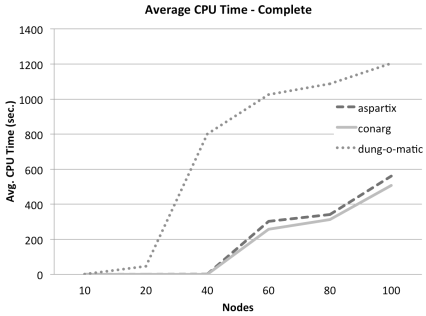

Fig. 2: Comparing complete extensions.

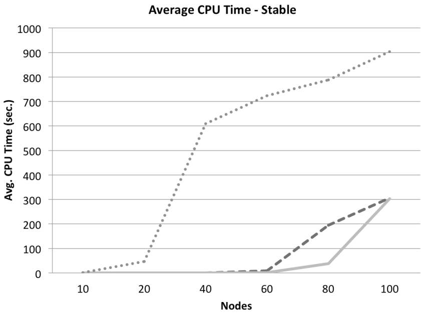

Fig. 3: Comparing stable extensions.

larger networks (e.g., 1,000 arguments), to check how feasible it is to use them in a real application, as, for instance, large-scale agreements via microdebates [9].

References

1. A. L. Barabasi and R. Albert. Emergence of scaling in random networks. *Science*, 286(5439):509–512, 1999.
2. T. J. M. Bench-Capon. Persuasion in practical argument using value-based argumentation frameworks. *J. Log. Comput.*, 13(3):429–448, 2003.
3. S. Bistarelli and F. Santini. A common computational framework for semiring-based argumentation systems. In *European Conference on Artificial Intelligence*, volume 215 of *Frontiers in A.I. and Applications*, pages 131–136. IOS Press, 2010.
4. S. Bistarelli and F. Santini. Conarg: A constraint-based computational framework for argumentation systems. In *ICTAI*, pages 605–612. IEEE, 2011.
5. P. M. Dung. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. *Artif. Intell.*, 77(2):321–357, 1995.
6. W. Dvorák, S. A. Gaggl, J. P. Wallner, and S. Woltran. Making use of advances in answer-set programming for abstract argumentation systems. *CoRR*, abs/1108.4942, 2011.
7. U. Egly, S. A. Gaggl, and S. Woltran. ASPARTIX: Implementing argumentation frameworks using answer-set programming. In *International Conference on Logic Programming (ICLP)*, pages 734–738. LNCS, Springer, 2008.
8. P. Erdős and A. Rényi. On the evolution of random graphs. *Bull. Inst. Internat. Statist*, 38(4):343–347, 1961.
9. S. Gabbriellini and P. Torroni. Large scale agreements via microdebates. In *AT*, volume 918 of *CEUR Workshop Proceedings*, pages 366–377. CEUR-WS.org, 2012.
10. S. Gabbriellini and P. Torroni. Arguments in social networks. In *AAMAS*, pages 1119–1120. IFAAMAS, 2013.
11. C. Martel and V. Nguyen. Analyzing Kleinberg's (and other) small-world models. In *PODC '04*, pages 179–188, New York, NY, USA, 2004. ACM.
12. F. Rossi, P. van Beek, and T. Walsh. *Handbook of Constraint Programming (Foundations of Artificial Intelligence)*. Elsevier Science Inc., New York, NY, USA, 2006.
13. D. J. Watts and S. H. Strogatz. Collective dynamics of "small-world" networks. *nature*, 393(6684):440–442, 1998.