Assessment of benchmarks for abstract argumentation

Jean-Guy Mailly\textsuperscript{a,*} and Marco Maratea\textsuperscript{b}

\textsuperscript{a} LIPADE, Université de Paris, France
E-mail: jean-guy.mailly@parisdescartes.fr
\textsuperscript{b} DIBRIS, Università di Genova, Italy
E-mail: marco@dibris.unige.it

Abstract. In this paper, we provide an overview of the benchmarks that have been recently employed in Abstract Argumentation. We first describe the benchmark suite from previous editions of the International Competition of Computational Models of Argumentation (ICCMA), and then briefly describe the benchmarks for non-Dung frameworks. This article is a contribution to the new Argument & Computation Community Resources (ACCR) corner.

Keywords: Computational argumentation, community resources, benchmarks

1. Introduction

Computational argumentation has been a topic of interest for decades, especially since Dung’s seminal paper on Abstract Argumentation Frameworks (AFs) \cite{Dung1995}. It has applications in various domains such as decision making \cite{Dung1996}, automated negotiation \cite{Dung2017}, reasoning with inconsistent knowledge \cite{Dung2018}, legal reasoning \cite{Dung2017}, and multi-agent systems \cite{Dung2017}. However, the main efforts towards the development of efficient computational approaches are quite recent. These efforts have been channeled by the organization of the first International Competition on Computational Models of Argumentation\textsuperscript{1} (ICCMA) in 2015 \cite{Dung2015}. This event allowed to compare the different approaches to solving the main reasoning tasks for an agent using mainly randomly generated AF benchmarks. The second edition of the competition (ICCMA 2017)\textsuperscript{2} \cite{Dung2016,Dung2017} introduced more semantics, and utilized for the first time a specific \textit{call for benchmarks} for improving the quality and quantity of the benchmarks set, in particular regarding the non-random benchmarks. The third edition of the competition (ICCMA 2019)\textsuperscript{3} has added a new kind of reasoning problem: dynamic argumentation, consisting in dealing with a sequence of updated versions of a previously solved instance of a “classical” reasoning problem for AFs.

\textsuperscript{*}Corresponding author. E-mail: jean-guy.mailly@parisdescartes.fr.
\textsuperscript{1}http://argumentationcompetition.org/
\textsuperscript{2}http://argumentationcompetition.org/2017/
\textsuperscript{3}https://www.iccma2019.dmi.unipg.it

1946-2166/19/$35.00 \textcopyright 2019 – IOS Press and the authors.
This article is published online with Open Access and distributed under the terms of the Creative Commons Attribution Non-Commercial License (CC BY-NC 4.0).
The regular organization of a solver competition is important for theoretical and practical reasons, as proved in a number of neighbor fields, such as SAT [5], ASP [11, 22], and SMT. In particular, it allows the comparison of different kinds of algorithms that can solve reasoning tasks. It highlights the recent improvements of solving techniques for argumentation, and assesses the benchmarks that should be used by the community. About benchmarks, we expect good benchmarks to present some features. Especially, to be relevant in the context of a competition, the related instances must not be too easy or too hard to solve. And, more importantly, they must be representative of real world applications as much as possible. Thus, users of argumentation applications are able to pick the best solver for dealing with their concrete problem instances; at least this is the case if the concrete instance is similar to the benchmarks that have been used in the competition. Indeed, the efficiency of methods for solving hard problems is generally highly depending on the structure of the problem instances to be solved [9].

In this paper, we focus our attention on benchmarks, by presenting an overview of such resources that have been recently employed in Abstract Argumentation. In particular, we first describe the benchmark suite from previous editions of ICCMA, followed by a presentation of the benchmarks that have been proposed for variants and extensions of the original AFs proposed by Dung.

2. ICCMA benchmarks

In this section we outline the benchmark domains employed in the first two ICCMA competitions.

2.1. ICCMA’15 benchmarks

ICCMA’15 introduced three new AF generators, called GroundedGenerator, StableGenerator, and SccGenerator [15], each of them aiming to produce challenging AFs addressing certain aspects of computational difficulty, which are summarized below:

**GroundedGenerator.** This tool has been developed with the goal of producing AFs with large grounded extensions. It takes the number of arguments \( n \) and probability \( \text{probAttacks} \) as parameters, linearly orders the arguments and adds an attack from argument \( a \) to argument \( b \) in case \( a < b \) with probability \( \text{probAttacks} \). It then adds random attacks between the arguments not yet connected and the graph component obtained in the first part.

**SccGenerator.** This tool has been developed with the goal of producing AFs such that the graph features many Strongly Connected Components. It first partitions the arguments (the number of which is given by parameter \( n \)) into \( n\text{SCCs} \) (also given as parameter) components, which are linearly ordered. Within each component, attacks between pairs of arguments are added with probability given by a parameter. Among arguments of different components, attacks are added with probability given by another parameter, but under the condition that the component of the attacking arguments is ranked lower with respect to the linear order on components than the component of the attacked argument.

**StableGenerator.** This tool has been developed with the goal of producing AFs with a large number of stable extensions. It first identifies a set of arguments to form an acyclic subgraph of the AF and, consequently, to contain the grounded extension. Among the other arguments, subsets are iteratively singled out to form stable extensions by attacking all other arguments. Besides the

---

parameter $n$ for the number of arguments, the tool generates AFs taking into account a number of parameters which determine minimum and maximum values for the number and size of stable extensions and for the size of grounded extensions.

