Abstract argumentation goes quantum: models and experiments

Francesco Santini

Department of Mathematics and Computer Science
University of Perugia, Italy

Introduction

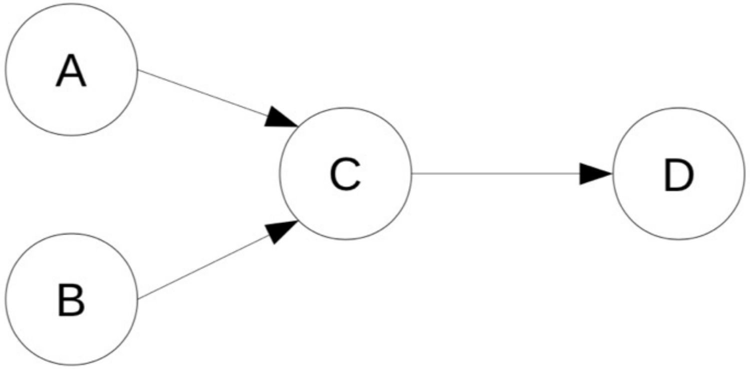
- *Abstract Argumention* [P.M. Dung ‘95]
- *Quadratic Unconstrained Binary Optimization* problems (*QUBO*) [Hammer, Rudeanu ‘68]

- A QUBO encoding for all (classical) NP-Complete problems in argumentation
- We solve the encoding by using *pycubo*
  - Simulated Annealers (locally)
  - Quantum annealers (on D-Wave™ cloud-based quantum computers)

Works

- M. Baioletti, F. Rossi, F. Santini A QUBO Approach to Value-Sensitive Reasoning in Argumentation Frameworks. AIQxQIA@ECAI2025
- M. Baioletti, F. Santini: An encoding of argumentation problems using quadratic unconstrained binary optimization. Quantum Mach. Intell. 6(2): 51 (2024)
- M. Baioletti, F. Rossi, F. Santini: Enumerating Extensions in Abstract Argumentation by Using QUBO. AIQxQIA@AI*IA 2024
- M. Baioletti, F. Santini: On using QUBO to enforce extensions in abstract argumentation (Short Paper). AIQxQIA@AI*IA 2023
- M. Baioletti, F. Santini: Abstract Argumentation Goes Quantum: An Encoding to QUBO Problems. PRICAI (1) 2022: 46-60

Abstract argumentation



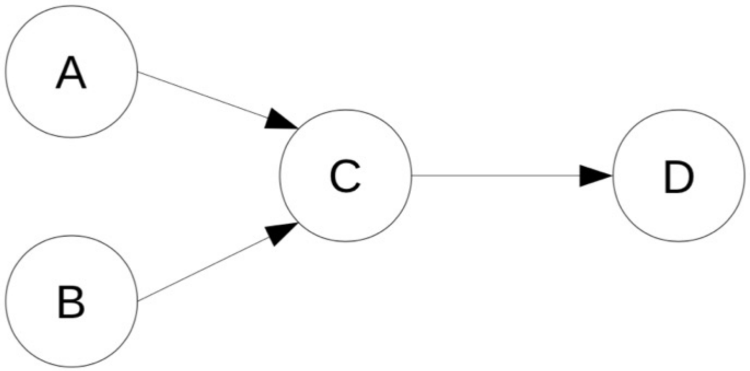
F = <**A**, **R**>

F = <{a,b,c,d}, {(a,c), (b,c), (c,d)}>

c attacks **d**

a defends **d** from **c**

Abstract argumentation



F = <**A**, **R**>

F = <{a,b,c,d}, {(a,c), (b,c), (c,d)}>

c attacks **d**

a defends **d** from **c**

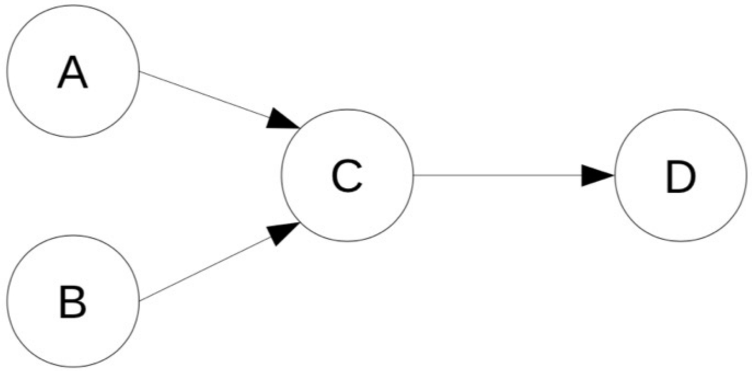
d is defended by {**a**, **b**} (or acceptable w.r.t. {**a**, **b**} in [Dung '95])

