A QUBO Approach to Value-Sensitive Reasoning in Argumentation Frameworks

Marco Baioletti[1,*], Fabio Rossi[1] and Francesco Santini[1]

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

Abstract

We propose a QUBO encoding for the *Subjective Acceptance Problem* (*SBA*) in *Value-Based Argumentation Frameworks* (*VAFs*), where argument acceptance depends on audience-specific value preferences. SBA, an NP-complete problem, decides whether an argument is accepted by at least one audience in the preferred extension. By translating SBA into the QUBO formalism (which is solvable with quantum annealing), we enable new computational approaches to value-sensitive reasoning. This is the first QUBO encoding of a preference-based problem in Argumentation, and we validate it through initial empirical testing.

Keywords

Quadratic unconstrained binary optimization, Value-based Argumentation, Simulated Annealing.

1. Introduction

In many real-world debating and reasoning scenarios, simply identifying arguments is not sufficient because multiple arguments often conflict and support each other in complex ways, particularly in moral and legal contexts [1]. To address this, there has been growing interest in logics for defeasible argumentation, which analyze how arguments can be defeated or defended.

For example, *Abstract Argumentation Frameworks* (*AF*s) [2] model conflicting arguments abstractly, focusing solely on their attack relations. Arguments are seen as unstructured tokens of information, and attacks symbolize a general directed conflict between two arguments: *a* may criticize *b* but not vice-versa. For example, with *a*: "We should ban single-use plastics because they cause long-term environmental damage", and *b*: "Single-use plastics are convenient for consumers", *a* attacks *b* because it highlights serious harm (environmental damage) that outweighs the convenience benefit mentioned by *b*. However, *b* does not attack *a*, as it only states a positive aspect without challenging or refuting the environmental concern raised by *a*. Conversely, in "structured" argumentation (e.g., [3]), which falls outside the direct scope of this paper, a formal language is used to represent knowledge and define how arguments and counterarguments are constructed from that knowledge. Arguments are considered structured because their premises and conclusions are explicitly stated, and the connection between them is formally specified, often through logical entailment.

In debates, arguments can conflict without necessarily defeating each other, as participants may prioritize/prefer different underlying *values*. For instance, one person may argue for higher taxes to promote equality, while another argues against it to reward enterprise; in this case, both people could accept the merit of the other's point but rank values differently. For this reason, to model debates effectively, it could be necessary to link arguments to the values they promote and to allow these values to be ordered according to the audience's preferences. Thus, the audience who attends the debate becomes a crucial factor in evaluating the arguments presented. *Value-based Argumentation Frameworks* (*VAFs*) [4] are one of the first attempts to extend the traditional AFs of Dung [2] by incorporating values into the structure of arguments. Values influence whether attacks between arguments succeed or fail, depending on their relative importance. In VAFs, an "audience" is formally represented through a preference ordering over these values; specifically, it is a total ordering (i.e., a complete, transitive,

AIQxQIA 2025: International Workshop on AI for Quantum and Quantum for AI | co-located with ECAI 2025, Bologna, Italy
*Corresponding author.
marco.baioletti@unipg.it (M. Baioletti); fabio.rossi@unipg.it (F. Rossi); francesco.santini@unipg.it (F. Santini)
0000-0001-5630-7173 (M. Baioletti); 0000-0002-8445-0142 (F. Rossi); 0000-0002-3935-4696 (F. Santini)

antisymmetric, and connected ranking). Intuitively, an audience models a perspective or worldview, that is, a particular way of ranking what is most important, such as different legal systems, ethical viewpoints, or stakeholder interests.

Quadratic Unconstrained Binary Optimization (*QUBO*) [5, 6, 7], also known as *Unconstrained Binary Quadratic Programming* (*UBQP*), is a combinatorial optimization problem with different applications in various fields, including finance, economics, and machine learning. The goal is to find the optimal combination of binary variables, each taking a value of 0 or 1, that minimizes (or maximizes) a quadratic expression. This expression can be represented by a matrix that assigns weights to each variable and each pair of variables, capturing both individual and pairwise interactions (i.e., constraints). The term "quadratic" refers to the fact that the objective function includes terms that involve the product of two variables. At the same time "unconstrained" means that there are no additional restrictions on the variables beyond being binary. QUBO is NP-hard to solve, and various classical problems in theoretical computer science, including maximum cut, graph coloring, and the partition problem, have been formulated as QUBO embeddings. Moreover, QUBO is used in machine learning, particularly in binary or combinatorial variants of problems. QUBO equivalents or approximations exist for tasks such as feature selection, clustering, and inference in graphical models.

The connection between QUBO problems and quantum annealing is fundamental. Quantum annealers[1] [8] are physical machines explicitly built to solve problems that can be expressed in QUBO (or its equivalent, the *Ising* model [5]). In quantum annealing, the machine encodes the optimization problem into a physical system, where the system's energy corresponds to the value of the QUBO objective function. The system is then allowed to settle into its lowest-energy (minimum) configuration, which corresponds to the optimal solution to the original QUBO problem. This is enabled by the mathematical equivalence between the QUBO model and the Ising model: in the latter model, the variables take values of -1 or $+1$ instead of 0 or 1, but a simple transformation allows the conversion between the two.

This paper aims to propose and test a QUBO encoding of the *Subjective Acceptance Problem* (*SBA*) defined in [9] to be NP-complete. Formally, given a VAF and an argument a, SBA is a decision problem that tests the existence of any audience such that a is included in the unique *preferred extension* determined by that audience.[2] Subjective acceptance captures the notion that an argument can be persuasive or justifiable, but only to specific groups of people or under certain value assumptions.

