On using QUBO to enforce extensions in abstract argumentation

Marco Baioletti[1,*], Francesco Santini[1,†]

[1]Dipartimento di Matematica e Informatica, Università degli Studi di Perugia, Perugia, Italy

Abstract

This paper summarizes our first discoveries in encoding *Abstract Argumentation* problems as *Quadratic Unconstrained Binary Optimization problems.* Classical and quantum annealers can then solve such formulations. In particular, we focus here on an optimization approach to extension enforcement, where a set of arguments is requested to satisfy a given semantics through possible changes of the considered Abstract Argumentation framework.

Keywords

Argumentation, Quadratic Unconstrained Binary Optimization, Extension enforcing

1. Introduction

Abstract Argumentation [1] is powerful enough to model a whole range of formalisms in logics and *nonmonotonic reasoning* [2] in particular. In its original formulation, an *Argumentation Framework* (*AF*) [1] is simply a directed graph in which the arguments are represented as nodes, and the arrows represent an attack relation. Arguments are "abstract" in that their structure - usually in the form of some premises leading to a claim - is not considered: for example, "[claim] The death penalty should be abolished since [premise] it legitimizes an irreversible act of violence" is condensed to a simple graph node attacking, for example, "The US Supreme Court has upheld the death penalty as constitutional" (here the claim is implied).

An *extension* is a set of arguments in an AF that can survive the conflict together and thus collectively represent a reasonable position to be taken during an e.g., decision-making process involving those arguments. Several problems in the literature consider extension-related problems that are computationally hard to solve and are consequently of interest to Artificial Intelligence. The *Quadratic Unconstrained Binary Optimization* problem (*QUBO*) [3] has become a unifying model for representing a wide range of combinatorial optimization problems and linking various disciplines that face these problems. QUBO problems are NP-complete, and a vast literature is dedicated to approximate solvers based on heuristics or meta-heuristics,

AIQxQIA International Workshop on AI for Quantum and Quantum for AI, AIxIA, Rome

*Corresponding author.

†The author has been supported by Project "Empowering Public Interest Communication with Argumentation (EPICA)", (MUR PRIN); Projects FICO and AIDMIX ("Fondo Ricerca di Ateneo", University of Perugia, year 2021); Project VITALITY, CUP J97G22000170005 (NRRP-MUR).

marco.baioletti@unipg.it (M. Baioletti); francesco.santini@unipg.it (F. Santini)

0000-0001-5630-7173 (M. Baioletti); 0000-0002-3935-4696 (F. Santini)

such as *simulated annealing* approaches (*SA*), *tabu-serch*, *genetic algorithms* or *evolutionary computing* [4]. *Quantum* annealers and Fujitsu's *digital annealers*[1] can be used to find global minima by using quantum *fluctuations*. QUBO models are at the heart of experimentation with quantum computers built by D-Wave Systems.[2]

In this paper, we focus on the problem of enforcing an extension: an agent may be interested in determining arguments to be added to enforce the set of arguments they prefer. Such enforcement is helpful in many scenarios, particularly when considering an argumentative debate between several agents because the arrival of new arguments in the debate typically questions the existing extensions. However, in many other situations, no new arguments are available to explain the change, and one is forced to evaluate the arguments' attacks on one another. Several of these problems have been proven NP-complete [5]; hence, they represent a perfect setting for proposing an encoding to QUBO as we advance in this work.

Some preliminary results on Abstract Argumentation and QUBO have been proposed in [6]. Section 2 reports necessary background information on Argumentation and QUBO, then Sect. 3 describes the encoding of extension-enforcing into QUBO. Section 4 highlights the interest in approximate solvers in Abstract Argumentation by introducing a dedicated international competition. While QUBO problems can also be solved with classical annealers, our ultimate interest is represented by using quantum annealers (as tested in [6]), or more in general, to bring Argumentation-related problems to the quantum world. Finally, Sect. 5 wraps up the paper with final thoughts and ideas about possible future work.