{**a**, **b**, **c**} is conflict-free

Semantics declaratively defined by elaborating on:

1. **Conflict-freeness**
2. **Acceptability**

Abstract argumentation: semantics



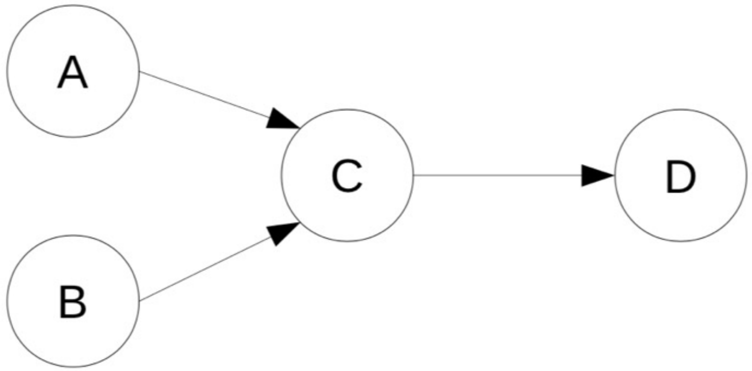
Admissible sets: {{}, {A}, {B}, {A,B}, {A,D}, {B,D}, {A,B,D}}

Complete: all acceptable arguments must be in. {{A,B,D}}

Preferred: maximal admissible w.r.t. set inclusion. {{A,B,D}}

Stable: conflict-free and attacks all arguments left outside. {{A,B,D}}

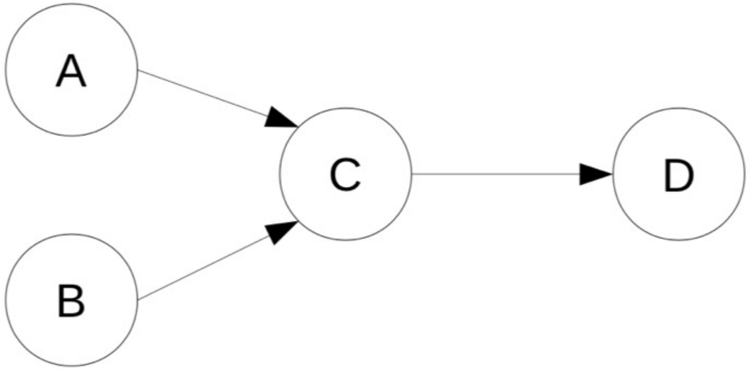
Semantics



Grounded: minimal complete w.r.t. set inclusion – unique. {A,B,D}

Or intersection of complete extensions

Semantics



Grounded: minimal complete w.r.t. set inclusion – unique. {A,B,D}

Or intersection of complete extensions

- In the following, {A,B,D} is an extension satisfying the grounded semantics
- σ will stand for a generic semantics
- E ∈ σ(F), means a subset of arguments E satisfy a semantics σ

Classical problems

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

Complexity

	Ver_σ	$Cred_\sigma$	$Scept_\sigma$	$Exists_\sigma$	$Exists_\sigma^{\neg\emptyset}$	$Unique_\sigma$	$Enum_\sigma$
Conflict-free	in L	in L	triv.	triv.	in L	in L	$DelayP_P$
Admissible	in L	**NP-c**	triv.	triv.	**NP-c**	coNP-c	nOP
Complete	in L	**NP-c**	P-c	triv.	**NP-c**	coNP-c	nOP
Preferred	coNP-c	**NP-c**	Π_2^P-c	triv.	**NP-c**	coNP-c	nOP
Semi-stable	coNP-c	Σ_2^P-c	Π_2^P-c	triv.	**NP-c**	in Θ_2^p	nOP
Stable	in L	**NP-c**	coNP-c	**NP-c**	**NP-c**	DP-c	nOP
Stage	coNP-c	Σ_2^P-c	Π_2^P-c	triv.	in L	in Θ_2^p	nOP
Grounded	P-c	P-c	P-c	triv.	P-c	triv.	DelayP
Ideal	Θ_2^p	Θ_2^p	Θ_2^p	triv.	Θ_2^p	triv.	nOP

Other approximate solvers

- International Competition on Computational Models of Argumentation (ICCMA)
  - Exact (SAT, ASP, CP, ad-hoc) and approximate solvers
  - 2015, 2017, 2019, **2021**, 2023(?), …

- **Harper++** (by M. Thimm). A positive answer to DS-gr implies a positive answer to DC and DS for σ ∈ {co,st,pr,sst,*stg*,*id*}.

- **AFGCN** (by L. Malmqvist) uses a Graph Convolutional Network, to compute approximate solutions to DC and DS tasks for σ ∈ {co,st,pr,sst,*stg*,*id*}.