This paper continues the line of research initiated in [10, 11, 12, 13] by encoding and testing different NP-Complete problems in Argumentation to a QUBO formulation. In this paper, we introduce and encode problems related to preference-based AFs (i.e., VAFs) for the first time. As we have previously discussed in this section, preferences (or values) are necessary to make reasoning in debates more effective than using the original formulation [2], where all arguments have the same appeal to the audience. The paper is structured as follows: after this introduction (Sect. 1), outlining the field of research and its motivations, Sect. 2 presents the necessary preliminary notions related to VAFs and QUBO. Then, Sect. 3 proposes a QUBO encoding of the problem, and Sect. 4 describes the tests performed to validate such an encoding empirically. Finally, we end the paper with concluding thoughts and ideas about future work in Sect. 5.

2. Background

This section provides the essential background required to understand the remainder of this work. Section 2.1 introduces key concepts from Abstract Argumentation [2], including AFs, their extension to VAFs, and the SBA problem, which is NP-Complete. Section 2.2 offers an overview of what QUBO problems are and how to formalize them.

[1] For example, D-Wave quantum annealers: https://docs.dwavequantum.com/en/latest/quantum_research/quantum_annealing_intro.html.

[2] An extension is a set of arguments that is considered collectively acceptable or defensible according to specific semantic rules, see Sect. 2.

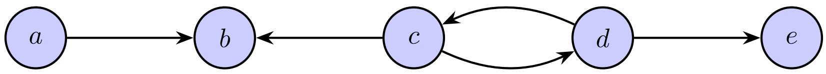


Figure 1: An example of an AF represented as a directed graph.

2.1. Problems in Abstract Argumentation Frameworks

An *Abstract Argumentation Framework* (AF, for short) [2] is a tuple $\mathcal{F} = (\mathsf{A}, \rightarrow)$ where A is a set of arguments and $\rightarrow$ is a relation $\rightarrow \subseteq \mathsf{A} \times \mathsf{A}$. For two arguments $a, b \in \mathsf{A}$, the relation $a \rightarrow b$ 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 $b \rightarrow a$, there is some $c \in S$ such that $c \rightarrow b$. A set $E \subseteq \mathsf{A}$ is *conflict-free* ($\mathbf{cf}$ in $\mathcal{F}$) if and only if there is no $a, b \in E$ with $a \rightarrow b$. E is *admissible* ($\mathbf{ad}$ in $\mathcal{F}$) if and only if it is conflict-free and if each $a \in E$ is defended by E. A directed graph can directly represent an AF: an example with five arguments is given in Fig. 1: $\mathcal{F} = (\{a, b, c, d, e\}, \{a \rightarrow b, c \rightarrow b, c \rightarrow d, d \rightarrow c, d \rightarrow e\})$.

The *collective acceptability* of arguments depends on the definition of different *semantics* [2]. Semantics determine sets of jointly acceptable arguments, that is, sets of arguments called *extensions*, by mapping each $\mathcal{F} = (\mathsf{A}, \rightarrow)$ 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. Four semantics were proposed by Dung in his seminal paper [2] (while some more have been defined in successive works [14]), namely the complete ($\mathbf{co}$), preferred ($\mathbf{pr}$), stable ($\mathbf{st}$), and grounded ($\mathbf{gr}$) semantics. Given $\mathcal{F} = (\mathsf{A}, \rightarrow)$ and a set $E \subseteq \mathsf{A}$, we report the definition of all these semantics:

- $E \in \mathbf{co}(\mathcal{F})$ 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. $E' \supset E$,
- $E \in \mathbf{st}(\mathcal{F})$ 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. $E' \subset E$.

For a more detailed view of these semantics, please refer to [14]. Note that the grounded semantics always uniquely identify a single extension [2] (i.e., it is a *single-status* semantics). In contrast, the other introduced semantics are *multi-status*, since several extensions may exist. The stable semantics is the only case where $\mathbf{st}(\mathcal{F})$ might be empty (i.e., no stable extension exists), while at least one extension always satisfies the other semantics. As an example, if we consider the framework $\mathcal{F}$ in Fig. 1, we obtain the following extensions:

- $\mathbf{cf}(\mathcal{F}) = \{\emptyset, \{a\}, \{b\}, \{c\}, \{d\}, \{e\}, \{a, c\}, \{a, d\}, \{a, e\}, \{b, d\}, \{b, e\}, \{c, e\}, \{a, c, e\}\}$;
- $\mathbf{ad}(\mathcal{F}) = \{\emptyset, \{a\}, \{c\}, \{d\}, \{a, c\}, \{a, d\}, \{c, e\}, \{a, c, e\}\}$;
- $\mathbf{co}(\mathcal{F}) = \{\{a\}, \{a, d\}, \{a, c, e\}\}$;
- $\mathbf{pr}(\mathcal{F}) = \{\{a, d\}, \{a, c, e\}\}$;
- $\mathbf{gr}(\mathcal{F}) = \{\{a\}\}$.

Value-based Argumentation Frameworks [4] enrich the expressiveness of AFs by incorporating subjective values and audience-dependent evaluation, making them particularly suited for domains where reasoning involves value conflicts rather than purely logical ones. They connect formal argumentation with real-world debates, where the outcome is not determined solely by logic but also by which values are most important to the people involved.