2. Background

This section summarizes the necessary background information regarding Abstract Argumentation and QUBO.

2.1. Abstract Argumentation

An *Abstract Argumentation Framework* (AF, for short) [1] is a tuple $\mathcal{F} = (\mathsf{A}, R)$ where A is a set of arguments and R is a relation $R \subseteq \mathsf{A} \times \mathsf{A}$. For two arguments $a, b \in \mathsf{A}$, the relation aRb means that argument a *attacks* argument b. An argument $a \in \mathsf{A}$ is *defended* by $S \subseteq \mathsf{A}$ (in $\mathcal{F}$) if for each $b \in \mathsf{A}$ such that bRa there is some $c \in S$ such that cRb. A set $E \subseteq \mathsf{A}$ is *conflict-free* (**cf** in $\mathcal{F}$) if and only if there are no $a, b \in E$ with aRb. E is *admissible* (**ad** in $\mathcal{F}$) if and only if it is conflict-free and each $a \in E$ is defended by E. Finally, the range of E in $\mathcal{F}$, i.e., $E^+_\mathcal{F}$, collects the same E and the set of arguments attacked by E: $E^+_\mathcal{F} = E \cup \{a \in \mathsf{A} \mid \exists b \in E : bRa\}$.

The *collective acceptability* of arguments depends on the definition of different *semantics* [1]. Semantics determine sets of jointly acceptable arguments, called *extensions*, by mapping each $\mathcal{F} = (\mathsf{A}, R)$ to a set $\sigma(\mathcal{F}) \subseteq 2^\mathsf{A}$, where 2^A is the power set of A, and σ parametrically stands for any of the considered semantics. The extensions under complete (**co**), preferred (**pr**), semi-stable (**sst**), stable (**st**), and grounded (**gr**) semantics are defined as follows; given $\mathcal{F} = (\mathsf{A}, R)$ and a set $E \subseteq \mathsf{A}$:

[1]Fujitsu's digital annealer: https://www.fujitsu.com/global/services/business-services/digital-annealer/.
[2]D-Wave website: https://www.dwavesys.com.

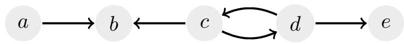

Figure 1: An example of WAAF.

	Ver-σ	DC-σ	DS-σ	Ex-σ	NE-σ
Conflict-free	in L	in L	triv.	triv.	in L
Admissible	in L	**NP-c**	triv.	triv.	**NP-c**
Complete	in L	**NP-c**	P-c	triv.	**NP-c**
Preferred	coNP-c	**NP-c**	$\prod_2^P$-c	triv.	**NP-c**
Semi-stable	coNP-c	$\sum_2^P$-c	$\prod_2^P$-c	triv.	**NP-c**
Stable	in L	**NP-c**	coNP-c	**NP-c**	**NP-c**
Grounded	P-c	P-c	P-c	triv.	P-c

Table 1
The complexity of some problems in Abstract Argumentation.

- $E \in \mathbf{co}(\mathcal{F})$ iff E is admissible in $\mathcal{F}$ and if $a \in \mathsf{A}$ is defended by E in $\mathcal{F}$ then $a \in E$,
- $E \in \mathbf{pr}(\mathcal{F})$ iff $E \in \mathbf{co}(\mathcal{F})$ and there is no $E' \in \mathbf{co}(\mathcal{F})$ s.t. $E' \supset E$,
- $E \in \mathbf{sst}(\mathcal{F})$ iff $E \in \mathbf{co}(\mathcal{F})$ and there is no $E' \in \mathbf{co}(\mathcal{F})$ s.t. $E'^{+}_{\mathcal{F}} \supset E^{+}_{\mathcal{F}}$,
- $E \in \mathbf{st}(\mathcal{F})$ iff $E \in \mathbf{co}(\mathcal{F})$ and $E^{+}_{\mathcal{F}} = \mathsf{A}$,
- $E \in \mathbf{gr}(\mathcal{F})$ iff $E \in \mathbf{co}(\mathcal{F})$ and there is no $E' \in \mathbf{co}(\mathcal{F})$ s.t. $E' \subset E$.