QUBO

- QUBO is an NP-Complete Combinatorial Optimization Problem: a vast literature is dedicated to approximate solvers based on heuristics or meta-heuristics, such as *simulated annealing* approaches, *tabu-serch*, *genetic algorithms* or *evolutionary computing*.
- There exist also exact methods (100-500 variables).

- QUBO encompasses SAT Problems, Constraint Satisfaction Problems, Maximum Cut Problems, Graph Coloring Problems, Maximum Clique Problems, General 0/1 Programming Problems and many more

QUBO

- A solution of a QUBO problem simply corresponds to minimize a quadratic function over binary variables (0/1)
  - Coefficients represented with a symmetric square matrix

$$\begin{bmatrix} -5 & 2 & 4 & 0 \\ 2 & -3 & 1 & 0 \\ 4 & 1 & -8 & 5 \\ 0 & 0 & 5 & -6 \end{bmatrix}$$

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

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

QUBO

Minimize y = $-5x_1$ $-3x_2$ $-8x_3$ $-6x_4$ $+4x_1x_2$ $+8x_1x_3$ $+2x_2x_3$ $+10x_3x_4$

Linear $-5x_1$ $-3x_2$ $-8x_3$ $-6x_4^2$
and quadratic $4x_1x_2$ $+8x_1x_3$ $+2x_2x_3$ $+10x_3x_4$

Since $x_j = x_j^2$ the linear part can be rewritten as $-5x_1^2$ $-3x_2^2$ $-8x_3^2$ $-6x_4^2$

$$\text{Minimize } y = \begin{pmatrix} x_1 & x_2 & x_3 & x_4 \end{pmatrix} \begin{bmatrix} -5 & 2 & 4 & 0 \\ 2 & -3 & 1 & 0 \\ 4 & 1 & -8 & 5 \\ 0 & 0 & 5 & -6 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix}$$

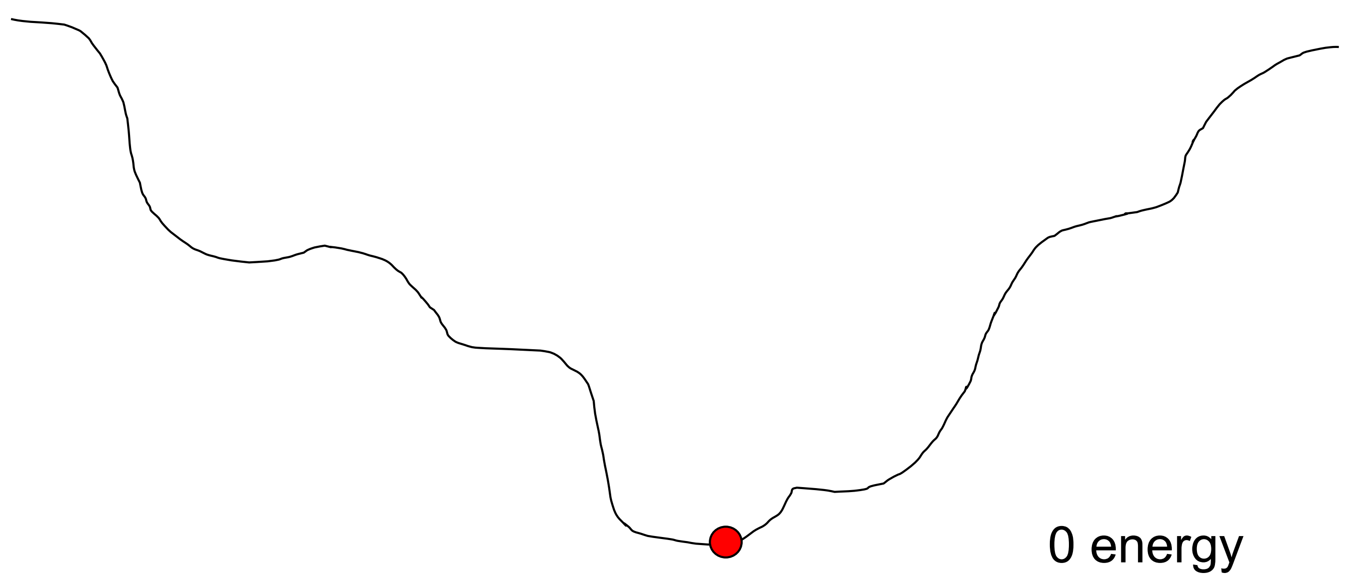
Example

$$Q = \begin{pmatrix} 2 & -1 & 4 \\ -1 & 5 & -2 \\ 4 & -2 & -1 \end{pmatrix} \quad \text{(symmetric)}$$

$$U = \begin{pmatrix} 2 & -2 & 8 \\ 0 & 5 & -4 \\ 0 & 0 & -1 \end{pmatrix} \quad \text{(upper triangular)}$$

