Fundamenta Informaticae 124 (2013) 383–401
DOI 10.3233/FI-2013-840
IOS Press

Coalitions of Arguments: An Approach with Constraint Programming*†

Stefano Bistarelli
Dipartimento di Matematica e Informatica, Università di Perugia, Via Vanvitelli 1, Perugia, Italy
Instituto di Informatica e Telematica (CNR), Via Moruzzi 1, Pisa, Italy
bista@dmi.unipg.it, stefano.bistarelli@iit.cnr.it

Francesco Santini‡
Contraintes, INRIA - Rocquencourt, Domaine de Voluceau BP 105, Le Chesnay Cedex, France
Centrum voor Wiskunde en Informatica, Science Park 123, Amsterdam, The Netherlands
Dipartimento di Matematica e Informatica, Università di Perugia, Via Vanvitelli 1, Perugia, Italy
francesco.santini@inria.fr, F.Santini@cwi.nl, francesco.santini@dmi.unipg.it

Abstract. The aggregation of generic items into coalitions leads to the creation of sets of homogenous entities. In this paper we accomplish this for an input set of arguments, and the result is a partition according to distinct *lines of thought*, i.e., groups of "coherent" ideas. We extend Dung's Argumentation Framework (AF) in order to deal with coalitions of arguments. The initial set of arguments is partitioned into not-intersected subsets. All the found coalitions show the same property inherited by Dung, e.g., all the coalitions in the partition are admissible (or conflict-free, complete, stable): they are generated according to Dung's principles. Each of these coalitions can be assigned to a different agent. We use *Soft Constraint Programming* as a formal approach to model and solve such partitions in weighted AFs: semiring algebraic structures can be used to model different optimization criteria for the obtained coalitions. Moreover, we implement and solve the presented problem with JaCoP, a Java constraint solver, and we test the code over a small-world network.

*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.
†Research partially supported by MIUR PRIN 2010-2011 2010FP79LR project "Logical Methods of Information Management", by the European Integrated Project 257414 ASCENS, and by COST Action IC0801
‡Address for correspondence: Contraintes, INRIA - Rocquencourt, Domaine de Voluceau BP 105, Le Chesnay Cedex, France

1. Introduction and Motivations

A coalition structure is a temporary alliance or partnering of similar entities (according to some criteria) in order to achieve a common purpose [23]. To form a successful coalition, it is necessary to efficiently recognise compatible interests, since each piece of information in the set to be clustered may be incorporated into two (or more) similar "lines of thought": the same information could be coherently accepted by both coalitions.

The abstract nature of Dung's seminal theory [20] of argumentation accounts for its widespread application for various species of non-monotonic reasoning. A Dung argumentation framework (see Section 2) is classically instantiated by arguments and a binary conflict-based attack relation, defined by some underlying logical theory. The justified arguments under different extensional semantics (e.g., conflict-free ones) are then evaluated, and the claims of these arguments define the inferences of the underlying theory.

The aim of this paper is to partition a given set of arguments into coalition-structures of them [17, 1, 14], where each coalition inherits the same Dung-like properties, e.g., admissibility (or stability). Therefore, each coalition of the obtained partition is eventually admissible (or stable). Thus, even if each coalition corresponds to an extension in the classical vision of Dung [20], we call it a "coalition" and not an "extension", to highlight the fact that now all the arguments are partitioned and each coalition forms a line of thought on his own, in "contrast" with the others. An application scenario is represented, for example, by the need to aggregate a set of distinct arguments into several acceptable or defensible lines of thought at the same time, in order to assign them to the different agents that have produced them. Suppose, for example, to collect some statements belonging to candidates of different political parties; it would be interesting to check how consistent their ideas are. For example, *a)* "We do not want immigrants with the right to vote" is clearly closer to *b)* "Immigration must be stopped", than to *c)* "We need a multicultural and open society in order to enrich the life of everyone and boost our economy". For this reason, arguments *a)* and *b)* on one side, and argument *c)* on the other one, may be assigned to two different admissible coalitions, corresponding to politicians P and Q respectively.

In general, cooperating groups, referred to as "coalition structures" [26], have been thoroughly investigated in AI and Game Theory and have proved to be useful in both real-world economic scenarios and Multi-agent Systems [26, 29, 3]. Some applications might be Task Allocation Problems (let tasks be the agents), Sensor Network Problems (agents must form groups of "event detectors"), distributed winner determination in Combinatorial Auctions, or agents grouping to handle work-flows [26, 29, 3].

In order to model and solve the proposed extended problems we use *(Soft) Constraint Programming* (*(S)CP*) [28] (see Section 3), which is a powerful paradigm for solving combinatorial problems that draws on a wide range of techniques from AI, Databases, Programming Languages, and Operations Research [28]. The idea of the semiring-based constraint formalism presented in [9, 5] is to further extend the classical constraint notion by adding the concept of a structure representing the levels of satisfiability of constraints. Such a structure is similar to a semiring (see Section 3). Problems defined according to the semiring-based framework are called *Soft Constraint Satisfaction Problems* (SCSPs) [9, 5, 28]. There already exist many efficient techniques (as constraint propagation, see Chapter 4 in [28]) to solve such complex problems. The solution of the obtained SCSP represents the partition of the arguments (see Section 4) where each subset (i.e., coalition) of arguments has the same property originally defined by Dung in [20], e.g., each coalition in the partition is admissible. Semirings can be used to relax conflict-free partitions, by allowing a certain degree of conflict inside coalitions, by representing a weight (or preference) associated with each attack between arguments.

At last, we propose an implementation of the proposed frameworks as crisp CSPs (equivalent to use a *Boolean* semiring in SCSPs) with the *JaCoP* (*i.e., Java Constraint Programming*) solver, and we test it over a small-world network randomly generated with the *Java Universal Network/Graph Framework* (*JUNG*) [27].[1] We also refer to the tool we have recently developed, named *ConArg* [12, 13], which is able to solve many Argumentation-related problems, the one presented in this paper included.

The paper extends the work in [6] and it is organized as follows: in Section 2 we provide the background about Argumentation Frameworks, while Section 3 summarizes the background on semiring-based constraints. Section 4 explains how to extend the theory behind [20] to deal with partitions of extensions and Section 5 presents how these partitions can be described as weighted. Then, Section 6 shows the problem of finding these partitions as a SCSP and Section 7 shows an implementation in JaCoP. At last, Section 8 reports the related work and Section 9 draws the final conclusions.

2. Dung's Argumentation

In his seminal papers, Dung has proposed an abstract framework for argumentation in which he focuses on the definition of the status (*attacked/defended*) of arguments [20]. It is assumed that the sets of arguments and conflicts among them are given as part of the problem.

Definition 2.1. ([20])
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 $c \in \mathcal{C}$.

Figure 1. A graphical rerepsentation of an AF on weather forecast; e.g., b attacks c and viceversa.

In Figure 1 we show an example of AF represented as an *interaction graph*: we have three different weather forecasts in contrast, where $R(a, b)$, $R(b, c)$, $R(b, a)$ and $R(c, b)$. Dung [20] gives several semantics of "acceptability", which produce none, one, or several acceptable sets of arguments, called extensions. The stable semantics is only defined via the notion of attack:

Definition 2.2. ([20])
A set $\mathcal{B} \subseteq \mathcal{A}$ is conflict-free iff for no two arguments a and b in $\mathcal{B}$, a attacks b. A conflict-free set $\mathcal{B} \subseteq \mathcal{A}$ is a stable extension iff each argument not in $\mathcal{B}$ is attacked by an argument in $\mathcal{B}$.

[1] The home page of JaCoP is `http://www.jacop.eu`

The other semantics for "acceptability" rely upon the concept of defense. An admissible set of arguments according to Dung must be a conflict-free set which defends all its elements. Formally:

Definition 2.3. ([20])
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. 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, one semantics refining admissibility has been introduced by Dung [20]:

Definition 2.4. ([20])
An admissible $\mathcal{B} \subseteq \mathcal{A}$ is a complete extension iff each argument which is defended by $\mathcal{B}$ is in $\mathcal{B}$.

