A First Comparison of Abstract Argumentation Reasoning-Tools

Stefano Bistarelli[1] and **Fabio Rossi**[1] and **Francesco Santini**[2]

Abstract. We compare three different implementations of reasoning tools dedicated to Abstract Argumentation Frameworks. These systems are ASPARTIX, ConArg2, and Dung-O-Matic. They have been tested over three different random graph-models, corresponding to the Erdős-Rényi model, Kleinberg small-world model, and scale-free Barabasi model.

1 Introduction and Related Work

An *Abstract Argumentation Framework* (*AAF*), or System, as introduced in a 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. An abstract argument is not assumed to have any specific structure but, roughly speaking, an argument is anything that may attack or be attacked by another argument. The sets of arguments (or *extensions*) to be considered are then defined under different semantics, which are related to varying degrees of scepticism or credulousness.

The main goal of this paper is to better understand how well these extensions can be computed at the state of the art of modern Abstract Argumentation reasoners, in terms of argument networks with different properties and size. Therefore, we test three tools whose main objective is the pure computation of these extensions, i.e. ASPARTIX[3] [8], ConArg2[4] [4] (our tool, tested in this paper for the first time), and Dung-O-Matic[5], which implements different well-known algorithms in the literature. We consider three different random graph-models, thus assembling a variegate benchmark for this kind of testing. These networks are respectively generated according to the Erdős-Rényi [9] model (generated with the *NetworkX*[6] library), Kleinberg [11], and the Barabasi-Albert [1] models (both generated with the *JUNG*[7] library). We have not considered AAFs collected from the "real-world" due to the current lack of mining tools.

The semantics of interest in our comparison are the admissible (*adm*), complete (*com*), stable (*stb*), grounded (*gde*), preferred (*com*), semi-stable (*sem*), and ideal semantics (*ide*).

To the best of our knowledge, the performance results presented in [4, 6, 7] are the first ones proposed on medium-large problems, and [4] groups the first ones using random networks showing well-known small-world topologies (i.e., Kleinberg and Barabasi-Albert). The justification behind using this kind of graphs is that several works in the Argumentation literature investigate AAFs extracted from social networks [10].

In [6, 7] the authors randomly generate AAFs by using two parameterized methods for generating the attack relation. The first one generates arbitrary graphs and inserts for any pair (a, b) an attack from a to b with a given probability p (i.e., similarly to Erdős-Rényi). The other method generates AAFs with a $n \times m$ grid structure. They consider two different neighbourhoods, one connecting arguments vertically and horizontally, and one that additionally connects the arguments diagonally. Such a connection is a mutual attack with a given probability p and in only one direction otherwise. The probability p is chosen between 0.1 and 0.4. The authors generate AAFs with 60-200/25-500 arguments and try to solve each problem within a timeout of 300 seconds.

2 Tests and Discussion

The results are collected on an Intel(R) Core(TM) i7 CPU 970 @3.20GHz (6 core, 2 threads per core), and 16GB of RAM. For all the tools, the output has been redirected to */dev/null*. To test ASPARTIX we used *metasp* optimization where available in conjunction with *gringo 3.0.5* and *claspD 1.1.4*, *DLV* build "BEN/Dec 16 2012 gcc 4.6.1" otherwise; for ideal extensions only a DLV model is available. We set a timeout of 300 seconds to solve each kind of semantics.

In Tab. 1 we show the results for finding all the extensions for a given semantics. Time results (in seconds) are averaged over 100 networks, for each given model and number of nodes, and only for the extensions found within the timeout. For each tool we tested a total of 8.300 random AAFs.

Table 2 summarises the tests showing the winner for each extension. Dung-O-Matic (D) works well with grounded extensions, meaning that the polynomial algorithm is often better than representing the problem in a declarative way. With all the other problems, however, Dung-O-Matic is often not able to solve the instances within the timeout. For what concerns the other two tools, we can see that ConArg2 (C) works very well with Barabasi networks, slightly better than ASPARTIX (A) on Kleinberg graphs (but worse on ideal and preferred), and, finally, higher-level extensions in Erdős-Rényi

[1] Department of Mathematics and Computer Science, University of Perugia, Italy, email: [bista,rossi]@dmi.unipg.it. Stefano Bistarelli is supported by MIUR-PRIN "Metodi logici per il trattamento dell'informazione".