Definition 2.1 (Value-based Argumentation Frameworks [4])**.** A *Value-based Argumentation Framework* (VAF) is a triple $\mathcal{F}_v = ((\mathsf{A}, \rightarrow), V, \eta)$, where $(\mathsf{A}, \rightarrow)$ is an AF, $V = \{v_1, \ldots, v_k\}$ is a set of k values, and $\eta : \mathsf{A} \rightarrow V$ is a function that assigns a value to each argument. An *audience* $\leq$ for a VAF is a total ordering over the set of values V. For values $v_i, v_j \in V$, we say that v_i is preferred to v_j in the audience $\leq$, denoted $v_i \leq v_j$, if v_i is ranked higher than v_j in the total ordering.

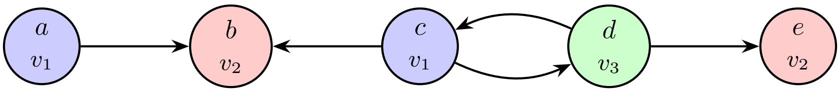


Figure 2: An example of VAF, obtained from Fig. 1.

In Fig. 2 we show an example of VAF $\mathcal{F}_v = ((\mathsf{A}, \rightarrow), V, \eta)$ obtained from the AF in Fig. 1, where $\mathsf{A} = \{a, b, c, d, e\}$, $\rightarrow = \{(a, b), (c, b), (d, e), (c, d), (d, c)\}$, $V = \{v_1, v_2, v_3\}$, and $\eta(a) = v_1$, $\eta(b) = v_2$, $\eta(c) = v_1$, $\eta(d) = v_3$, $\eta(e) = v_2$.

The introduction of audiences affects the classical notion of attacks given for AFs in [2].

Definition 2.2 (Attacks in VAFs [4])**.** Given a VAF, an argument $a \in \mathsf{A}$ *successfully attacks* an argument $b \in \mathsf{A}$ with respect to an audience $\geq$, if $(a, b) \in \rightarrow$ and $\eta(b) \not> \eta(a)$.

Therefore, an attack only "works" if the attacker's value is not less preferred than the one being attacked. Otherwise, the argument is no longer considered part of the underlying $\mathcal{F}$, and the definition of the semantics presented before changes accordingly. For example, if we consider the VAF in Fig. 2 and the audience $v_1 \geq v_2 \geq v_3$, the successful attacks are only $\{(a, b), (c, b), (c, d)\}$: $\{d, e\}$ is now considered conflict-free and $\{d\}$ is no longer admissible or complete, compared to Fig. 1, for example.

The SBA problem [9] is a decision problem that answers the question of whether there is at least one perspective or value system under which a given argument can be accepted as justified. This has implications in real-world reasoning and decision-making, particularly in areas where people or institutions have different values or priorities, as, ethical debates (e.g., euthanasia, privacy vs. security), political discourse (e.g., freedom, equality, security), or in general systems where different agents have individual preferences or objectives: SBA helps determine whether a proposal is acceptable to at least one agentev en if it is not universally persuasive.

Definition 2.3 (Subjective Acceptance (SBA) [9])**.** Given a VAF $\mathcal{F}_v = ((\mathsf{A}, \rightarrow), V, \eta)$ and an argument $a \in \mathsf{A}$ in this framework, the SBA problem accepts the instance $(\mathcal{F}_v, a)$ if there exists at least one audience $\leq$ such that $a \in P_{\leq}$, where $P_{\leq}$ is the preferred extension of $\mathcal{F}$ (i.e., $\{P\} = \mathbf{pr}(\mathcal{F})$.) relative to the audience $\leq$.

In SBA, the preferred semantics is used because it identifies arguments that are defensible under specific value orderings, allowing for multiple acceptable viewpoints that reflect the diversity of audience preferences. Unlike stable semantics, it is not overly strict, and, unlike the grounded semantics, it is not excessively skeptical, making it a balanced and suitable approach for capturing the notion of subjective acceptability. Note that Def. 2.3 relies on the existence of a single preferred extension in $(\mathsf{A}, \rightarrow)$ given an audience $\leq$ because of Prop. 2.1.

Proposition 2.1 (Unicity of preferred/stable [15])**.** Let $\mathcal{F}_v = ((\mathsf{A}, \rightarrow), V, \eta)$ be a VAF, and let $\leq$ be an audience. We assume that every directed cycle in the argument graph $(\mathsf{A}, \rightarrow)$ involves at least two distinct values: i.e., there are no cycles consisting only of arguments associated with the same value by η. There exists a unique non-empty preferred extension $P \subseteq \mathsf{A}$ (i.e., $\{P\} = \mathbf{pr}(\mathcal{F})$) considering $\leq$, that is, there exists only one $P_{\leq}$. Moreover, P is also the only stable extension given $\leq$: $\{P\} = \mathbf{st}(\mathcal{F})$, whose existence is thus guaranteed in this case.

The SBA problem is shown to be NP-complete in [9], where the proof is obtained by means of a reduction from a 3-SAT problem: the reduction constructs a VAF such that an argument is subjectively acceptable if and only if the 3-CNF formula is satisfiable.[3]

The no-cycle condition on the values in Prop. 2.1 is not stringent because single-value cycles typically represent paradoxes or irrational structures [15]. Such cycles rarely occur in practical reasoning, where conflicts commonly arise between differing values, rather than within the same value. Figure 3 shows a VAF that respects the no-cycle condition on the values, while Fig. 4 shows a VAF that does not respect this condition.

[3]A 3CNF formula is a Boolean formula in conjunctive normal form where each clause contains exactly three literals.

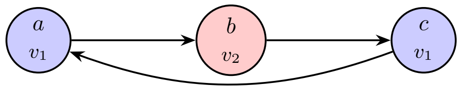


Figure 3: Valued no-cycle condition respected.

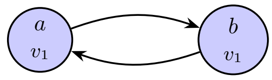


Figure 4: Valued no-cycle condition not respected.