2.2. ICCMA’17 benchmarks

Herewith we briefly describe the domains that were evaluated at ICCMA’17, on top of those of ICCMA’15:

ABA2AF. These are assumption-based argumentation (ABA) benchmarks translated to AFs. ABA problems are one of the prevalent forms of structured argumentation in which, differently from AFs, the internal structure of arguments is made explicit through derivations from a more basic structure \[23,29\].

AdmBuster. These are crafted benchmark examples for (strong) admissibility featuring a fixed structure composed of 4 sets of arguments and predetermined sets of attacks. Two “starting” and “terminal” sets are composed of only one element, one having only outgoing edges and the other only incoming edges. Given a parameter $n$ of the generator, the two “intermediate” sets have cardinality $n$, and their attack relations are constructed in order to have only one complete labelling \[12\].

AFBenchGen2. This is a set of random AFs generators of three different graph classes, with a configurable number of arguments \[14\]. The three classes correspond to: (i) Erdös-Rényi \[19\], which selects attacks randomly; (ii) Watts-Strogatz \[30\], which aims for a small-world topology of networks being not completely random nor regular, and (iii) Barabasi-Albert \[6\] for large networks.

Planning2AF. These are AFs obtained from translating the well-known Blocksworld and Ferry planning domains. Each planning instance is first encoded as a propositional formula, by using the method in \[27\]; then, each clause is transformed into a material implication; and, finally, to each material implication the transformation in \[31\] is applied.

SemBuster. This is a crafted benchmark example for semi-stable semantics, with a fixed structure composed by 3 sets of arguments of equal cardinality, and predetermined sets of attacks. Attack relations are defined in a way that each instance has exactly $n + 1$ (with $n$ the number of arguments of each set) complete labellings that correspond also to preferred labellings, but only one among those corresponds to a semi-stable extension \[13\].

Traffic. These are graphs obtained from real world traffic networks data available at https://transitfeeds.com/ expressed as AFs. Given a graph, the corresponding AF contains the same set of vertices as the graph, and the attack relation is defined as follows: Given an existing edge, and a probability for the attack of being symmetric, the generator decides whether there are both attacks, or randomly selects the attack.

More detailed descriptions for all such domains can be found in the ICCMA’15 and ICCMA’17 home pages at http://argumentationcompetition.org.

3. Other benchmarks

In this section, we briefly describe other argumentation benchmarks that have been proposed, and for some of them we point out some directions that could lead to interesting new benchmarks.
Abstract Dialectical Frameworks (ADFs) are one of the main extensions of AFs, where every argument is equipped with an acceptance condition. The benchmarks employed in some of the recent papers on ADFs (e.g., [10,25]) are generated starting from some of the most “practical” benchmarks of ICCMA’17, e.g., Traffic and Planning2AF, and randomly generating acceptance conditions. Some of these benchmarks can be found at the YADF system page (http://www.dbai.tuwien.ac.at/proj/adf/yadf/).

Dynamic argumentation. Argumentation is highly dynamic by nature. Indeed, debates are situations where agents add new attacks and arguments at each step of the process. However, computational approaches specifically designed for such dynamic situations have only been proposed recently. For different semantics and reasoning problems, the approach to generate benchmarks is the same: AFs from ICCMA’15 or ICCMA’17 are used, and updates are randomly generated (see, e.g., [1]). A similar approach is used to generate dynamic bipolar AFs [2] and dynamic extended AFs [3]. This generation approach allows to easily benefit from the existing ICCMA benchmarks. However, like any random-based benchmarks, they miss some connection with real world instances. A possible direction to obtain more meaningful dynamic AFs is the study of real life debates (e.g., debates coming from a democratic process, from negotiation situations, …).

Structured argumentation. Obtaining an argumentation framework from a knowledge base is not a trivial problem. Indeed, even for some relatively small knowledge base, the number of arguments and attacks can be exponentially high with respect to the size of the knowledge base. We are aware of some efforts to efficiently generate argumentation frameworks from a logical knowledge base. For instance, [34] provides a method for generating AFs from existential rules knowledge bases; a data set and a generator are available online. A similar contribution for AFs based on Datalog± has been proposed in [32], while [33] provides such a generator for argumentation hypergraphs (i.e., with collective attacks) based on Datalog±.

Assumption-based argumentation. Assumption-based Argumentation (ABA) [16] is a well-known approach to structured argumentation, in which arguments are represented in a compact way in terms of graph-based derivations from a given rule-based deductive system over sentences, starting from assumptions, and checking for acceptability queries. Randomly generated ABA frameworks and queries used in a number of papers (e.g., [16,24]) can be found at http://robertcraven.org/proarg/experiments.html, and correspond to those frameworks utilized for contributing to ICCMA’17.

4. Conclusions

In this article we have presented a first contribution to the new born Argument & Computation Community Resources (ACCR) corner, in terms of an assessment of benchmarks for Abstract Argumentation. Subsequent contributions to the corner are expected to be focused on one particular resource (benchmark, software tool, etc), cf. the editorial introducing the corner in this issue, possibly among those we have overviewed here.

Acknowledgements

We would like to thank the Argument & Computation Editors-in-Chief, Pietro Baroni and Bart Verheij, for giving us the possibility to write this opening contribution to the new ACCR corner. We also would like to thank Martin Caminada for useful comments on a draft of this contribution.
References