Figure 1 shows an AF with five arguments and five attacks. Given $\mathcal{F}$, the set of complete extensions is $\mathbf{co}(\mathcal{F}) = \{\{a\}, \{a, d\}, \{a, c, e\}\}$, while $\mathbf{st}(\mathcal{F}) = \{\{a, d\}, \{a, c, e\}\}$ is the set of stable extensions, for example.

We now report the definition of five well-known decision problems in Abstract Argumentation, whose complexity parametrized for a given semantics is shown in Tab. 1.

- *Verification of an extension* ($\mathbf{VER}$-σ): given $\mathcal{F} = (\mathsf{A}, R)$ and a set of arguments $E \subseteq \mathsf{A}$, is $E \in \sigma(\mathcal{F})$?
- *Credulous acceptance of an argument* ($\mathbf{DC}$-σ): given $\mathcal{F} = (\mathsf{A}, R)$ and an argument $a \in \mathsf{A}$, is a contained in some $E \in \sigma(\mathcal{F})$?
- *Skeptical acceptance of an argument* ($\mathbf{DS}$-σ): given $\mathcal{F} = (\mathsf{A}, R)$ and an argument $a \in \mathsf{A}$, is a contained in all $E \in \sigma(\mathcal{F})$?
- *Existence of an extension* ($\mathbf{EX}$-σ): given $\mathcal{F} = (\mathsf{A}, R)$, is $\sigma(\mathcal{F}) \neq \varnothing$?
- *Existence of non-empty extension* ($\mathbf{NE}$-σ): given $\mathcal{F} = (\mathsf{A}, R)$, does there exist $E \neq \varnothing$ such that $E \in \sigma(\mathcal{F})$?

In addition, the work in [7] presents the task of *extension enforcement*: we consider the objective to change the attack relationship R of a framework $\mathcal{F} = (\mathsf{A}, R)$ such that a given set $T \subseteq \mathsf{A}$ becomes (a subset of) an extension under a given semantics σ. In this case, the enforcement is *argument-fixed*, since only the attack relationship can be modified. *Strict enforcement* is satisfied if T is a σ-extension, while in *non-strict enforcement* T is only required to be a subset of a σ-extension. If we consider the Hamming distance of the changes, i.e., $|R \Delta R'| = |R \setminus R'| + |R' \setminus R|$, in [7] the authors impose a threshold $|R \Delta R'| \leq k$ as a further parameter of these problems. The complexity of some of these problems is reported in Tab. 2.

σ	strict	non-strict
Admissible	P	**NP-c**
Complete	**NP-c**	**NP-c**
Preferred	$\sum_2^P$-c	**NP-c**
Stable	P	**NP-c**
Grounded	**NP-c**	**NP-c**

Table 2
The complexity of extension enforcement [5].

In this paper, as proposed in [5], we look at the problem from an optimization point of view:

Definition 1 ([5])**.** *Given $\mathcal{F} = (\mathsf{A}, R)$, $T \subseteq \mathsf{A}$, and semantics σ, strict extension enforcement is an optimization problem where the goal is to find $\mathcal{F}^* = (\mathsf{A}, R^*)$ s.t.:*

$$R^* \in \underset{R' \in enfst(\mathcal{F},T,\sigma)}{\operatorname{argmin}} |R \Delta R'|$$

where $enfst(\mathcal{F}, T, \sigma) = \{R' | \mathcal{F}' = (\mathsf{A}, R'), T \in \sigma(\mathcal{F}')\}$. Similarly, we can define the same problem by considering non-strict enforcement (by defining $enfnst$).

2.2. Quadratic Unconstrained Binary Optimization

Quadratic Unconstrained Binary Optimization (in short, QUBO) [8] is an important form of optimization problems which has recently gained great popularity because of fast solvers and dedicated computing devices, such as quantum and digital annealers. Hence, several optimization problems, in a large range of application domains, have been formulated as QUBO problems, to be solved by these new methods [8, 9].