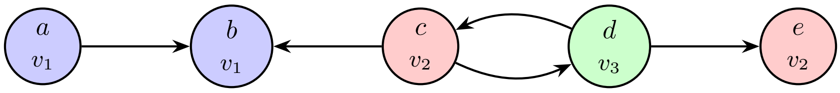


Figure 5: A second ecample of VAF.

As an example, given the VAF $\mathcal{F}_v$ in Fig. 2 and the argument c, the SBA accepts the instance $(\mathcal{F}_v, c)$ because there exists an audience, for example $v_1 \geq v_2 \geq v_3$, for which the preferred/stable extension is $P_{\geq} = \{a, c, e\}$, that is, $c \in P_{\geq}$ as requested by Def. 2.3. Note that with the VAF in Fig. 2, all the arguments are subjectively acceptable: a is always acceptable with any audience, b is acceptable if $\eta(b) \geq \eta(a) \wedge \eta(b) \geq \eta(c)$ (e.g., $v_2 \geq v_1$), c is acceptable if $\eta(c) \geq \eta(d)$ (e.g., $v_1 \geq v_3$), d is acceptable if $\eta(d) \geq \eta(c)$ (e.g., $v_3 \geq v_1$), and finally e is acceptable if $\eta(e) \geq \eta(d)$ (e.g., $v_2 \geq v_3$). Given the VAF $\mathcal{F}_v$ in Fig. 2 and the argument b instead, the SBA does not accept the instance $(\mathcal{F}_v, b)$ as for any audience $\geq$ the attack (a, b) is always successful since $\eta(a) = \eta(b) = v_1$.

2.2. Quantum Annealing and QUBO

Quantum Annealing is an approach [16] that uses a quantum computer to tackle optimization problems by identifying the lowest energy state configuration. This technique relies on the *Quantum Adiabatic Theorem* and leverages quantum tunneling to deliver optimal or nearly optimal solutions for discrete optimization problems [16]. The process begins with an initial function ϕ_s, for which the solution that minimizes the energy is easily determined. It gradually transitions to the final function ϕ_f, which corresponds to the function to be optimized. If the transition is slow enough, the *Quantum Adiabatic Theorem* guarantees that the solution with the minimum energy adapts to the change of the objective function.

Quantum Annealing is implemented by contemporary architectures such as D-Wave, as highlighted in Sect. 1. This company develops a line of quantum annealing devices that execute quantum annealing algorithms. In particular, their most sophisticated machines can handle problems involving thousands of variables.

Quantum annealers can be "programmed" by encoding the problems as *Quadratic Unconstrained Binary Optimization* [6, 7] (in short, *QUBO*) or as Ising models [17]. In this paper, we employ the QUBO formalism. It is expressive enough to encode several optimization problems formulated in various application domains [18]. QUBO has been extensively studied and is used to define and address a wide range of 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 [7, 18]. Moreover, QUBO embeddings exist for Support Vector Machines, Clustering algorithms, and Markov Random Fields [19].

In a QUBO problem, n binary variables $x_1, \ldots, x_n$ and an $n \times n$ upper-triangular matrix Q are used to formulate the task, which involves minimizing (or sometimes maximizing) the second-order 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 square matrix of coefficients can be organized in a symmetric way, where for all i, j except $i = j$, $Q_{i,j}$ is replaced by $(Q_{i,j} + Q_{j,i})/2$, or, as stated before, in an upper-diagonal form where for all i, j s.t. $i > j$, $Q_{i,j}$ is replaced by $Q_{i,j} + Q_{j,i}$ and then all $Q_{i,j}$ are replaced by 0 for $j < i$.

To formulate a discrete constrained optimization problem into a QUBO, one must: *i)* determine a binary representation of the potential solutions, and *ii)* define a penalization function to deter non-feasible solutions, specifically those that violate a constraint.

Except for the 0/1 limitations on the decision variables, QUBO is an unconstrained model, where the matrix Q contains all problem data. Rather than applying constraints in the conventional sense, classical constrained models can be successfully reformulated as QUBO models by inserting quadratic penalties into the objective function. In constrained optimization problems, selecting a penalty that is too small may lead to infeasible solutions. In contrast, employing a significant penalty to ensure constraints are met can make it challenging for the optimization algorithm to find feasible solutions. This strategy may also introduce challenges, such as sensitivity to penalty values, increased computational demands during optimization, and instability in the iterative process of the solver. For this reason, a substantial amount of literature has been produced to find good coefficients [20], or techniques to model classical constraints into a function, e.g., a logical *and* constraint between x_1 and x_2 as a multiplication $x_1 \cdot x_2$.

The literature on precise approaches to QUBO on conventional computers includes various algorithms [18], all of which converge to a globally optimal solution given sufficient time and memory. Most of these techniques use a standard branch-and-bound tree search, although alternative methods are also available. For example, in [21], a QUBO solution is based on the inherent geometric properties of the minimum circumscribed sphere that contains the ellipsoidal contour of the objective function, while in [22], the authors adopt Lagrangean decompositions. The NP-hardness of QUBO and its various potential applications have led to a significant number of research papers being published in recent years. These documents outline a variety of heuristic methods designed to rapidly identify high-quality solutions to medium- to large-scale problem cases. Although some of these techniques may be simple enough to be regarded as heuristics, the most effective ones are metaheuristic processes. These involve more complex compound strategies that outperform basic heuristics. For example, in addition to simulated annealing [23], the work in [24] presents a guided tabu-search algorithm alternating between a basic tabu-search procedure and a variable fix/free phase. In contrast, [25] presents a hybrid metaheuristic approach that combines crossover and update operators with tabu search to evaluate the offspring solutions.