In Figure 2 we show an example of a stable, an admissible, and a complete extension, respectively (A), (B), and (C). For instance, (B) is admissible but not complete because x_6 is defended by the extension (through x_4), but it is not contained in it. Moreover, (B) is not stable because x_6, which is not taken in the extension, is not attacked by it.

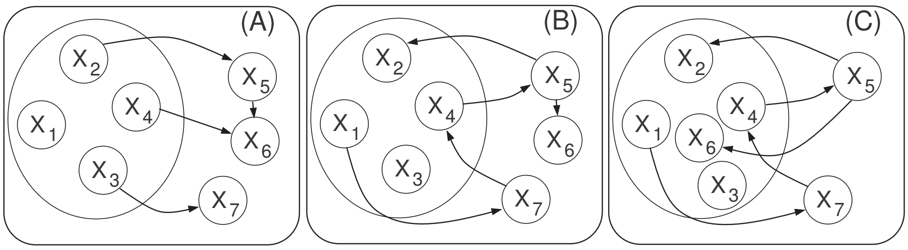

Figure 2. A stable (A), an admissible (B) and a complete (C) extension (clearly conflict-free as well).

3. Semirings and Soft Constraints

A c-semiring [9] (simply semiring in the sequel) is a tuple $S = \langle A, +, \times, \mathbf{0}, \mathbf{1} \rangle$, where A is a possibly infinite set with two special elements $\mathbf{0}, \mathbf{1} \in A$ (respectively the bottom and top elements of A) and with two operations $+$ and $\times$ that satisfy certain properties over A: $+$ is commutative, associative, idempotent, closed, with $\mathbf{0}$ as its unit element and $\mathbf{1}$ as its absorbing element; $\times$ is closed, associative, commutative, distributes over $+$, $\mathbf{1}$ is its unit element, and $\mathbf{0}$ is its absorbing element. The $+$ operation defines a partial order $\leq_S$ over A such that $a \leq_S b$ iff $a + b = b$; we say that $a \leq_S b$ if b represents a value *better* than a. Moreover, $+$ and $\times$ are monotone on $\leq_S$, $\mathbf{0}$ is the min of the partial order and $\mathbf{1}$ its max, $\langle A, \leq_S \rangle$ is a complete lattice and $+$ is its *least upper bound* operator (i.e., $a + b = lub(a, b)$) [9].

Some practical instantiations of the generic semiring structure are the *boolean* $\langle \{false, true\}, \vee, \wedge, false, true \rangle$, *fuzzy* $\langle [0..1], \max, \min, 0, 1 \rangle$, *probabilistic* $\langle [0..1], \max, \hat{\times}, 0, 1 \rangle$ and *weighted* $\langle \mathbb{R}^+ \cup \{+\infty\}, \min, \hat{+}, \infty, 0 \rangle$ (where $\hat{\times}$ and $\hat{+}$ respectively represent the arithmetic multiplication and addition). The *boolean* semiring can be used to represent classical crisp constraints.

Given a semiring $\langle A, +, \times, \mathbf{0}, \mathbf{1} \rangle$ and $a, b \in A$, we define the residuated negation of a as $\neg a = \max\{b : b \times a = \mathbf{0}\}$, where $\max$ is according to the ordering defined by $+$. Note that over the boolean

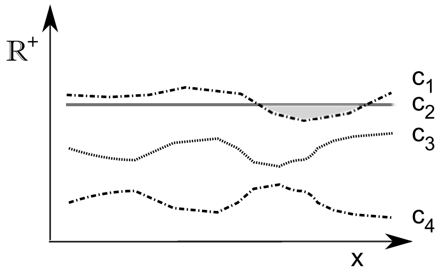

Figure 3. A graphical representation of four *weighted* constraints, e.g., $c_2 = c_3 \otimes c_4$.

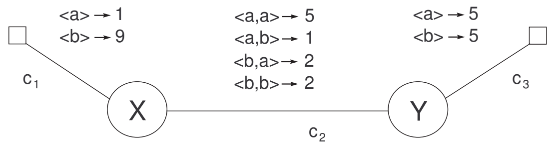

Figure 4. An SCSP based on a *weighted* semiring.

semiring the negation operator corresponds to the logic negation, since $\neg\mathbf{0} = \max\{b : b \times \mathbf{0} = \mathbf{0}\} = \mathbf{1}$, and $\neg\mathbf{1} = \max\{b : b \times 1 = \mathbf{0}\} = \mathbf{0}$.

A *soft constraint* [9] may be seen as a constraint where each instantiation of its variables has an associated preference. Given $S = \langle A, +, \times, \mathbf{0}, \mathbf{1}\rangle$ and an ordered finite set of variables V over a domain D, a soft constraint is a function that, given an assignment $\eta : V \rightarrow D$ of the variables, returns a value of the semiring, i.e., $c : (V \rightarrow D) \rightarrow A$. Let $\mathcal{C} = \{c \mid c : D^{|I \subseteq V|} \rightarrow A\}$ be the set of all possible constraints that can be built starting from S, D and V: any function in $\mathcal{C}$ depends on the assignment of only a (possibly empty) finite subset I of V, called the *support*, or *scope*, of the constraint. For instance, a binary constraint $c_{x,y}$ (i.e., $\{x, y\} = I \subseteq V$) is defined on the support $supp(c) = \{x, y\}$. Note that $c\eta[v = d]$ means $c\eta'$ where η' is η modified with the assignment $v = d$. Note also that $c\eta$ is the application of a constraint function $c : (V \rightarrow D) \rightarrow A$ to a function $\eta : V \rightarrow D$; what we obtain is, thus, a semiring value $c\eta = a$.[2]

Given the set $\mathcal{C}$, the combination function $\otimes : \mathcal{C} \times \mathcal{C} \rightarrow \mathcal{C}$ is defined as $(c_1 \otimes c_2)\eta = c_1\eta \times c_2\eta$ [9]; $supp(c_1 \otimes c_2) = supp(c_1) \cup supp(c_2)$. Given the set $\mathcal{C}$, the combination function $\oplus : \mathcal{C} \oplus \mathcal{C} \rightarrow \mathcal{C}$ is defined as $(c_1 \oplus c_2)\eta = c_1\eta + c_2\eta$ [5]; $supp(c_1 \oplus c_2) = supp(c_1) \cup supp(c_2)$. Informally, $\otimes$/$\oplus$ builds a new constraint that associates with each tuple of domain values for such variables a semiring element that is obtained by multiplying/summing the elements associated by the original constraints to the appropriate sub-tuples. Given a constraint $c \in \mathcal{C}$ and a variable $v \in V$, the *projection* [9] of c over $V \backslash \{v\}$, written $c \Downarrow_{(V \backslash \{v\})}$ is the constraint c' such that $c'\eta = \sum_{d \in D} c\eta[v = d]$. Informally, projecting means computing the best possible rating over all values of the remaining variables.

The partial order $\leq_S$ over $\mathcal{C}$ can be easily extended among constraints by defining $c_1 \sqsubseteq_S c_2 \iff \forall\eta, c_1\eta \leq_S c_2\eta$. In order to define constraint equivalence we have $c_1 \equiv_S c_2 \iff \forall\eta, c_1\eta =_S c_2\eta$ and $supp(c_1) = supp(c_2)$.

In Fig. 3 we show a graphical example of four *weighted* constraints (i.e., defined in the *weighted* semiring), where we have $c_3 \otimes c_4 = c_2$, $c_3 \sqsubseteq c_4$, $c_2 \sqsubseteq c_3$, $c_1 \sqsubseteq c_3$, but $c_1 \not\sqsubseteq c_2$ because of the grey region, where $c_2 \sqsubseteq c_1$ instead; moreover, in Fig. 3 we can see that $supp(c_1) = supp(c_2) = supp(c_3) = supp(c_4) = \{x\}$.

An SCSP [9] is defined as a quadruple $P = \langle S, V, D, C\rangle$, where $C \subseteq \mathcal{C}$ is the constraint set of the problem P. The *best level of consistency* notion defined as $blevel(P) = Sol(P) \Downarrow_{\emptyset}$, where $Sol(P) = \bigotimes C$ [9]. A problem P is α-consistent if $blevel(P) = \alpha$ [9]; P is instead simply "consistent" iff

[2]The constraint function $\bar{a}$ always returns the value $a \in A$ for all assignments of domain values, e.g., the $\bar{\mathbf{0}}$ and $\bar{\mathbf{1}}$ functions always return $\mathbf{0}$ and $\mathbf{1}$ respectively.