QUBO has been intensively investigated and is used to characterize and solve many optimization problems. For example, it encompasses SAT Problems, Constraint Satisfaction Problems, Maximum Cut Problems, Graph Coloring Problems, Maximum Clique Problems, General 0/1 Programming Problems and many more [8]. There exist QUBO embeddings also for Support Vector Machines, Clustering algorithms, and Markov Random Fields [10].

A QUBO problem is defined in terms of n binary variables $x_1, \ldots, x_n$ and a $n \times n$ upper-diagonal matrix $\mathcal{Q}$ and consists in minimizing the function

$$f(x) = \sum_{i=1}^{n} Q_{i,i} x_i + \sum_{i<j}^{n} Q_{i,j} x_i x_j.$$

The diagonal terms $Q_{i,i}$ are the linear coefficients and the non-zero off-diagonal terms $Q_{i,j}$ are the quadratic coefficients. This can be expressed more concisely as

$$\min_{x \in \{0,1\}^n} x^T Q x$$

where x^T denotes the transpose of the vector x.

The formulation of a discrete constrained optimization problem as QUBO requires the following steps: *i)* find a binary representation for the solutions, *ii)* define a penalization function, which penalizes unfeasible solutions (i.e., violating a constraint).

3. QUBO Encodings

In [6], we proposed for the first time an encoding of two well-known NP-complete problems in Abstract Argumentation as QUBO problems: **DC**-σ and **Exists**-$\sigma^{\neg\emptyset}$, while the considered semantics was only **co**. Moreover, in [6], we solved this problem on some frameworks by directly implementing them using the D-Wave Ocean SDK. We used a SA algorithm and a real quantum annealer provided by the *LeapTM Quantum Cloud Service*.[3]

Concerning [6], by continuing on this research line, we have extended the encoding to all classical NP-complete problems highlighted in bold in Tab. 1. Moreover, we have empirically validated all the encodings by comparing the obtained results with the simulated annealing algorithm against *ConArg* [11], an exact solver using *Constraint Programming*.

3.1. Encoding Complete Extensions in QUBO

In this section, we describe the basics of the encodings that lead to the model of the complete semantics, which will be at the core of enforcing complete extensions in Sect. 3.2.

We first assign to each argument an index, hence $\mathsf{A} = \{a_1, \ldots, a_n\}$, where n is the number of arguments. We use a set of n binary variables $x_1, \ldots, x_n$ to represent a set E of arguments: $a_i \in E$ if and only if $x_i = 1$. We denote by $\underline{x}$ the tuple $(x_1, \ldots, x_n)$ and by $\mathbf{x} \in \{0,1\}^n$ a vector of possible values for $x_1, \ldots, x_n$. Each semantics σ will be associated with a quadratic penalization function (or *Pfunction* for short) P_σ such that P_σ assumes its minimum value at $\mathbf{x}$ if and only if the corresponding set $E = \{a_i \in \mathsf{A} : x_i = 1\}$ is an extension valid for σ.

Most of the argumentation semantics require admissible sets. Hence, we define a Pfunction P_{adm}, which enforces this property. P_{adm} is the sum of four terms and contains new additional variables. The first term forces the set E to be **conflict-free**: $P_{cf} = \sum_{iRj \text{ or } jRi} x_i x_j$. In fact, the value of P_{cf} corresponds to the number of self attacks in E and its value is 0 if and only if E is conflict-free.

The constraints to model the notion of **defense** are more complicated: we use a first set of additional variables $t_1, \ldots, t_n$, denoting which arguments are attacked by E: $t_i = 1$ if and only if some argument of E attacks a_i. The variables $d_1, \ldots, d_n$ of the second set denote which arguments are defended by E: $d_i = 1$ if and only if a_i is defended (from all the possible attacks) by some arguments of E. For each argument a_i, the Pfunction P_t^i forces t_i to be 1 if and only if a_i is attacked by E, i.e., $t_i = \bigvee_{jRi} x_j$.

