ConArg: Argumentation with Constraints *,**,***

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

[1] Dipartimento di Matematica e Informatica, Università di Perugia, Italy
bista,francesco.santini@dmi.unipg.it
[2] Istituto di Informatica e Telematica, IIT-CNR, Pisa, Italy
stefano.bistarelli@iit.cnr.it
[3] Centrum Wiskunde & Informatica, Amsterdam, Netherlands
F.Santini@cwi.nl

ConArg [3,4][4] is a tool based on *Constraint Programming* that is able to model and solve different problems related to (*Abstract*) *Argumentation Frameworks* (*AFs*) [6]. For the implementation we adopted *JaCoP*, which is a Java library that provides the user with a *Finite Domain Constraint Programming* paradigm [7]. Through its graphic interface, it is possible to select the extensions (e.g., admissible) the user wants to find, and to browse the obtained solutions.

Constraint Programming (CP) [7] is a powerful paradigm for solving combinatorial search problems, which exploits a wide range of techniques from artificial intelligence and operations research. The basic idea in constraint programming is that the user states the constraints and a general purpose constraint solver is used to solve them. Constraints are just relations, and a *Constraint Satisfaction Problem* (*CSP*) [7] states which relations should hold among the variables.

ConArg [3,4] is able to find all Dung's classical extensions [6] (i.e., conflict-free, admissible, complete, stable, grounded and preferred extensions) by defining the properties of these extensions through constraints, and solving the related CSP. To show the feasibility of such solution, in [3,4] we test the tool on different randomly generated small-world networks (i.e., *Barabasi* and *Kleinberg* ones) and we report the performance of the search in time. Since the total number of these extensions may explode for large sets of arguments (particularly in case of conflict-free extensions, i.e., the less constrained ones), it is important to use techniques to tackle this inherent complexity, as CP-based ones.

Moreover, ConArg can solve different classical hard-problems that concern weighted AFs (where attacks are associated with a "strength" value), as the ones related to weighted grounded extensions presented in [5]. For example, given a weighted argument system, a set of arguments $S \subseteq \mathcal{A}_{rgs}$ and an inconsistency budget β (i.e., the tolerated sum of the considered strength values), to find if β is minimal w.r.t. S represents a co-NP-complete problem [5].

$\star$ This work was carried out during the tenure of the ERCIM "Alain Bensoussan" Fellowship Programme, which is supported by the Marie Curie Co-funding of Regional, National and International Programmes (COFUND) of the European Commission.

$\star\star$ Research partially supported by MIUR PRIN 20089M932N project: "Innovative and multi-disciplinary approaches for constraint and preference reasoning".

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

Recently ConArg has been extended to encompass and solve some other semantics, as the stage, semi-stable and ideal extensions [6]. In addition, we enhanced the tool with the implementation of the extensions developed in [1,2].

In [1] we extend the Dung AFs in order to deal with coalitions of arguments. The initial set of arguments is partitioned into subsets. Each subset represents a different "line of thought" and can be considered as a coalition of arguments. All the found coalitions inherit the same semantics, e.g., all the coalitions in the same partition are, for instance, admissible. Therefore, in [1] we extend Dung's semantics from extensions to partitions of arguments, whose number, in general, can be combinatorial.

In [2] we revisit the concept of *Value-based AFs* [6] with the goal to unify many of the proposals into the same semiring-based framework, as long as the considered system of weights can be represented with a semiring structure. We suggest semirings as a mean to parametrically represent attack-weights of different Value-based AFs. For instance, a value may represent a "fuzziness", a "cost" or a probability score for a given attack. The novel idea is to provide a common quantitative framework, where it is possible to represent and compute weighted extensions. The defined Value-based AF is mapped into a semiring-based *Soft Constraint Satisfaction Problem* (*SCSP*) [7], and then solved [2].

In the future we plan to further extend ConArg along different lines. For example, we would like to introduce other extensions, as the *CF2* or the *Prudent* semantics [6]. Moreover, we want to develop ad-hoc heuristics to be used during the search, in order to improve the performance. Eventually, we want to test ConArg over large real small-world (i.e., social) networks, and to retrieve some statistical data for the different classical extensions (e.g., their average size).

References

1. Bistarelli, S., Campli, P., Santini, F.: Finding partitions of arguments with dung's properties via scsps. In: ACM Symposium on Applied Computing (SAC). pp. 913–919. ACM (2011)
2. Bistarelli, S., Santini, F.: A common computational framework for semiring-based argumentation systems. In: European Conference on Artificial Intelligence (ECAI10) (2010)
3. Bistarelli, S., Santini, F.: Conarg: A constraint-based computational framework for argumentation systems. In: 23rd International Conference on Tools with Artificial Intelligence, ICTAI 2011. pp. 605–612. IEEE (2011)
4. 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. Lecture Notes in Computer Science, vol. 7132, pp. 99–116. Springer (2012)
5. Dunne, P.E., Hunter, A., McBurney, P., Parsons, S., Wooldridge, M.: Weighted argument systems: Basic definitions, algorithms, and complexity results. Artif. Intell. 175(2), 457–486 (2011)
6. Rahwan, I., Simari, G.R.: Argumentation in Artificial Intelligence. Springer Publishing Company, Incorporated, 1st edn. (2009)
7. Rossi, F., van Beek, P., Walsh, T.: Handbook of Constraint Programming (Foundations of Artificial Intelligence). Elsevier Science Inc., New York, NY, USA (2006)