Generating Challenging Benchmark AFs

Federico CERUTTI [a,1], Massimiliano GIACOMIN [b], and Mauro VALLATI [c]

[a] *Department of Computing Science, King's College, University of Aberdeen, UK*

[b] *Department of Information Engineering, University of Brescia, Italy*

[c] *School of Computing and Engineering, University of Huddersfield, UK*

Abstract. In this paper we describe the AFBenchGen system, which allows to automatically generate randomised argumentation frameworks for testing purposes.

Keywords. argumentation frameworks, argumentation benchmarks, algorithm evaluation

1. Introduction

Dung's abstract argumentation framework (AF) [1] provides a fundamental reference in computational argumentation. An AF consists of a set of arguments and an *attack* relation between them. The concept of *extension* plays a key role in this simple setting, where an *extension* is a set of arguments which can "survive the conflict together". Different notions of extensions and of the requirements they should satisfy correspond to alternative *argumentation semantics*.

Recently, several algorithms (e.g. [2, 3, 4]) have been proposed to solve problems associated to argumentation semantics. So far, no extensive and comprehensive comparison of different solvers have been performed: this is mainly due to the lack of a large set of challenging and suitable AFs. Testing frameworks can be generated either by encoding real (or more general) problems or randomly. While there are some repositories of real argumentation instances[2], the generation of "good" random AFs instances is still an open question, differently from other communities in AI such as SAT or Automated Planning.

In this work we present AFBenchGen, a framework for generating random argumentation frameworks. The properties of generated AFs can be customised through a number of parameters.

2. Overview of AFBenchGen

AFBenchGen is implemented[3] in C++, and the generated argumentation frameworks are represented primarily in the Aspartix file format [2]. The Aspartix format has been

[1] Corresponding Author

[2] http://www.arg.dundee.ac.uk/aif-corpora/

[3] http://tiny.cc/afbenchgen

chosen since it is already used in many existing argumentation systems, and it can be easily exported in other formats.

Among other "standard" parameters, such as the number of arguments and the probability of attacks (exploited in [3]), `AFBenchGen` allows one to specify a set of parameters related to the structure of the generated graph. Currently, we implemented the possibility to generate complex graphs structures relying on the notion of strongly connected components (SCCs), which lay at the basis of the evaluation carried out in [4]. Among others, `AFBenchGen`'s parameters include the total number of SCCs and their dimensions. Moreover, `AFBenchGen` has the capability to deal with non-uniform random distribution: as a proof of concept, we allow such parameters to be provided under the form of average and standard deviation of a normal distribution, thus allowing intra-class (e.g. the class of AFs with fixed number of arguments) randomisation.

Further parameters allow the configuration of the generated frameworks. For instance, it is possible to determine (probabilistically) the density of attacks for each SCC; and how many arguments (probabilistically) in each SCC attack how many arguments (probabilistically) in how many (probabilistically) other SCCs.

3. Discussion

We believe that computational argumentation field is now mature enough for proposing solutions to industry. In order to foster the implementation of efficient algorithms, and to allow a fair comparison between them, a large number of argumentation frameworks with different structural characteristics are needed.

`AFBenchGen` is at its first stage of development, but the implementation showed good scalability performance. In [4], it demonstrated to be able to generate AFs with a number of arguments and attacks respectively up to 5,000 and 270,000. Despite this, we are interested in identifying parameters for increasing the ability of `AFBenchGen` in generating significantly different AFs, that would provide a more complete overview on algorithms performance. This is clearly related to the study of relevant features for argumentation problems which is still at its early stage of study [5].

References

[1] 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):321–358, 1995.

[2] Wolfgang Dvořák, Sarah Alice Gaggl, Johannes Wallner, and Stefan Woltran. Making use of advances in answer-set programming for abstract argumentation systems. In *Proc. of INAP 2011*, 2011.

[3] Federico Cerutti, Paul E. Dunne, Massimiliano Giacomin, and Mauro Vallati. Computing Preferred Extensions in Abstract Argumentation: A SAT-Based Approach. In *TAFA 2013*, pages 176–193. 2014.

[4] Federico Cerutti, Massimiliano Giacomin, Mauro Vallati, and Marina Zanella. A SCC Recursive Meta-Algorithm for Computing Preferred Labellings in Abstract Argumentation. In *Proc. of KR 2014*, 2014.

[5] Mauro Vallati, Federico Cerutti, and Massimiliano Giacomin. Argumentation Frameworks Features: an Initial Study. In *Proc. of ECAI2014*, 2014. to appear.