Let h_i be the number of attackers of a_i and let $i_1, \ldots, i_{h_i}$ be their indices. If $h_i = 0$, then t_i is simply 0, while if $h_i = 1$, then $t_i = x_{h_1}$: in these cases, we set $P_t^i = 0$. If $h_i = 2$, then $P_t^i = OR(t_i, x[i_1], x[i_2])$, where $OR(Z, X, Y) = W + X + Y + XY - 2Z(X+Y)$ is the way of expressing as a quadratic function the constraint that the binary variable Z is the disjunction of the binary variables X and Y, as shown in [12]. Finally, if $h_i > 2$, then $P_t^i = OR(t_i, x[i_1], \alpha_i^1) + OR(\alpha_i^1, x[i_2], \alpha_i^2) + \ldots\ + OR(\alpha_i^{h_i-3}, x[i_{h_i-2}], \alpha_i^{h_i-2}) + OR(\alpha_i^{h_i-2}, x[i_{h_i-1}], x[i_{h_i}])$,, where $\alpha_i^1, \ldots, \alpha_i^{h_i-2}$ are $h_i - 2$ auxiliary binary variables.

The other Pfunction P_d^i forces d_i to be 1 if and only if a_i is defended by E, i.e., $d_i = \bigwedge_{jRi} t_j$. If $h_i = 0$, then d_i is simply 1, while if $h_i = 1$, then $d_i = t_{h_1}$: in these cases, $P_d^i = 0$. If $h_i = 2$,

[3]D-Wave Ocean SDK: https://github.com/dwavesystems/dwave-ocean-sdk.

then $P_d^i = AND(d_i, t[i_1], t[i_2])$, where $AND(Z, X, Y) = 3Z + XY - 2Z(X+Y)$ is the way of expressing the conjunction $Z = X$ and Y as a quadratic function [12]. Otherwise, if $h_i > 2$ then $P_d^i = AND(d_i, t[i_1], \delta_i^1) + AND(\delta_i^1, t[i_2], \delta_i^2) + \dots + AND(\delta_i^{h_i-3}, t[i_{h_i-2}], \delta_i^{h_i-2}) + AND(\delta_i^{h_i-2}, t[i_{h_i-1}], t[i_{h_i}])$, where $\delta_i^1, \dots, \delta_i^{h_i-2}$ are new $h_i - 2$ auxiliary binary variables.

The number of auxiliary variables needed for this encoding is $N = 2n + 2\sum_{i=1}^n \max(h_i - 2, 0)$, excluding the n variables $x_1, \dots, x_n$. Note that, if $h = \max h_i$, then $N = O(nh)$. The final term $P_{def} = \sum_{i=1}^n x_i(1 - d_i)$ forces each argument in E to be defended by E. Summing up, the Pfunction for **admissible** sets is $P_{adm} = P_{cf} + \sum_{i=1}^n P_t^i + \sum_{i=1}^n P_d^i + P_{def}$. It is easy to prove that the minimum value of P_{adm} is 0, and the related values for $\underline{x}$ correspond to admissible sets. For the **complete** semantics, we need to add term to P_{adm} which forces all the arguments defended by E to be elements of E: $P_{co} = P_{adm} + \sum_{i=1}^n (1 - x_i)d_i$.

3.2. Extension Enforcement in QUBO

An important subject that emerged in the literature in the last years concerns changes in AFs. In particular, attention has been paid to the problem of enforcing a set E of arguments, i.e., ensuring that E is an extension (or a subset of an extension) of a given framework $\mathcal{F}$.

The task of extension enforcement can be formulated with similar techniques. Let us focus on the strict version of this problem, concerning the complete semantics (see Tab. 2). To simplify the notation, the arguments in the set T are the first k arguments $a_1, \dots, a_k$ in A.

We use a first set of binary variables r_{ij}, for $i, j = 1, \dots, n$. Each variable r_{ij} is 1 whether in the new attack relationship R', a_i attacks a_j. Moreover, we use the binary variables t_i, for $i = 1, \dots, n$, and d_i, for $i = 1, \dots, k$, as in the encoding of the acceptance.