$blevel(P) >_S \mathbf{0}$ [9]. P is inconsistent if it is not consistent. Figure 4 shows an SCSP as a graph: S corresponds to the *weighted* semiring, i.e., $\langle \mathbb{R}^+ \cup \{+\infty\}, \min, \hat{+}, \infty, 0\rangle$. Variables ($V = \{x, y\}$) and constraints ($C = \{c_1, c_2, c_3\}$) are represented respectively by nodes and arcs (unary for c_1 and c_3, and binary for c_2), and semiring values are written to the right of each variable assignment of the constraint, where $D = \{a, b\}$. The solution of P in Fig. 4 associates a preference to every domain value of x and y by combining all the constraints, i.e., $Sol(P) = \bigotimes C$. For instance, for the assignment $\langle a, a\rangle$ (that is, $x = y = a$), we compute the sum of 1 (which is the value assigned to $x = a$ in constraint c_1), 5 (which is the value assigned to $\langle x = a, y = a\rangle$ in c_2) and 5 (which is the value for $y = a$ in c_3). Hence, the resulting preference value for this assignment is 11. The *blevel* for the example in Fig. 4 is 7, corresponding to the assignment $x = a, y = b$.

4. Extending Dung Argumentation to Coalitions

Given the set of arguments $\mathcal{A}_{rgs}$, the problem of coalition formation consists in selecting the appropriate partition of $\mathcal{A}_{rgs}$, $\mathcal{G} = \{\mathcal{B}_1, \ldots, \mathcal{B}_n\}$ ($|\mathcal{G}| = |\mathcal{A}_{rgs}|$ if each argument forms a coalition on its own), such that $\forall \mathcal{B}_i \in \mathcal{G}$, $\mathcal{B}_i \subseteq \mathcal{A}_{rgs}$ and $\mathcal{B}_i \cap \mathcal{B}_j = \emptyset$, if $i \neq j$. In this section we extend the Dung's semantics (see Section 2) in order to deal with a partition of arguments, that is we cluster the arguments into different subsets representing *distinct lines of thought*. An example representing the difference between the original framework [20] and our extension is illustrated in Figure 5: Figure 5 (A) represents a conflict-free extension as described in Definition 2.3, while Figure 5 (B) represents a conflict-free partition of arguments, since each coalition is conflict-free (see Definition 4.1). Thus, while in Dung it is sufficient to find only one set with the conflict-free property, here we want to find a partition of conflict-free sets with the given arguments; we can compute partitions by considering also the other properties as well, i.e., admissible, complete and stable semantics. Notice that, in general, we can have a combinatorial number of partitions for a given set of arguments [15, 26]. For example, instead of $P_1 = \{\{x_1, x_2, x_3\}, \{x_4, x_5, \}, \{x_6, x_7, x_8, x_9\}\}$ we can have $P_2 = \{\{x_1, x_2, x_3, x_4\}, \{x_5, \}, \{x_6, x_7, x_8, x_9\}\}$. We can have 21147 different partitions for the 9 elements in Figure 5 (B): this number is called the *Bell Number*, and it is recursively computed as $B_{n+1} = \sum_{k=0}^{n} \binom{n}{k} B_k$, with $B_0 = B_1 = 1$ [15].

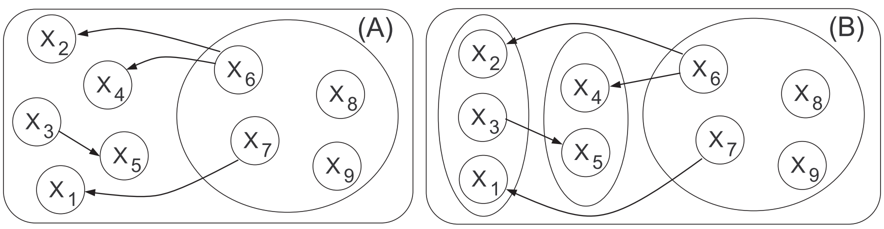

Figure 5. The graphical difference between a conflict-free extension (A) and a conflict-free partition (B).

In the following, we extend the definitions given in Section 2 in order to consider coalitions.

Definition 4.1. A partition of coalitions $\mathcal{G} = \{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_n\}$ is **conflict-free** iff for each $\mathcal{B}_i \in \mathcal{G}$, $\mathcal{B}_i$ is conflict-fre, i.e., $\forall a, b \in \mathcal{B}_i.(a, b) \notin R$: no attacking arguments inside the same coalition.

From the argumentation theory point of view, finding a conflict-free partition of coalitions corresponds to partitioning the arguments into coherent subsets, in order to find feasible lines of thought which do not internally attack themselves. Now we revise the concept of attack/defense among coalitions and arguments and the notion of stable partitions of coalitions:

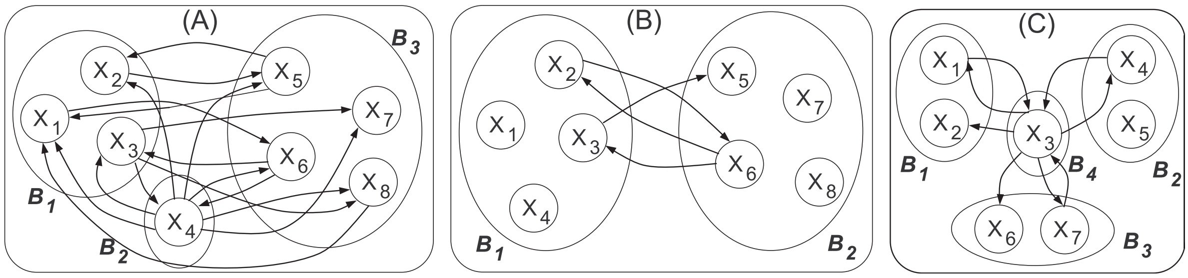

Figure 6. A stable (A), an admissible and complete (B), and an admissible but not complete (C) partition of arguments.

Definition 4.2. A coalition $\mathcal{B}_i$ **attacks** another coalition $\mathcal{B}_j$ if one of its elements attacks at least one element in $\mathcal{B}_j$, i.e., $\exists a \in \mathcal{B}_i, b \in \mathcal{B}_j$ s.t. $a\,R\,b$. $\mathcal{B}_i$ **defends** an attacked argument a, e.g., $b\,R\,a$, if $\exists c \in \mathcal{B}_i$ s.t. $c\,R\,b$.

Definition 4.3. A *conflict-free* partition $\mathcal{G} = \{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_n\}$ is **stable** iff for each coalition $\mathcal{B}_i \in \mathcal{G}$, all its elements $a \in \mathcal{B}_i$ are attacked by all the other coalitions $\mathcal{B}_j$ with $j \neq i$, i.e., $\forall a \in \mathcal{B}_i, \exists b \in \mathcal{B}_j . b\,R\,a$ $(\forall j \neq i)$.

Figure 6 (A) represents a stable partition: each argument in $\mathcal{B}_2$ (i.e., x_4) is attacked by at least one argument in $\mathcal{B}_1$ (i.e., x_3) and one argument in $\mathcal{B}_3$ (i.e., x_6), and the same holds also for the arguments in $\mathcal{B}_2$ and $\mathcal{B}_3$. To have a stable partition means that each of the arguments cannot be moved from one coalition to another without inducing a conflict in the new coalition. In the next two definitions we respectively extend the concept of admissible and complete extensions.

Definition 4.4. A *conflict-free* partition $\mathcal{G} = \{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_n\}$ of coalitions is **admissible** iff for each argument $a \in \mathcal{B}_i$ attacked by $b \in \mathcal{B}_j$ (i.e., $b\,R\,a$), then $\exists c \in \mathcal{B}_i$ that attacks $b \in \mathcal{B}_j$ (i.e., $c\,R\,b$), that is each $\mathcal{B}_i$ defends all its arguments.