3. Encoding VAFs and Audiences in QUBO

The encoding of Argumentation problems in QUBO uses a set of n binary variables $x_1, \ldots, x_n$ associated with arguments $\{a_1, \ldots, a_n\}$ in A. The variables $x_1, \ldots, x_n$ represent a subset E of A: $a_i \in \mathsf{A}$ if and only if $x_i = 1$ [10]. Each semantics σ will be associated with a quadratic penalty function P_σ such that P_σ assumes its minimum value in the values $(x_1, \ldots, x_n)$ if and only if the corresponding set $E = \{a_i \in \mathsf{A} : x_i = 1\}$ is an extension valid for σ.

For the SBA problem, we also use $k^2 - k$ binary variables u_{ij}, with $i, j = 1, \ldots, k$, with $i \neq j$, with the meaning that $u_{ij} = 1$ if and only if v_i is preferred to v_j in the audience $\leq$. To enforce that the variables u_{ij} encode a total order, three quadratic terms are employed.

The first term encodes that for each $i, j = 1, \ldots, k$, with $i \neq j$, v_i is preferred to v_j or viceversa. This term is:

$$T_{anti-symm} = \sum_{i \neq j} (u_{ij} + u_{ji} - 1)^2. \tag{1}$$

The second term is used to encode the transitivity constraint:

$$T_{trans} = \sum_{i \neq j, i \neq l, j \neq l} \omega_{ijl}(1 - u_{il}), \tag{2}$$

where the binary variables ω_{ijl}, for $i, j, l = 1, \ldots, k$ are used to represent the product (or conjunction) $u_{ij} \cdot u_{jl}$. Here and in other equations, we use sets of additional binary variables (and corresponding penalty terms to avoid cubic terms, which are products of three variables).

The third term has the purpose of encoding the constraint on ω_{ijl}:

$$T_\omega = \sum_{i \neq j, i \neq l, j \neq l} AND(\omega_{ijl}, u_{ij}, u_{jl}). \tag{3}$$

The AND function [26] used in Eq. 3 and later in other equations,

$$AND(z, x, y) = 3z + xy - 2z(x + y), \tag{4}$$

expresses the conjunction $z = (x \text{ and } y)$ of binary variables x, y as a quadratic function.

We denote by $T = T_{anti-symm} + T_{trans} + T_\omega$ the sum of these three quadratic terms. To encode the constraint of a stable extension, we first need to express that the set is complete. Hence, we define a penalty function P_{co} that enforces this property, which is the sum of 4 terms. The first term forces the set E to be **conflict-free**:

$$P_{cf} = \sum_{a_i \to a_j} y_{ij} u_{\eta(a_i), \eta(a_j)}. \tag{5}$$

The value of P_{cf} corresponds to the number of internal attacks in E that respect the audience $\leq$, and its value is 0 if and only if E is conflict-free.

The n^2 binary variables y_{ij}, for $i, j = 1, \ldots, n$ are used to represent the product (or conjunction) $x_i x_j = 1$. Hence, an additional penalty function must be included:

$$P_y = \sum_{i,j=1}^{n} AND(y_{ij}, x_i, x_j).$$

The constraints for modeling the notion of **defense** are more complex and require additional variables. The first set contains the variables $t_1, \ldots, t_n$, which denote the arguments that are successfully attacked by E: $t_i = 1$ if and only if some argument of E attacks a_i with respect to $\leq$.

For each argument a_i, the penalty function P_t^i enforces the logical constraint

$$t_i = \bigvee_{a_j \to a_i} x_j u_{\eta(a_j), \eta(a_i)}.$$

To express this constraint in QUBO, three "ingredients" are needed. The function

$$OR(z, x, y) = z + x + y + xy - 2z(x + y) \tag{6}$$

is used to express the constraint that the binary variable z is the disjunction $z = (x \text{ or } y)$ of the binary variables x, y by means of a quadratic function, as shown in [26].

Each product of the form $x_j u_{\eta(a_j), \eta(a_i)}$ must be replaced by a single binary variable ϕ_{ji}, for $j = 1, \ldots, n$ such that $a_j \to a_i$. The constraints $\phi_{ji} = x_j u_{\eta(a_j), \eta(a_i)}$ for $j = \ldots, n$ are enforced by the penalization term

$$P_\phi^i = \sum_{a_j \to a_i} AND(\phi_{ji}, x_j, u_{\eta(a_j), \eta(a_i)}).$$

Each penalty function P_t^i requires $\max\{h_i - 2, 0\}$ auxiliary variables α_i^j, where h_i is the number of possible attackers of a_i, to represent the $\bigvee$ operator as a composition of OR functions. If $h_i \leq 2$, no additional variable is required, as explained in [10]. On the other hand, if $h_i > 2$ and the possible attackers of a_i are $a_{i_1}, \ldots, a_{i_h}$, then

Set	Size
$\{x_i\}$	n
$\{t_i\}$	n
$\{u_{ij}\}$	$k^2 - k$
$\{\omega_{ijl}\}$	k^3
$\{y_{ij}\}$	$\sum_{i=1}^{n} h_i$
$\{\phi_{ji}\}$	$\sum_{i=1}^{n} h_i$
$\{\theta_{ji}\}$	$\sum_{i=1}^{n} h_i$
$\{\alpha_{ji}\}$	$\sum_{i=1}^{n} \max(h_i - 2, 0)$
$\{\delta_{ji}\}$	$\sum_{i=1}^{n} \max(h_i - 2, 0)$

$$P_t^i = OR(t_i, \phi[i_1, i], \alpha_i^1) + OR(\alpha_i^1, \phi[i_2, i], \alpha_i^2) + \ldots + OR(\alpha_i^{h_i-3}, \phi[i_{h_i-2}, i], \alpha_i^{h_i-2}) + OR(\alpha_i^{h_i-2}, \phi[i_{h_i-1}, i], \phi[i_{h_i}, i]), \quad (7)$$

The binary variables $d_1, \ldots, d_n$ denote the arguments that are defended by E: $d_i = 1$ if and only if a_i is defended (from all successful attacks) by some arguments of E.