We define a Penalty function P_{co}^r, which is zero if and only if T is a complete extension under the attack relationship described by r_{ij}. P_{co}^r is the sum of 5 terms.

The first term $P_{cf}^r = \sum_{i,j=1}^k r_{ij}$. enforces the set T to be conflict-free, in fact when $r_{ij} = 1$, with $i, j \leq k$, we have a self attack in T.

The second term is $P_t^r = \sum_{i=1}^n P_t^{r,i}$, where $P_t^{r,i}$, for each $i = 1, \dots, n$, enforces the constraint $t_i = \bigvee_{j=1}^k r_{ji}$, which means that $t_i = 1$ if and only if the argument a_i is attacked by some argument $a_j \in T$. This term is encoded in QUBO using auxiliary binary variables, similar to what is done for P_t^i.

The third term is $P_d^r = \sum_{i=1}^k P_d^{r,i}$, where $P_d^{r,i}$, for each $i = 1, \dots, k$, enforces the constraint $d_i = \bigwedge_{j=1}^n (r_{ji} \implies t_j)$, which means that $d_i = 1$ if and only if the argument $a_i \in T$ is defended against all its attackers by some elements of T. This term is encoded in QUBO using a new set of auxiliary variables to represent the implication $(r_{ji} \implies t_j)$, other than the same auxiliary variables used for P_d^i.

The fourth term is simply $\sum_{i=1}^k (1 - d_i)$, which requires that all arguments in T are defended, while the last term is $\sum_{i=k+1}^n d_i$, which add a penality for each argument defended by T, but not belonging to T.

The overall objective function to be minimized is $f = \sum_{a_i R a_j} (1 - r_{ij}) + \sum_{\neg a_i R a_j} r_{ij} + \lambda P_{co}^r$, where λ is a constant large, such that the minimum of f is obtained for $P_{co}^r = 0$.

4. Related Work

In the literature, we find many computational techniques and practical implementations for solving problems related to formal argumentation in AI. We point the interested reader to the survey of participants and results achieved in ICCMA15 [13], ICCMA17 [14], and ICCMA19 [15]. ICCMA21 included a track for approximate approaches: only decision problems $\mathbf{DC}$-σ and $\mathbf{DS}$-σ were considered different sub-tracks, such as $\sigma \in \{\mathbf{co}, \mathbf{pr}, \mathbf{st}, \mathbf{sst}\}$. Solvers were evaluated for accuracy, i.e., the ratio of correctly solved instances. The main motivation behind approximate algorithms over exact algorithms was their (potentially) lower execution time.

An approximate solver from ICCMA21 is HARPER++ by M. Thimm: such a solver can only determine the grounded extension of an input framework and then uses that to approximate results for $\mathbf{DC}$ and $\mathbf{DS}$ tasks. A positive answer to $\mathbf{DS}$-$\mathbf{gr}$ implies a positive response to $\mathbf{DC}$ and $\mathbf{DS}$ for the other semantics. On the contrary, if an argument in the grounded extension attacks an argument, the answer to $\mathbf{DC}$ and $\mathbf{DS}$ is negative. According to [16], skeptical reasoning with any semantics generally overlaps with reasoning with the grounded semantics in many practical cases.

AFGCN, by Lars Malmqvist, competed in ICCMA21 as well. It uses a Graph Convolutional Network [17] to compute approximate solutions to $\mathbf{DC}$ and $\mathbf{DS}$ tasks for several semantics σ. The model is trained using a randomized training process using a dataset of AFs from previous ICCMA competitions to maximize generalization from the input frameworks. To speed up calculation and improve accuracy, the solver uses the pre-computed grounded extension as an input feature to the neural network.

5. Conclusion

As introduced in Section 1, our general goal is to study computational problems in Abstract Argumentation in the quantum world. QUBO represents the first approach to encoding and solving NP-complete problems on quantum annealers, as first accomplished in [6] on D-Wave annealers.