[2] Institute of Informatics and Telematics (C.N.R Pisa), Italy, email: francesco.santini@iit.cnr.it. The author is supported by MIUR PRIN "Security Horizons".

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

[4] http://www.dmi.unipg.it/bista/tt/conarg/

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

[6] http://networkx.github.io

[7] http://jung.sourceforge.net

		Barabasi-Albert				Kleinberg				Erdős-Rényi			
		30	**40**	**50**	**60**	**25**	**36**	**49**	**64**	**100**	**150**	**200**	**300**
adm	ASPARTIX	0.06	0.97	30.3	124	0.08	2.57	157	-	0.98	15.1	81.3	96.4
	ConArg2	0.01	0.04	1.54	28.7	0.01	0.13	8.43	-	0.05	1.91	13.3	11
	Dung-O-Matic	0.93	5.74	80.1	169	214	-	-	-	158	-	-	-
		2.5K	**5K**	**10K**	**20K**	**25**	**36**	**49**	**64**	**100**	**150**	**200**	**300**
com	ASPARTIX	0.54	1.16	2.57	5.49	0.03	0.67	20.9	-	0.02	0.12	1.69	83.5
	ConArg2	0.19	0.47	1.12	2.5	0.01	0.06	1.59	84.4	0.01	0.07	0.73	68.7
	Dung-O-Matic	-	-	-	-	-	-	-	-	160	-	-	-
		2.5K	**5K**	**10K**	**20K**	**25**	**36**	**49**	**64**	**100**	**150**	**200**	**300**
stb	ASPARTIX	0.51	1.1	2.47	5.25	0.01	0.05	0.54	8.01	0.01	0.04	0.22	11.32
	ConArg2	0.12	0.29	0.65	1.43	0.01	0.01	0.04	0.5	0.01	0.02	0.09	5.88
	Dung-O-Matic	-	-	-	-	217	-	-	-	160	-	-	-
		2.5K	**5K**	**10K**	**20K**	**25**	**36**	**49**	**64**	**100**	**150**	**200**	**300**
prf	ASPARTIX	5.62	25.4	114.3	252	0.35	4.67	85.2	-	0.33	0.6	0.99	3.04
	ConArg2	0.01	0.04	0.1	0.17	0.04	1.96	230	-	0.04	0.44	4.53	117
	Dung-O-Matic	-	-	-	-	212	-	-	160	-	-	-	-
		2.5K	**5K**	**10K**	**20K**	**25**	**36**	**49**	**64**	**100**	**150**	**200**	**300**
gde	ASPARTIX	-	-	-	-	0.44	9.48	39.1	131	0.16	0.5	1.16	3.92
	ConArg2	0.07	0.2	1.12	2.51	0.01	0.07	0.11	0.33	0.01	0.09	0.9	75.9
	Dung-O-Matic	20.8	283	-	-	0.19	0.51	0.98	1.61	0.17	0.2	0.24	0.39
		2.5K	**5K**	**10K**	**20K**	**25**	**36**	**49**	**64**	**100**	**150**	**200**	**300**
sem	ASPARTIX	4.03	16.7	76.5	165.5	0.24	1.33	9.6	141	0.38	0.73	1.24	9.4
	ConArg2	0.01	0.03	0.07	0.11	0.01	0.02	0.17	2.73	0.01	0.09	0.81	68.4
	Dung-O-Matic	-	-	-	-	259	-	-	-	-	-	-	-
		2.5K	**5K**	**10K**	**20K**	**16**	**25**	**36**	**49**	**100**	**150**	**200**	**300**
ide	ASPARTIX	-	-	-	-	0.67	6.15	38.7	179	119	-	-	-
	ConArg2	0.13	0.5	5.98	21.2	0.01	0.04	1.96	230	0.04	0.45	4.53	117
	Dung-O-Matic	-	-	-	-	1.21	214	-	-	160	-	-	-

Table 1: The average time (over 100 networks and in seconds) for each tested semantics; in bold, the number of arguments.

graphs are usually solved better by ASPARTIX.