The penalty function P_d^i forces x_i to be 1 if and only if a_i is defended by E respecting $\leq$, i.e., $d_i = \bigwedge_{a_j \to a_i} (t_j \vee \neg u_{\eta(a_j),\eta(a_i)})$.

In fact, a_i is defended if for each possible attacker a_j, either a_j is attacked by E or the attack is not successful, i.e., $u_{\eta(a_j),\eta(a_i)} = 0$. In addition, in this case, each term $(t_j \vee \neg u_{\eta(a_j),\eta(a_i)})$ must be replaced by the binary variable θ_{ji}, for $j = 1, \ldots, n$ such that $a_j \to a_i$.

The constraints $\theta_{ji} = (t_j \vee \neg u_{\eta(a_j),\eta(a_i)})$, for $j = 1, \ldots, n$, are enforced by the penalization term

$$P_\theta^i = \sum_{a_j \to a_i} OR(\theta_{ji}, t_j, 1 - u_{\eta(a_j),\eta(a_i)})$$

Each penalty function P_d^i requires $\max\{h_i - 2, 0\}$ auxiliary variables δ_i^j, for $j = 1, \ldots, h_i - 2$ to express the $\bigwedge$ operator as a composition of the AND function.

When $h_i > 2$ the penalty function is

$$P_d^i = AND(x_i, \theta[i_1, i], \delta_i^1) + AND(\delta_i^1, \theta[i_2, i], \delta_i^2) + \ldots + AND(\delta_i^{h_i-3}, \theta[i_{h_i-2}, i], \delta_i^{h_i-2}) + AND(\delta_i^{h_i-2}, \theta[i_{h_i-1}, i], \theta[i_{h_i}, i]) \quad (8)$$

To encode that E is a complete extension, the constraint $x_i = d_i$ must hold for $i = 1, \ldots, n$.

Hence, the penalty function for **complete** sets is

$$P_{co} = T + P_{cf} + P_y + \sum_{i=1}^{n} (P_t^i + P_d^i) + \sum_{i=1}^{n} \sum_{a_j \to a_i} (P_\phi^{ji} + P_\theta^{ji}) + \sum_{i=1}^{n} (x_i(1 - d_i) + d_i(1 - x_i))$$

The sets of binary variables appearing in P_{co} are listed in Tab. 3. The number of these variables is $N = 2n + k^2 - k + k^3 + 3\sum_{i=1}^{n} h_i + 2\sum_{i=1}^{n} \max(h_i - 2, 0)$, which can be simplified as $N = O(nh + k^3)$, where $h = \max h_i$.

Finally, the encoding of the **stable** semantics is obtained by adding an additional term to P_{co} which forces all arguments not belonging to E to be attacked by E:

$$P_{st} = P_{co} + \sum_{i=1}^{n} (1 - x_i)(1 - t_i) \quad (9)$$

It is straightforward to prove that the minimum value of P_{st} is 0 if and only if the corresponding E is a stable extension. Hence, this encoding can be used to determine whether a stable extension exists.

Solving the SBA problem for a given argument a_i is sufficient to minimize P_{st} after setting $x_i = 1$. If the minimum is 0, then a_i is accepted; the values of variables x_i indicate the stable/preferred extension, and the values of u_{ij} reveal the order relation $\leq$.

Table 1
Experimental results on $n = 20, 25, 30, 35$ nodes/arguments graphs generated with *Erdős-Rényi* [28].

n	na	#AFs	$\min e_{min}$	$\max e_{min}$
20	0	19	0	0
	1	16	1	1
	2	14	1	1
	3	1	1	1
25	0	19	0	0
	1	9	1	1
	2	6	1	1
	3	6	1	1
	4	3	1	1
	6	1	1	2
	7	3	1	2
	8	1	1	1
	9	1	1	2
	10	1	1	2
30	1	2	1	1
	3	1	1	1
	10	1	1	2
	12	2	1	3
	16	1	1	3
	17	1	1	2
	19	2	1	3
	20	1	1	4
	23	2	1	4
	24	1	1	4
	30	36	5	212
35	35	50	9	275

4. Experiments

As a proof of concept, we have conducted a preliminary series of experiments using the Simulated Annealing (SA) [27] algorithm present in the Ocean SDK[4] package. As noted in Sect. 2.2, the Hamiltonian presented in Sect. 3 can be minimized using methods other than SA; however, in the future, we would like to test the implementation on actual quantum annealers (Sect. 5).

To achieve this aim, we have generated four sets of 50 random directed graphs, each with $n = 20, 25, 30, 35$ nodes/arguments, for a total of 200 graphs. For every graph, representing a different VAF, we used the *Erdős-Rényi* random generator [28], and we used an edge/attack probability of $p = 0.09$. In each VAF, we have randomly assigned a value $\eta(a)$ to all arguments among the $\lfloor n/2 \rfloor$ possible values $v_1, \ldots, v_{\lfloor n/2 \rfloor}$. In this way, the average number of arguments that share the same value is two. Then, we have tested whether, in all the cycles of the graph, the arguments are assigned to at least two different values, as required by Prop. 2.1. In the negative case, we generate another argument-value assignment η until the VAF satisfies the condition.