According to Dung's definition of admissible extension, *"the set of all arguments accepted by a rational agent is a set of arguments which can defend itself against all attacks on it"* [20]. Notice that if only one argument a in the interaction graph has no grandparents, it is not possible to obtain even one admissible partition: no argument in $\mathcal{A}_{rgs}$ is able to defend a. In Definition 4.4, we have naturally extended the definition of admissible extension [20] to coalitions: since each coalition represents the line of thought of an agent, each rational agent is able to defend its line of thought because it counter-attacks all its attacking lines.

Figure 6 (B) represents an admissible partition as it is conflict-free and both $\mathcal{B}_1$ and $\mathcal{B}_2$ defend themselves: x_5 is defended by x_6, x_3 is defended by x_2, and, finally, x_2 and x_6 defend themselves.

Definition 4.5. An admissible partition $\mathcal{G} = \{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_n\}$ is a **complete** partition of coalitions iff each argument a which is defended by $\mathcal{B}_i$ is in $\mathcal{B}_i$ (i.e., $a \in \mathcal{B}_i$).

Figure 6 (B) is a complete partition because all the elements defended by $\mathcal{B}_2$ (i.e., x_5, x_6) belong to $\mathcal{B}_2$ and all the elements defended by $\mathcal{B}_1$ (i.e., x_2, x_3) belong to $\mathcal{B}_1$. Figure 6 (C) represents an admissible but not complete partition because x_6 is defended also by coalitions $\mathcal{B}_1$ (via x_1) and $\mathcal{B}_2$ (via x_4) but it only belongs to $\mathcal{B}_3$. Intuitively, the notion of complete partition captures all the rational agents who believe in every argument they can defend [20], possibly attacking the arguments of the other coalitions in the partition, i.e., the *line of thought* of the other agents.

In Theorem 4.6 we prove that each of the coalitions in every possible conflict-free partition is a conflict-free extension as defined by Dung [20]. Respectively, we can prove the same property also for admissible, complete and stable partitions.

Theorem 4.6. Given an AF $\langle \mathcal{A}_{rgs}, R \rangle$,

(a) given the set of all conflict-free extensions CFE, each CFP conflict-free partition (as defined in Definition 4.1) is a subset of them, i.e., $CFP \subseteq CFE$.

(b) given the set of all admissible extensions AE, each AP admissible partition (as defined in Definition 4.4) is a subset of them, i.e., $AP \subseteq AE$.

(c) given the set of all complete extensions CE, each CP complete partition (as defined in Definition 4.5) is a subset of them, i.e., $CP \subseteq CE$.

(d) given the set of all stable extensions SE, each SP stable partition (as defined in Definition 4.3) is a subset of them, i.e., $SP \subseteq SE$.

Proof:
Since all the coalitions of arguments $\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_n$ in the CFP partition named $\mathcal{G}$ are conflict-free extensions by definition (see Definition 4.1), that is $\mathcal{B}_1$ is conflict-free, $\mathcal{B}_2$ is conflict-free and so on, then $\mathcal{G}$ belongs to the set of all conflict-free extensions, i.e., $\mathcal{G} \subseteq CFE$. The same reasoning also holds for AP, CP and SP partitions. □

We can now define the hierarchy of the set inclusions among the proposed partitions like Dung has shown for set inclusions among classical extensions [20]:

Proposition 4.7. Given the $CFPS$ the set of all conflict-free partitions, and APS, CPS and SPS respectively the set of all admissible, complete and stable partitions, we have that $SPS \subseteq CPS \subseteq AS \subseteq CFPS$.

Proof:
From the theory behind AFs we have that $SE \subseteq CE \subseteq AE \subseteq CFE$ (see Section 2), i.e., stable extension is also a complete extension, which is also an admissible extension, which is, finally, also a conflict-free extension. Moreover, Theorem 4.6 proves that each CFP is composed by conflict-free extensions, each AP by admissible extensions, each CP by complete extensions and each SP by stable extensions. Therefore, the inclusions among all the CFP, AP, CP and SP is preserved. □

Note that it is possible that, given the same set of arguments, a stable (for example) extension exists, but, at the same time, a stable partition cannot be found over the same set of arguments. Let us consider the following example: $\mathcal{A}_{rgs} = \{a, b, c, d, e\}$ and $R = \{(b, c), (c, d), (d, e), (e, b)\}$ (i.e., a loop of attacks). According to Dung's stable semantics, this framework has two stable extensions: $\{a, b, d\}$ and $\{a, c, e\}$. However, this framework has no stable partition because argument a is not attacked by any argument and it cannot belong to two sets at the same time (it cannot be a partition). This is not a limitation: our goal is to simultaneously form distinct stable extensions within the same set of arguments, since our goal is to assign all these arguments to different agents.

5. Semiring-based Weighted Partitions

Weighted AFs extend Dung's AFs by adding weight values to every edge in the attack graph, intuitively corresponding to the strength of the attack, or equivalently, how reluctant we would be to disregard it [11, 21]. In this section we define a quantitative framework where attacks have an associated preference/weight and, consequently, also the computation of the coalitions as presented in this paper has an associated weight representing how much inconsistency we tolerate in the solution: more specifically, "how much conflict" we tolerate inside a conflict-free partition, which can now include attacking arguments in the same coalition. Modeling this kind of problems as SCSPs (see Section 3) leads to a partition that optimizes the criteria defined by the chosen semiring, which is used to mathematically represent the attack weights.

Figure 7 represents three contradictory weather forecasts of *BBC*, *CNN* and *Fox* with a weighted interaction graph:

a: Today will be dry in London since BBC forecasts sunshine
b: Today will be wet in London since CNN forecasts rain
c: Today will be dry in London since Fox forecasts sunshine

In this example $\mathcal{A}_{rgs} = \{a, b, c\}$, $a\,R\,b$ and $c\,R\,b$. In Figure 7 each of these two attack relationships is associated with a fuzzy weight (in $[0, 1]$) representing the strength of the attack: since BBC forecast is more reliable than Fox forecast, a attacks b with more strength, i.e., 0.5, than c attacks b, i.e., 0.9 (0 represents the strongest possible attack and 1 the weakest one).

Figure 7. A fuzzy Argumentation Framework with fuzzy scores modeling the attack strength.

Many other classical weighted AFs in literature can be modeled with semirings [11]. An argument can be seen as a chain of possible events that makes the hypothesis true. The credibility of a hypothesis can then be measured by the total probability that it is supported by arguments. The proper semiring to solve this problem consists in the *Probabilistic* semiring [5]: $\langle [0..1], max, \hat{\times}, 0, 1\rangle$, where the arithmetic multiplication (i.e., $\hat{\times}$) is used to compose the probability values together (assuming that the probabilities being composed are independent). Weights can be interpreted as subjective beliefs [21]. For example, a weight of $w \in (0, 1]$ on the attack of argument a_1 on argument a_2 might be understood as the belief that (a decision-maker considers) a_2 is false when a_1 is true. This belief could be modeled using probability [21].

The Fuzzy Argumentation [11] approach enriches the expressive power of the classical argumentation model by allowing to represent the relative strength of the attack relationships between arguments, as well as the degree to which arguments are accepted. Hence, the *Fuzzy* semiring $\langle[0..1], max, min, 0, 1\rangle$ can be used (e.g., in Figure 7). In addition, the *Weighted* semiring $\langle\mathbb{R}^+ \cup \{\infty\}, min, \hat{+}, \infty, 0\rangle$, where $\hat{+}$ is the arithmetic plus ($\mathbf{0} = \infty$ and $\mathbf{1} = 0$), can model the (e.g., monetary) cost of the attack: for example, the number of votes in support of the attack [21].

By using the *Boolean* semiring $\langle\{true, false\}, \vee, \wedge, false, true\rangle$ we can cast the classic AF originally defined by Dung [20] in the same semiring-based framework ($\mathbf{0} = false, \mathbf{1} = true$). In this case, if $W(a, b) = true$ then it means there is no attack between a and b. Definition 5.1 rephrases the notion of AF given by Dung (see Section 2) into *semiring-based AF*, i.e., an AF_S:

Definition 5.1. (Semiring-based Weighted Argumentation Frameworks [11])
A semiring-based Argumentation Framework (AF_S) is a quadruple $\langle\mathcal{A}_{rgs}, R, W, S\rangle$, where S is a semiring $\langle A, +, \times, \mathbf{0}, \mathbf{1}\rangle$, $\mathcal{A}_{rgs}$ is a set of arguments, R the attack binary relation on $\mathcal{A}_{rgs}$, and $W : \mathcal{A}_{rgs} \times \mathcal{A}_{rgs} \longrightarrow A$ a binary function called the *weight* function. Given $a, b \in \mathcal{A}_{rgs}$, $\forall(a, b) \in R$, $W(a, b) = s$ means that a attacks b with a strength level $s \in A$.