However, the optimization behind mapping QUBO models derived from an Argumentation problem to the architecture of quantum machines is still unexplored and challenging: several parameters need further investigation to exploit better the hardware and the connections among qubits, which are limited on D-Wave's architectures. We need to leave this to future work.

We also would like to try encodings different from QUBO and compare other quantum platforms and their Python API, such as *IBM Q* and *Google quantum*, where qubits are differently connected (less sparsely than D-Wave annealers).

References

[1] P. M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, Artificial Intelligence 77 (1995) 321–358.

[2] P. Baroni, M. Caminada, M. Giacomin, An introduction to argumentation semantics, The Knowledge Engineering Review 26 (2011) 365–410.
[3] P. Hammer, S. Rudeanu, Boolean methods in operations research and related areas, ökonometrie und unternehmensforschung/econometrics and operations research, Springer, Berlin 1007 (1968) 978–3.
[4] G. A. Kochenberger, J. Hao, F. W. Glover, M. W. Lewis, Z. Lü, H. Wang, Y. Wang, The unconstrained binary quadratic programming problem: a survey, J. Comb. Optim. 28 (2014) 58–81.
[5] J. P. Wallner, A. Niskanen, M. Järvisalo, Complexity results and algorithms for extension enforcement in abstract argumentation, J. Artif. Intell. Res. 60 (2017) 1–40.
[6] M. Baioletti, F. Santini, Abstract argumentation goes quantum: An encoding to QUBO problems, in: PRICAI 2022: 19th Pacific Rim International Conference on Artificial Intelligence, volume 13629 of *LNCS*, Springer, 2022, pp. 46–60.
[7] S. Coste-Marquis, S. Konieczny, J. Mailly, P. Marquis, Extension enforcement in abstract argumentation as an optimization problem, in: International Joint Conference on Artificial Intelligence, IJCAI, AAAI Press, 2015, pp. 2876–2882.
[8] F. W. Glover, G. A. Kochenberger, Y. Du, Quantum bridge analytics I: a tutorial on formulating and using QUBO models, 4OR 17 (2019) 335–371.
[9] F. W. Glover, G. A. Kochenberger, M. Ma, Y. Du, Quantum bridge analytics II: qubo-plus, network optimization and combinatorial chaining for asset exchange, 4OR 18 (2020) 387–417.
[10] S. Mücke, N. Piatkowski, K. Morik, Learning bit by bit: Extracting the essence of machine learning, in: R. Jäschke, M. Weidlich (Eds.), Proceedings of the Conference on "Lernen, Wissen, Daten, Analysen", volume 2454 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2019, pp. 144–155.
[11] S. Bistarelli, F. Rossi, F. Santini, Conarglib: an argumentation library with support to search strategies and parallel search, J. Exp. Theor. Artif. Intell. 33 (2021) 891–918.
[12] I. Rosenberg, Reduction of bivalent maximization to the quadratic case, Cahiers du Centre d'Etudes de Recherche Opérationnelle 17 (1975) 71–74.
[13] M. Thimm, S. Villata, The first international competition on computational models of argumentation: Results and analysis, Artif. Intell. 252 (2017) 267–294.
[14] S. A. Gaggl, T. Linsbichler, M. Maratea, S. Woltran, Design and results of the second international competition on computational models of argumentation, Artif. Intell. 279 (2020).
[15] S. Bistarelli, L. Kotthoff, F. Santini, C. Taticchi, A first overview of iccma'19, in: Advances In Argumentation In Artificial Intelligence (AIxIA 2020), volume 2777 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020, pp. 90–102.
[16] F. Cerutti, M. Thimm, M. Vallati, An experimental analysis on the similarity of argumentation semantics, Argument Comput. 11 (2020) 269–304.
[17] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, P. S. Yu, A comprehensive survey on graph neural networks, IEEE Trans. Neural Networks Learn. Syst. 32 (2021) 4–24.