For each VAF, we have tested the acceptance of each argument by running the SA on 1000 reads of the corresponding QUBO model. We denote by e_{min} the minimum value of the energy (i.e., the objective function) obtained in all reads for a given VAF and a given argument a.

The aggregate results are shown in Tab. 1, where n is the number of arguments, na is the possible number of not accepted arguments, #AFs denotes the number of AFs of n arguments in which the number of not accepted arguments is na, $\min e_{min}$ and $\max e_{min}$ are respectively the minimum value of e_{min} and the maximum value of e_{min} computed for all not accepted arguments.

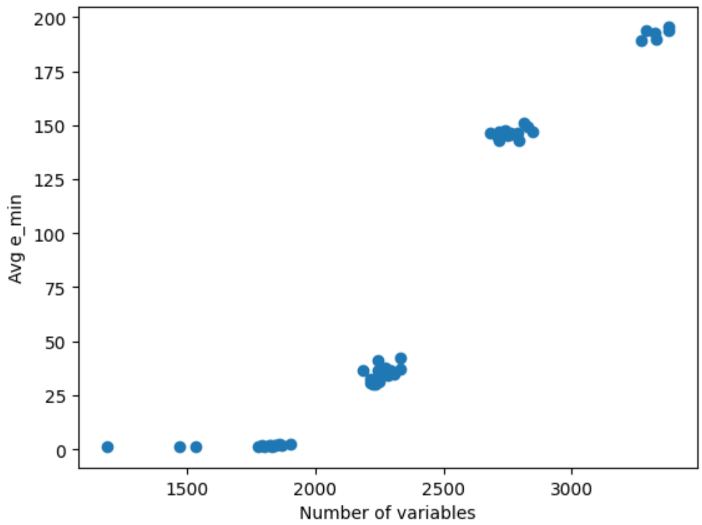
The results can be summarized as follows. For $n = 20$, in 19 VAFs all the arguments are accepted, in

[4] Ocean SDK: https://docs.dwavequantum.com/en/latest/ocean/index.html.

Table 2
Size of QUBO models for each set of graphs, in the number N of variables.

n	min N	avg N	max N
20	416	718.16	1016
25	638	1291.92	1761
30	1187	2381.96	3379
35	2343	3577.60	4908

Figure 6: Average e_{min} compared to the number of variables N.



16 VAFs only one argument is not accepted, in 14 VAFs two arguments are not accepted, and in one VAF three arguments are not accepted. It is interesting to see that the $e_{min} = 1$ in the AFs when the argument is not accepted.

For $n = 25$, again in 19 VAFs all arguments are accepted. In the remaining 24 VAFs, the number of arguments not accepted ranges from 1 to 4. Also, in these cases, $e_{min} = 1$ when the argument is not accepted. The same value for the minimum energy was found in an AF with $na = 8$ and in an AF with $na = 7$ (the latter data does not appear in Tab. 1). The situation is different for the remaining 5 VAFs with $n = 25$. In fact, we have found that in these VAFs, e_{min} is 2 for at least one argument. Currently, we do not know whether 2 is the correct value of the objective function when the argument is not accepted. Hence, we cannot determine whether the value 2 indicates that either the argument is unacceptable or the SA was unable to find a lower-energy solution.

For $n = 30$, apparently, in no AF are all arguments accepted. We have found 2 VAFs with one argument not accepted and 3 VAFs with 3 arguments not accepted. In these cases, the minimum energy is 1. In all the other AFs, the number of arguments not accepted is unclear. In particular, in 36 VAFs, no argument was accepted, which is highly unlikely. In these combinations, the minimum energy ranges from 5 to 212. Moreover, in 20 VAFs, we have $e_{min} > 90$ for all arguments, which is a very large value and a strong indication that the SA has not been able to optimize the objective function. Finally, all experiments with $n = 35$ have a unique outcome: in each AF, none of the arguments are accepted. This outcome is likely due to the non-optimal solutions found by the SA.

A partial justification can be given by looking at Tab. 2, in which the number of variables N appearing in the corresponding QUBO models is reported for each value of n. It is evident that the average value of N nearly doubles when n increases from 25 to 30. The SA is likely trapped in a local minimum when N exceeds 2000.

In fact, a significant correlation between N and the average value of e_{min} can be observed in Fig. 6.

5. Conclusion

This paper elaborates on the research initiated in [10, 11, 12, 13] and introduces the first QUBO encoding for a preference-based problem in Argumentation, explicitly addressing the Subjective Acceptance problem in Value-based Argumentation Frameworks. By translating the NP-complete SBA problem into the QUBO formalism, we enable novel computational approaches to energy-minimization reasoning, particularly allowing the use of quantum annealers to solve the problem.

The encoding involves defining binary variables for arguments and value preferences and constructing quadratic penalty functions to enforce various constraints, such as conflict-freeness and defense.

Future research will focus on several directions. First, we plan to explore the scalability of this QUBO encoding for larger and more complex VAFs, investigating its performance on current and future quantum annealing hardware. The experimental results presented in this article are preliminary, and further investigation is necessary in the following steps of this work. For example, we will check the correctness of the solutions we find against the outputs provided by *ASPARTIX* [29],[5] which consists of a set of AF/VAF encodings to be solved by *Answer-set Programming* (*ASP*) solvers.

Acknowledgments

M. Baioletti has been partially supported by: European Union - NextGenerationEU, Mission 4, Component 2, under the Italian Ministry of University and Research (MUR) National Innovation Ecosystem grant ECS00000041 - VITALITY - CUP J97G22000170005.