We suppose that a semiring value of $\mathbf{1}$ means that there is not attack between two arguments. In Definition 5.2 we redefine the notion of α-conflict-free partition: conflicts inside the same coalition can be now part of the solution until a cost threshold α is met, and not worse:

Definition 5.2. (α-conflict free property [11] extended to partitions)
Given a semiring-based AF_S, a partition of coalitions $\mathcal{G} = \{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_n\}$ is α-conflict-free for AF_S iff $\prod_{b,c \in \mathcal{B}_i} W(b, c) \geq_S \alpha$ (the $\prod$ uses the $\times$ of the semiring).

In Figure 8 there is an example of a 0.5-conflict-free partition using a *Fuzzy* semiring, i.e., the $\times$ used to compose the weights corresponds to min. Notice that only the attacks within the same coalition are considered: $min(0.6, 0.7, 0.5) = 0.5$.

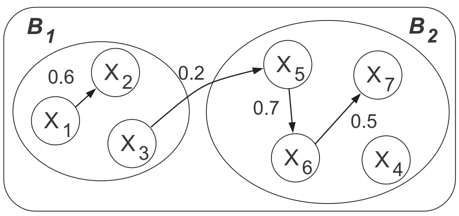

Figure 8. A 0.5-conflict-free partition by using the *Fuzzy* semiring, i.e., $min(0.6, 0.7, 0.5) = 0.5$. The attack between x_3 and x_5 is not considered since they belong to different coalitions.

Proposition 5.3. If a partition is α_1-conflict-free, then the same partition is also α_2-conflict-free if $\alpha_1 >_S \alpha_2$.

Proof:
From Definition 5.2 if we have that $\prod_{b,c \in \mathcal{B}_i} W(b,c) \geq_S \alpha_1$ and $\alpha_1 >_S \alpha_2$, then $\prod_{b,c \in \mathcal{B}_i} W(b,c) \geq_S \alpha_2$. □

For instance, in *Weighted* semirings a 3-conflict-free partitions is also 4-conflict-free. In Definition 5.4 we extend with weights also the other kinds of partitions.

Definition 5.4. Given an AF_S, a partition of coalitions $\mathcal{G} = \{\mathcal{B}_1, \mathcal{B}_2, \ldots, \mathcal{B}_n\}$ can be defined as α-stable (or α-admissible or α-complete) by replacing conflict-free partitions with α-conflict-free partitions in Definition 4.3 (or Definition 4.4 or Definition 4.5).

In Proposition 5.5 we relate the weighted partitions with not-weighted ones presented in Section 4.

Proposition 5.5. Iff a partition is $\mathbf{1}$-conflict-free (or $\mathbf{1}$-stable, $\mathbf{1}$-admissible, $\mathbf{1}$-complete), then the same partition is also conflict-free (or respectively, stable, admissible, complete) as shown in Section 4.

Proof:
Since we suppose that $W(a,b) = \mathbf{1}$ means there is no attack between a and b and we have that $\mathbf{1}$ is the unit element for $\times$ (see Section 3), then a partition is $\mathbf{1}$-conflict-free iff there are no attacks among the arguments of each of its coalitions, i.e., all the attacks are weighted with $\mathbf{1}$. □

6. Mapping Partition Problems to SCSPs

In this section we show a mapping from the AF_S extended to coalitions (see Section 5) to SCSPs (see Section 3), i.e., $\mathcal{M} : AF_S \rightarrow SCSP$. $\mathcal{M}$ is described as follows: given an AF_S as described in Section 5, we define a variable for each argument $a_i \in \mathcal{A}_{rgs}$, i.e., $V = \{a_1, a_2, \ldots, a_n\}$. The value of a variable represents the coalition to which argument a_i belongs: i.e., each variable domain is $D = \{1, n\}$. For example if $a_1 = 2$ it means that the first argument belongs to the second coalition. We can have a maximum of n coalitions, that is all singletons.

In the following explanation, b attacks a means that b is a parent of a in the interaction graph, and c attacks b attacks a means that c is a grandparent of a. For the following constraint classes we consider a $AF_S = \langle \mathcal{A}_{rgs}, R, W, S \rangle$ where $S = \langle A, +, \times, \mathbf{0}, \mathbf{1} \rangle$ and $s \in A$:

1. **Conflict-free constraints.** Since our goal is to find a α-conflict-free partition, if $a_i \, R \, a_j$ and $W(a_i, a_j) = s$ we need to assign a s preference to the solution that includes both a_i and a_j in the same coalition of the partition: $c_{a_i,a_j}(a_i = p, a_j = p) = s$ ($p \in [1..n]$). Otherwise $c_{a_i,a_j}(a_i = p, a_j = q) = \mathbf{1}$ (with $q \neq p$ and $q \in [1..n]$)). Using Figure 9 as example and considering only the conflict-constraints between x_1 and x_2, we have $c_{x_1,x_2}(x_1 = 1, x_2 = 1) = \mathbf{0}$, $c_{x_1,x_2}(x_1 = 2, x_2 = 2) = \mathbf{0} \ldots c_{x_1,x_2}(x_1 = 5, x_2 = 5) = \mathbf{0}$ (having 5 nodes, max 5 coalitions are possible), otherwise, $c_{x_1,x_2} = \mathbf{1}$ (i.e., the assignment satisfies the conflict-free constraints).

2. **Admissible constraints.** For the admissibility of a partition, if a_i has several grandparents $a_{g_1}, a_{g_2}, \ldots, a_{g_k}$ through the same parent a_f, we add the following $k+1$-ary constraint $c_{a_i,a_{g1},\ldots,a_{g_k}}(a_i = p, a_{g_1} = q_1, \ldots, a_{g_k} = q_k) = \mathbf{0}$ if $\forall q_i . q_i \neq h$ ($\mathbf{1}$ otherwise). This is because at least a grandparent must be taken in the same coalition, in order to defend a_i from his parent a_f. Notice

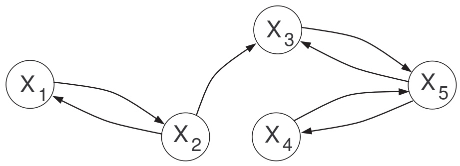

Figure 9. An example of an interaction graph.

that, if an argument is not attacked (i.e., he has no parents), it can be taken or not in any admissible set. Moreover, if a_i has a parent but no grandparents, it is not possible to find any admissible partition, that is the SCSP is inconsistent (see Section 3). With respect to Figure 9, considering only argument x_3 and its parent x_2, for example we have $c_{x_3,x_1}(x_3 = 1, x_1 = 1) = \mathbf{1}, c_{x_3,x_1}(x_3 = 2, x_1 = 2) = \mathbf{1} \ldots c_{x_3,x_1}(x_3 = 5, x_1 = 5) = \mathbf{1}$ ($\mathbf{0}$ for any other possible assignment of x_3 and x_1). This constraint clearly forces x_1 to be taken in the same coalition of x_3.