Moreover, we would like to point out that the model of the graph sensitively impacts on the performance of the tool: for instance, Barabasi networks are easier to be solved (with ConArg2 in particular) since the number of nodes can be raised to thousands still solving the problem. On the other end, it is possible to work only with Kleinberg graphs with less than one hundred nodes (except for grounded extensions). Erdős-Rényi stays in the middle. Therefore, these tests shows it is really important to discover the structure of real AAFs, before developing the technology to efficiently work on them.

In addition, we tested the "Anthropogenic Climate Change" map (1.190 arguments), which we imported from DebateGraph[8], a collaborative tool for the development of debates on specific topics. With the premise that edges in this kind of maps also represent relations different from attacks (e.g., "is related to"), the same tests show a strong similarity with the Barabasi model, i.e., ConArg2 solves all the semantics almost instantly, while ASPARTIX stops after 35 (prf), 2.457 (gde), and 77 (sem) seconds (timeout for the others). Dung-O-Matic is only able to find the gde in 0.3 seconds.

3 Future Work

We plan to extend ConArg2 to solve weighted problems [3], and the hard problems implemented in CEGARTIX and dynPARTIX (e.g., the credulous or skeptical acceptance of an argument in preferred extensions) in order to have a comparison with these tools (as recently proposed in [2] on the stable semantics). We also plan to design and implement specific search-heuristics in order to improve the performance with higher-order extensions (e.g., preferred), so to better manage maximality of set inclusion. Note that such heuristics can be tuned on a specific graph model. Finally, we would like to test the same tools over AAFs extracted from real debates, and to study their topology, in order to match them to specific random graph-models.

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

	adm	com	stb	prf	gde	sem	ide
Barabasi	C	C	C	C	C	C	C
Kleinberg	C	C	C	A	C/D	C	A
Erdős	C	C	A/C	A	A/D	A	C

Table 2: Winners from Tab. 1, considering the behaviour on largest graphs: (A)SPARTIX, (C)onArg2, (D)ung-O-Matic.

REFERENCES

[1] A. L. Barabasi and R. Albert, 'Emergence of scaling in random networks', *Science*, **286**(5439), 509–512, (1999).

[2] S. Bistarelli, F. Rossi, and F. Santini, 'Benchmarking hard problems in random abstract AFs: The stable semantics', in *Proceedings of the Fifth International Conference on Computational Models of Argument*, FAIA. IOS Press, (2014).

[3] S. Bistarelli and F. Santini, 'A common computational framework for semiring-based argumentation systems', in *ECAI 2010 - 19th European Conference on Artificial Intelligence*, volume 215 of *FAIA*, pp. 131–136. IOS Press, (2010).

[4] S. Bistarelli and F. Santini, 'Conarg: A constraint-based computational framework for argumentation systems', in *Proceedings of the 2011 IEEE 23rd International Conference on Tools with Artificial Intelligence*, ICTAI '11, pp. 605–612. IEEE Computer Society, (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. Dvořák, M. Morak, C. Nopp, and S. Woltran, 'dynpartix-a dynamic programming reasoner for abstract argumentation', in *Applications of Declarative Programming and Knowledge Management*, 259–268, Springer, (2013).

[7] W. Dvořák, M. Järvisalo, J. P. Wallner, and S. Woltran, 'Complexity-sensitive decision procedures for abstract argumentation', *Artif. Intell.*, **206**, 53–78, (January 2014).

[8] U. Egly, S. A. Gaggl, and S. Woltran, 'Answer-set programming encodings for argumentation frameworks', *Argument & Computation*, **1**(2), 147–177, (2010).

[9] P. Erdős and A. Rényi, 'On the evolution of random graphs', *Bull. Inst. Internat. Statist*, **38**(4), 343–347, (1961).

[10] S. Gabbriellini and P. Torroni, 'Arguments in social networks', in *Proceedings of the 2013 International Conference on Autonomous Agents and Multi-agent Systems*, AAMAS '13, pp. 1119–1120. IFAAMAS, (2013).

[11] C. Martel and V. Nguyen, 'Analyzing Kleinberg's (and other) small-world models', in *Proceedings of the ACM symposium on Principles of distributed computing*, PODC '04, pp. 179–188, New York, NY, USA, (2004). ACM.