F. Santini has been partially supported by: European Union - Next Generation EU PNRR MUR PRIN - Project J53D23007220006 EPICA: "Empowering Public Interest Communication with Argumentation", MUR PNRR project SERICS (PE00000014 AQuSDIT: CUP_H73C22000880001), funded by the European Union – Next Generation EU; University of Perugia - Fondo Ricerca di Ateneo (2022) – Project RATIONALISTS, WP4.1 on "AI Data Management and Data Science".

Declaration on Generative AI

During the preparation of this work, the authors used Grammarly to perform grammar and spelling checks and to reword. After using this tool, the authors reviewed and edited the content as needed and take full responsibility for the publication's content.

References

[1] D. Walton, Fundamentals of critical argumentation, Cambridge University Press, 2005.

[2] 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.

[3] P. Besnard, A. J. García, A. Hunter, S. Modgil, H. Prakken, G. R. Simari, F. Toni, Introduction to structured argumentation, Argument Comput. 5 (2014) 1–4. URL: https://doi.org/10.1080/19462166.2013.869764.

[4] T. J. M. Bench-Capon, Value-based argumentation frameworks, in: S. Benferhat, E. Giunchiglia (Eds.), 9th International Workshop on Non-Monotonic Reasoning (NMR 2002), April 19-21, Toulouse, France, Proceedings, 2002, pp. 443–454.

[5] F. Glover, G. Kochenberger, Y. Du, A tutorial on formulating and using QUBO models, 2019. URL: https://arxiv.org/abs/1811.11538. arXiv:1811.11538.

[6] 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.

[5] ASPARTIX for Value-based Argumentation Frameworks (VAFs): https://www.dbai.tuwien.ac.at/research/argumentation/aspartix/vaf.html.

[7] 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.
[8] A. Rajak, S. Suzuki, A. Dutta, B. K. Chakrabarti, Quantum annealing: An overview, Philosophical Transactions of the Royal Society A 381 (2023) 20210417.
[9] P. E. Dunne, T. J. M. Bench-Capon, Complexity in value-based argument systems, in: Logics in Artificial Intelligence, 9th European Conference, JELIA, volume 3229 of *LNCS*, Springer, 2004, pp. 360–371.
[10] M. Baioletti, F. Santini, Abstract argumentation goes quantum: An encoding to QUBO problems, in: PRICAI 2022: Trends in Artificial Intelligence - 19th Pacific Rim International Conference on Artificial Intelligence, volume 13629 of *LNCS*, Springer, 2022, pp. 46–60.
[11] M. Baioletti, F. Santini, On using QUBO to enforce extensions in abstract argumentation (short paper), in: Proceedings of the International Workshop on AI for Quantum and Quantum for AI (AIQxQIA 2023), volume 3586 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2023.
[12] M. Baioletti, F. Rossi, F. Santini, Enumerating extensions in abstract argumentation by using QUBO, in: Proceedings of the International Workshop on AI for Quantum and Quantum for AI (AIQxQIA 2024), volume 3913 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2024.
[13] M. Baioletti, F. Santini, An encoding of argumentation problems using quadratic unconstrained binary optimization, Quantum Mach. Intell. 6 (2024) 51.
[14] P. Baroni, M. Caminada, M. Giacomin, An introduction to argumentation semantics, The Knowledge Engineering Review 26 (2011) 365–410.
[15] T. J. Bench-Capon, Agreeing to differ: modelling persuasive dialogue between parties with different values, Informal Logic 22 (2002) 231–246.
[16] S. E. Venegas-Andraca, W. Cruz-Santos, C. McGeoch, M. Lanzagorta, A cross-disciplinary introduction to quantum annealing-based algorithms, Contemporary Physics 59 (2018) 174–197.
[17] A. Lucas, Ising formulations of many np problems, Frontiers in Physics 2 (2014).
[18] 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.
[19] 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, Berlin, Germany, 2019, pp. 144–155.
[20] A. Verma, M. W. Lewis, Penalty and partitioning techniques to improve performance of QUBO solvers, Discret. Optim. 44 (2022) 100594.
[21] D. Li, X. Sun, C. Liu, An exact solution method for unconstrained quadratic 0-1 programming: a geometric approach, J. Glob. Optim. 52 (2012) 797–829.
[22] G. R. Mauri, L. A. N. Lorena, Improving a lagrangian decomposition for the unconstrained binary quadratic programming problem, Comput. Oper. Res. 39 (2012) 1577–1581.
[23] T. M. Alkhamis, M. Hasan, M. A. Ahmed, Simulated annealing for the unconstrained quadratic pseudo-boolean function, Eur. J. Oper. Res. 108 (1998) 641–652.
[24] J. Wang, Y. Zhou, J. Yin, Combining tabu hopfield network and estimation of distribution for unconstrained binary quadratic programming problem, Expert Syst. Appl. 38 (2011) 14870–14881.
[25] Z. Lü, F. W. Glover, J. Hao, A hybrid metaheuristic approach to solving the UBQP problem, Eur. J. Oper. Res. 207 (2010) 1254–1262.
[26] I. Rosenberg, Reduction of bivalent maximization to the quadratic case, Cahiers du Centre d'Etudes de Recherche Opérationnelle 17 (1975) 71–74.
[27] S. Kirkpatrick, C. D. Gelatt Jr, M. P. Vecchi, Optimization by simulated annealing, science 220 (1983) 671–680.
[28] P. Erdős, A. Rényi, On the evolution of random graphs, Bull. Inst. Internat. Statist 38 (1961) 343–347.
[29] U. Egly, S. A. Gaggl, S. Woltran, Answer-set programming encodings for argumentation frameworks, Argument Comput. 1 (2010) 147–177.