3. **Complete constraints.** To compute a complete extension $\mathcal{B}$, we impose that each argument a_i which is defended by $\mathcal{B}$ is in $\mathcal{B}$, except those a_i that, in such case, would be attacked by $\mathcal{B}$ itself [4]. This can be enforced by imposing that for each a_i taken in the coalition k, also all its $a_{s_1}, a_{s_2}, \ldots, a_{s_k}$ grandchildren (i.e., all the arguments defended by a_i), whose fathers are not taken in the same coalition, must be in $\mathcal{B}$. Formally, $c_{a_i,a_{s_1},\ldots,a_{s_k}}(a_i = p, a_{s_1} = p, \ldots, a_{s_k} = p) = \mathbf{1}$ only for those a_{s_i} for which it stands (premise of the constraint) that $a_{f_{s_1}} = q_1 \wedge a_{f_{s_2}} = q_2 \wedge \cdots \wedge a_{f_{s_z}} = q_z$, where $a_{f_{s_1}}, a_{f_{s_2}}, \ldots, a_{f_{s_z}}$ are fathers of a_{s_i} and $q_1, \ldots, q_z \neq p$, i.e., the fathers of a_{s_i} belong to a different coalition. With respect to Figure 9 and considering only argument x_1, we have constraints $c_{x_1,x_3}(x_1 = 1, x_3 = 1) = \mathbf{1}, c_{x_1,x_3}(x_1 = 2, x_3 = 2) = \mathbf{1}, \ldots, c_{x_1,x_3}(x_1 = 5, x_3 = 5) = \mathbf{1}$ ($\mathbf{0}$ otherwise).

4. **Stable constraints.** They can be represented with a constraint such that for each couple of arguments a_i, a_j belonging to two different coalitions, respectively r and s, at least one of the attacks to a_i has to come from an argument in coalition s: if $b_1, b_2, \ldots b_t$ are all the arguments that attack a_i, $c_{a_i,a_j,b_1,b_2,\ldots,b_n}(a_i = r, a_j = s, ((b_1 = s) \vee (b_2 = s) \vee \cdots \vee (b_n = s))) = \mathbf{1}$ ($\mathbf{0}$ otherwise); therefore, we model stable constraints with disjunctive constraints. Concerning the example in Figure 9, if we consider x_1 assigned to coalition 1 and x_5 to coalition 2 (i.e., the case of the stable partition P_1 in Tab. 1), we consequently add the constraint $c_{x_1,x_5,x_2}(x_1 = 1, x_5 = 2, (x_2 = 2)) = \mathbf{1}$ ($\mathbf{0}$ otherwise).

Notice that, in the mapping $\mathcal{M}$, only constraints describing the attacks between arguments are soft in the strictest sense, while the other ones return the values $\mathbf{0}$ or $\mathbf{1}$ in the semiring set, i.e., the corresponding variable assignment is respectively not admitted, or admitted. Other examples of argumentation-to-constraints mapping can be found in [11] (mappings with weights) and [12, 13] (mapping without weights), however not concerning coalitions.

Theorem 6.1. (Solution equivalence)
Given an $AF_S = \langle \mathcal{A}_{rgs}, R, W, S \rangle$, the solutions of the related SCSP obtained with the mapping $\mathcal{M}$ correspond to:

- all the α-conflict-free partitions of coalitions by using conflict-free constraints;

- all the α-admissible partitions by using admissible, and conflict-free;
- all the α-complete partitions by using complete, admissible and conflict-free constraints;
- all the α-stable partitions by using stable and conflict-free constraints.

Proof:
Given an AF_S named L, we can apply the mappings $\mathcal{M}(L) \rightarrow_{cf} P_{cf}$, $\mathcal{M}(L) \rightarrow_a P_a$, $\mathcal{M}(L) \rightarrow_c P_c$ and $\mathcal{M}(L) \rightarrow_s P_s$, respectively applying only the conflict-free (i.e., C_{cf}), admissible (i.e., C_a), complete (i.e., C_c) or stable (i.e., C_s) classes of constraints in the corresponding mapping. Then we have that $Sol(P_{cf}) = \bigotimes C_{cf}$, $Sol(P_a) = \bigotimes C_a$, $Sol(P_c) = \bigotimes C_c$ and $Sol(P_s) = \bigotimes C_s$. Since $blevel(P) = Sol(P) \Downarrow_\emptyset$ (see Section 3), then if $blevel(P_{cf}) = \alpha$, $blevel(P_a) = \alpha$, $blevel(P_c) = \alpha$ and $blevel(P_s) = \alpha$ we have respectively obtained the α-conflict-free, admissible, complete or stable partitions. □

Conflict-free, stable, admissible and complete partitions can be found by searching for $\mathbf{1}$-consistent solutions in the respective problems defined in Theorem 6.1, as defined by Proposition 5.5. Notice that finding $\mathbf{1}$-conflict-free partitions is equivalent to the well-known graph coloring problems, which has been deeply studied also from in constraint programming [28], and where no two adjacent vertices share the same color:

Proposition 6.2. The problem of finding a conflict-free partition of coalitions corresponds to finding a vertex-coloring partition of a graph [28], where each node of the same color belongs to the same coalition in a $\mathbf{1}$-conflict-free partition. The minimum number of colors needed to solve the problem corresponds to the minimum number of coalitions in a possible partition.

Proof:
By construction of the problem, an attack relationship with a weight of $\mathbf{0}$ represents that two nodes are neighbor nodes, while a weight of $\mathbf{1}$ means that they are not. The solutions of this problem groups in the same coalition those nodes who can share the same color. □

In Figure 9 we can see an example of classical (i.e., the attacks are not weighted) interaction graph. Only for this example we have 15 conflict-free partitions, as reported in Table 1. Among these conflict-free partitions, P_1, P_2, P_3, P_4, P_5 are also admissible partitions and P_1 is also the only one complete partition, and the only one stable partition as well; these partitions have been obtained with the implementation in Section 7.

Table 1. The list of all the conflict-free partitions of coalitions for the example in Figure 9.

$P_1 = \{\{x_1, x_3, x_4\}, \{x_2, x_5\}\}$, $P_2 = \{\{x_1, x_3, x_4\}, \{x_2\}, \{x_5\}\}$, $P_3 = \{\{x_1, x_3\}, \{x_2, x_4\}, \{x_5\}\}$, $P_4 = \{\{x_1, x_3\}, \{x_2, x_5\}, \{x_4\}\}$, $P_5 = \{\{x_1, x_3\}, \{x_2\}, \{x_4\}, \{x_5\}\}$, $P_6 = \{\{x_1, x_4\}, \{x_2, x_5\}, \{x_3\}\}$, $P_7 = \{\{x_1, x_4\}, \{x_2\}, \{x_3\}, \{x_5\}\}$, $P_8 = \{\{x_1, x_5\}, \{x_2, x_4\}, \{x_3\}\}$, $P_9 = \{\{x_1\}, \{x_2, x_4\}, \{x_3\}, \{x_5\}\}$, $P_{10} = \{\{x_1, x_5\}, \{x_2\}, \{x_3, x_4\}\}$, $P_{11} = \{\{x_1, \}, \{x_2, x_5\}, \{x_3, x_4\}\}$, $P_{12} = \{\{x_1\}, \{x_2\}, \{x_3, x_4\}, \{x_5\}\}$, $P_{13} = \{\{x_1, x_5\}, \{x_2\}, \{x_3\}, \{x_4\}\}$, $P_{14} = \{\{x_1\}, \{x_2, x_5\}, \{x_3\}, \{x_4\}\}$, $P_{15} = \{\{x_1\}, \{x_2\}, \{x_3\}, \{x_4\}, \{x_5\}\}$

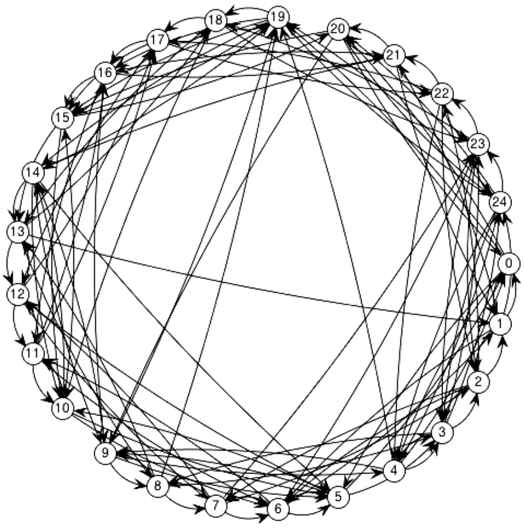

Figure 10. A small-world network with 25 nodes generated with JUNG by using the *KleinbergSmallWorldGenerator* class [27, 24].

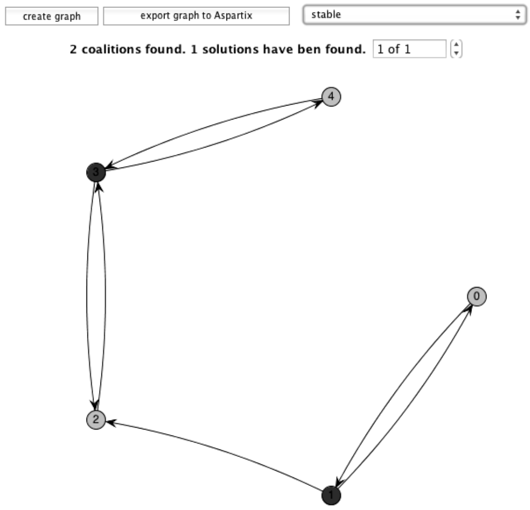

Figure 11. The *ConArg* window showing the stable solution P_1 in Table 1: the nodes of the two coalitions are in dark/light grey respectively.

7. Implementation in JaCoP

The *Java Constraint Programming* solver (JaCoP) is a Java library which provides Java user with *Finite Domain Constraint Programming* paradigm [28]. JaCoP provides different type of constraints: most commonly used primitive constraints, such as arithmetical constraints, equalities and inequalities, logical, reified and conditional constraints, combinatorial (global) constraints.

To develop in practice and test our model, we adopted the *Java Universal Network/Graph Framework* (*JUNG*) [27], a software library for the modeling, generation, analysis and visualization of graphs. Interaction graphs, where nodes are arguments and edges are attacks (see Section 2), clearly represent a kind of social network and consequently show the related properties [14]. Therefore, for the following tests we used the *KleinbergSmallWorldGenerator* class [27, 24] in JUNG, which randomly generates a $m \times n$ lattice with small-world properties [24]. In the implementation provided by JUNG [27], each node u has four local connections, one to each of its neighbors, and in addition one or more long range connections to some node v, where v is chosen randomly according to probability proportional to d^θ where d is the lattice distance between u and v and θ is the clustering exponent. An example of such random graphs with 25 nodes is shown in Figure 10.

In this first implementation we decided to only implement $\mathbf{1}$-conflict-free partitions, that is we do not consider weights on attacks; therefore, we only need crisp constraints, as offered by JaCoP. With this tool we can immediately check if a given partition is conflict-free, admissible, complete or stable. Moreover, we can exhaustively generate the partitions with such given properties: since the problem is $O(n^n)$ [26] (where n is the number of arguments), we limit the implementation to a partial search. In particular, we used the *Limited Discrepancy Search* (*LDS*) offered by JaCoP, which is a kind of *Depth First Search* procedure exploiting the method proposed in [22]: simplifying, If a predetermined number of different decisions is exhausted along a search path, then backtracking is initiated [22]. At each branching point during the search, we select the variable which has most constraints assigned to it, and we try the middle value from its current domain. Moreover, we set a timeout of 60 seconds to interrupt the search procedure, and to report the number of solutions found only in that interval.

We ran our experiments over 3 different randomly-generated *Kleinberg* graphs with 9, 25 and 100 nodes respectively, whose properties have been previously described in this section. The results are shown in Table 2: the table reports the number of edges/attacks, the numbers of found conflict-free (i.e., *CFPS*) and stable (i.e., *SPS*) partitions, which superiorly/inferiorly limit the number of admissible and complete partitions (as defined in Proposition 4.7). Moreover, the table reports the number of constraints used to model the problem and the measured maximum depth of the search tree, explored using branch-and-bound techniques limited by LDS principles [22]. Notice that, within the 60 seconds timeout, the adopted partial search is able to find only one stable partition for the graph with 100 nodes; for the same reason, that are timeout in conjunction with the depth of the search tree, the reported number of conflict-free solutions in Table 2 is less in the 100-nodes graph than in the 25-nodes one.

Table 2. The test on three different small-world graphs and the related statistics: CFPS and SPS respectively are all the found conflict-free and stable partitions.

Nodes	Attacks	CFPS	SPS	Constraints	Max. Search Depth
9	45	123	8	∼220	11
25	125	495984	119543	∼1440	61
100	500	92562	1	∼20600	218

Notice also that, in order to prevent symmetrically equivalent solutions, we have also implemented symmetry breaking constraints for graph coloring (see Proposition 6.2 for the analogies): any value permutation is a value symmetry in the coalition assignment of arguments. Performance in Table 2 have been retrieved by using a tool we recently developed, named *ConArg* [12, 13].[3]

In Figure 11 we show the only stable solution for the graph reported in Figure 9, that is P_1 in Table 1. Figure 11 reports a screenshot directly taken from *ConArg*. These results can be obtained by importing the graph in Figure 9 as textual input. All the found solutions can be browsed by using the graphical interface: the nodes belonging to the same coalition are highlighted with the same color.

8. Related Work and Comparison

The classical Dung's framework has been extended by Amgoud in [1] with a preference relation between elements; more in detail, Amgoud [1] provides the semantics (conflict-free, stable and preferred ones) of a coalition structure and a proof theory for testing whether a coalition is in the set of acceptable coalitions. An application of the model is also provided for the problem of task allocation among partitions of autonomous agents. With respect to the work in this paper, the view in [1] is not focused on generating partitions of arguments, but on directly checking the property of already given coalitions structures. Furthermore, [1] has no implementation to practically find solutions, as we instead perform in Section 7. Moreover, the method to compute the weights of coalitions is not quantitative (but it is only qualitative) and parametric, as we are alternatively able to represent with semirings.

[3]The tool is downloadable from `https://sites.google.com/site/santinifrancesco/tools/ConArg.zip`

In [17] an extension of the *Alternating-time Temporal Logic* (*ATL*) for modeling coalitions through argumentation is presented: a merge between ATL and the coalitional framework is obtained in order to express that agents are able to form a coalition which can successfully achieve a given property; the notions of defence and conflict-free are defined in terms of defeat rather than attack and preferences of arguments are given in a qualitative way (instead of quantitative as in our paper); to compute the desired classes of coalitions a model checker can be used; however, with such techniques, exponential complexity can be hardly faced while constraint programming provides a lot of techniques to tackle combinatorial problems [28].

In [14], social viewpoints (a model for goal based reasoning) are used to argue about coalitions in argumentation theory. The attack relation is based on the goal that agents have to achieve, that is, a coalition attacks another coalition if they share the same goal; this work does not provide a computational framework and only qualitative preferences over arguments are considered.

In [18], the authors see coalitions as conflict-free sets of arguments supporting each other. They start from a bipolar argumentation framework that extends Dung [20] with a support relation, and they come out to a meta-level representation of arguments (i.e., coalitions) in such a way to reuse Dung's principles and algorithms. Comparing [18] with our work, first, our solutions does not takes into account bipolar preferences, even if this bipolarity extension would be rather easy to cast in the same framework [10]; secondly, we use weights to model different strength values for attacks, while weights/preferences are out of the scope of the work in [18].

Eventually, our notion of coalition is completely different from the one presented in [18], since the latter exploits the support relationship (we have not), and also because we require a partition of the whole set of arguments into different coalitions with the same constraints (e.g., all admissible coalitions). In [18] arguments are not partitioned (arguments belong to "at least" one coalition), and *i)* coalitions consist in conflict-free sets of arguments (therefore, not considering admissibility constraints, for instance), *ii)* whose support sub-graph is connected, and *iii)* maximality considering *i)* and *ii)* is required.

In [16] the authors introduce an acceptability function which states when an argument is accepted with regard to the acceptability of a subset of arguments. The acceptability function establishes constraints on the acceptability of the arguments. Moreover, they introduces also weights represented mainly from the qualitative point of view. The main difference consists in the fact that in [16] no definition of coalition is provided, which consists in the core of our proposal instead. Moreover, our approach for attack weights is more quantitative than qualitative, since we adopt a semiring-like value structure (see Section 3). In addition, our solutions is parametrical with the chosen system of preferences.

In [25] the authors associate fuzzy degrees of acceptability with arguments in the context of games. They take advantage of probability theory to compute these degrees, which are the outcome of Game Theory. As a comparison, our system of preference is constrained to be a semiring structure: this does not allow the freedom to define acceptability degree functions as in [25], but it is parametrical and comes with several sound algebraic-properties.

In [2], arguments have an associated strength that influences the acceptability of the arguments they attack. Even this approach is mainly qualitative, while ours is quantitative and more from the "computational" point of view, since we offer a framework where we can use different aggregation functions for attack strength-values (as long they can be modelled as a semiring). Moreover, coalitions are not mentioned in [2].

At last, in [19] the proposed framework may be regarded as a belief revision model, based on argumentation. An agent interacts with the world by receiving arguments from one or more sources. The agent's internal mental state is completely described by a fuzzy set of trustful arguments, from which the beliefs of the agent may be derived. In addition to the fact that coalitions are not treated, our computational framework appears to be more comprehensive and less specialised for that particular case. Notice that the semiring-based framework has been already adopted to model trust-related problems, where semiring values can be used to compute a trust score between the entities in the relationship graph [7], or to define access policies [8].

In [11], some of the authors of this paper propose a computational framework where attacks have an associated weight to represent how much inconsistency can be tolerated in the final solution. This paper extends [11] by considering partitions of arguments and providing an implementation with JaCoP (no implementation is given in [11]), with tests on small-world graphs. Partitions of arguments implies redefining all the theory concepts w.r.t [11], e.g., stability or admissibility.

9. Conclusions

We extended classical argumentation frameworks of [20] to the problem of forming coalitions of arguments, partitioning all the arguments of a given starting set. We have redefined the classical definitions of Dung's extensions (conflict-free, admissible, stable and complete ones) in order to consider a partition of all the arguments into multiple coalitions, and modelled the problem of finding such coalitions with SCSPs [9, 5, 28]: this semiring-based formalism can be used to relax the concept of conflict-free partitions in order to allow some inconsistency (i.e., attacks) within the same coalition. The proposed quantitative framework can be used also to solve classical (i.e., crisp) CSPs. We have also solved a problem example considering only **1**-solutions with JaCoP and then we performed tests on a small-world network randomly generated with [27]. Starting from a single set of arguments, the goal has been to partition it into multiple coalitions with the same features (e.g., stability or admissibility) without discarding any argument.

In the future we want to improve the performance obtained in Section 7 by testing different solvers and constraint techniques (e.g., by taking the inspiration from [26]). Moreover, other kinds of networks can be used to execute the experiments, in order to catch different graph properties.

References

[1] Amgoud, L.: An Argumentation-Based Model for Reasoning About Coalition Structures, *ArgMAS05*, 4049, Springer, 2005, ISBN 3-540-36355-6.

[2] Amgoud, L., Prade, H.: Using arguments for making and explaining decisions, *Artif. Intell.*, **173**(3-4), 2009, 413–436.

[3] Apt, K. R., Witzel, A.: A Generic Approach to Coalition Formation, *CoRR*, **abs/0709.0435**, 2007.

[4] Besnard, P., Doutre, S.: Checking the acceptability of a set of arguments, *Workshop on Non-Monotonic Reasoning*, 2004.

[5] Bistarelli, S.: *Semirings for Soft Constraint Solving and Programming*, vol. 2962 of *LNCS*, Springer, 2004, ISBN 3-540-21181-0.

[6] Bistarelli, S., Campli, P., Santini, F.: Finding partitions of arguments with Dung's properties via SCSPs, *SAC* (W. C. Chu, W. E. Wong, M. J. Palakal, C.-C. Hung, Eds.), ACM, 2011, ISBN 978-1-4503-0113-8.

[7] Bistarelli, S., Foley, S., O'Sullivan, B., Santini, F.: Semiring-based frameworks for trust propagation in small-world networks and coalition formation criteria, *Security and Communication Networks*, **3**(6), 2010, 595–610, ISSN 1939-0122.

[8] Bistarelli, S., Martinelli, F., Santini, F.: A Semantic Foundation for Trust Management Languages with Weights: An Application to the RT Family, *Autonomic and Trusted Computing, 5th International Conference*, 5060, Springer, 2008.

[9] Bistarelli, S., Montanari, U., Rossi, F.: Semiring-based Constraint Solving and Optimization, *Journal of the ACM*, **44**(2), March 1997, 201–236.

[10] Bistarelli, S., Pini, M. S., Rossi, F., Venable, K. B.: From soft constraints to bipolar preferences: modelling framework and solving issues, *J. Exp. Theor. Artif. Intell.*, **22**(2), 2010, 135–158.

[11] Bistarelli, S., Santini, F.: A Common Computational Framework for Semiring-based Argumentation Systems, *ECAI'10*, 215, IOS Press, 2010, ISBN 978-1-60750-605-8.

[12] Bistarelli, S., Santini, F.: ConArg: A Constraint-Based Computational Framework for Argumentation Systems, *ICTAI*, IEEE, 2011, ISBN 978-1-4577-2068-0.

[13] Bistarelli, S., Santini, F.: Modeling and Solving AFs with a Constraint-Based Tool: ConArg, *TAFA* (S. Modgil, N. Oren, F. Toni, Eds.), 7132, Springer, 2011, ISBN 978-3-642-29183-8.

[14] Boella, G., van der Torre, L., Villata, S.: Social Viewpoints for Arguing about Coalitions, *PRIMA*, 5357, Springer, 2008.

[15] Bogart, K. P.: *Introductory Combinatorics*, Academic Press, Inc., Orlando, FL, USA, 2000, ISBN 0121108309.

[16] Brewka, G., Woltran, S.: Abstract Dialectical Frameworks, *KR* (F. Lin, U. Sattler, M. Truszczynski, Eds.), AAAI Press, 2010.

[17] Bulling, N., Dix, J.: Modelling and Verifying Coalitions using Argumentation and ATL, *Inteligencia Artificial, Revista Iberoamericana de Inteligencia Artificial*, **14**(46), 2010, 45–73.

[18] Cayrol, C., Lagasquie-Schiex, M.-C.: Coalitions of arguments: A tool for handling bipolar argumentation frameworks, *Int. J. Intell. Syst.*, **25**(1), January 2010, 83–109, ISSN 0884-8173.

[19] da Costa Pereira, C., Tettamanzi, A., Villata, S.: Changing Ones Mind: Erase or Rewind?, *IJCAI* (T. Walsh, Ed.), IJCAI/AAAI, 2011, ISBN 978-1-57735-516-8.

[20] Dung, P. M.: On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, *Artif. Intell.*, **77**(2), 1995, 321–357, ISSN 0004-3702.

[21] Dunne, P. E., Hunter, A., McBurney, P., Parsons, S., Wooldridge, M.: Weighted argument systems: Basic definitions, algorithms, and complexity results, *Artif. Intell.*, **175**(2), 2011, 457–486.

[22] Harvey, W. D., Ginsberg, M. L.: Limited Discrepancy Search, *IJCAI (1)*, 1995.

[23] Horling, B., Lesser, V.: A survey of multi-agent organizational paradigms, *Knowl. Eng. Rev.*, **19**(4), 2004, 281–316, ISSN 0269-8889.

[24] Kleinberg, J.: Navigation in a Small World, *Nature*, **406**, 2000, 845, ISSN 0269-8889.

[25] Matt, P.-A., Toni, F.: A Game-Theoretic Measure of Argument Strength for Abstract Argumentation, *Proceedings of the 11th European conference on Logics in Artificial Intelligence*, JELIA '08, Springer-Verlag, Berlin, Heidelberg, 2008, ISBN 978-3-540-87802-5.

[26] Ohta, N., Conitzer, V., Ichimura, R., Sakurai, Y., Iwasaki, A., Yokoo, M.: Coalition Structure Generation Utilizing Compact Characteristic Function Representations, *CP*, 5732, Springer, 2009.

[27] O'Madadhain, J., Fisher, D., White, S., Boey, Y.: *The JUNG (Java Universal Network/Graph) framework*, Technical report, UC Irvine, 2003.

[28] Rossi, F., van Beek, P., Walsh, T.: *Handbook of Constraint Programming*, Elsevier Science Inc., NY, USA, 2006, ISBN 0444527265.

[29] Shehory, O., Kraus, S.: Task Allocation Via Coalition Formation Among Autonomous Agents, *IJCAI (1)*, 1995.