Finding and encoding

- 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)



An example of encoding

$$P_{cf} = \sum_{i \to j \text{ or } j \to i} x_i x_j$$

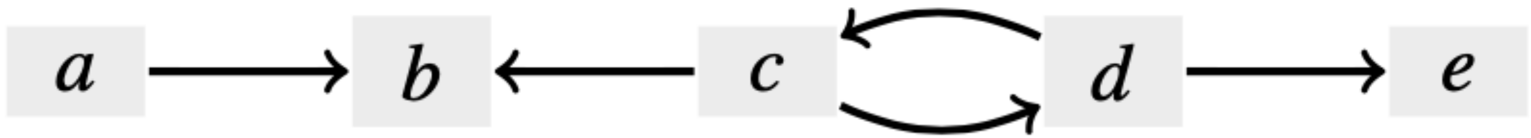
$$P_{adm} = P_{cf} + \sum_{i=1}^{n} P_t^i + \sum_{i=1}^{n} P_d^i + P_{def}$$

$$P_{co} = P_{adm} + \sum_{i=1}^{n} (1 - x_i) d_i$$

An example of encoding

$$P_{cf} = \sum_{i \to j \text{ or } j \to i} x_i x_j$$

$$P_{cf} = x_1 x_2 + x_3 x_2 + x_3 x_4 + x_4 x_5.$$



Defence

- Defense is not one constraint, but a composition of three layers:
  - Conflict constraints
  - Attack-tracking constraints (“who is attacked by E?”)
  - Defense-enforcement constraints (“you can select only defended arguments”)

- Or comnposition for attacks

$$t_i = \bigvee_{j \to i} x_j$$

Defence

- And composition for defences

$$d_i = \bigwedge_{j \to i} t_j$$

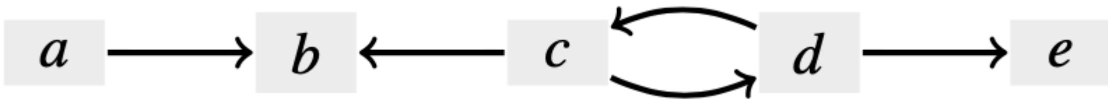
- Enforce defense (the final glue) to you connect defense back to selection.
- if $x_i = 1$ (you choose argument a_i) but $d_i = 0$ (it is not defended) then the penalty is positive
  - $x_i(1-d_i)$

Complete extensions

- For each argument a_i, add: $(1 - x_i)\, d_i$

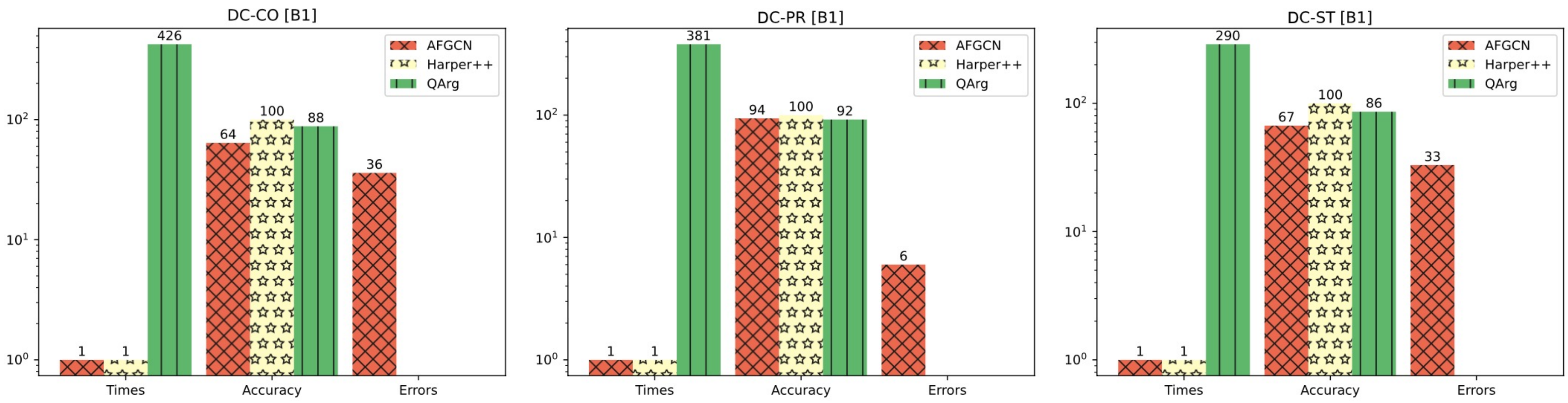
(x_i)	(d_i)	penalty
1	1	0
1	0	0
0	0	0
0	1	**> 0 (forbidden)**

Complete



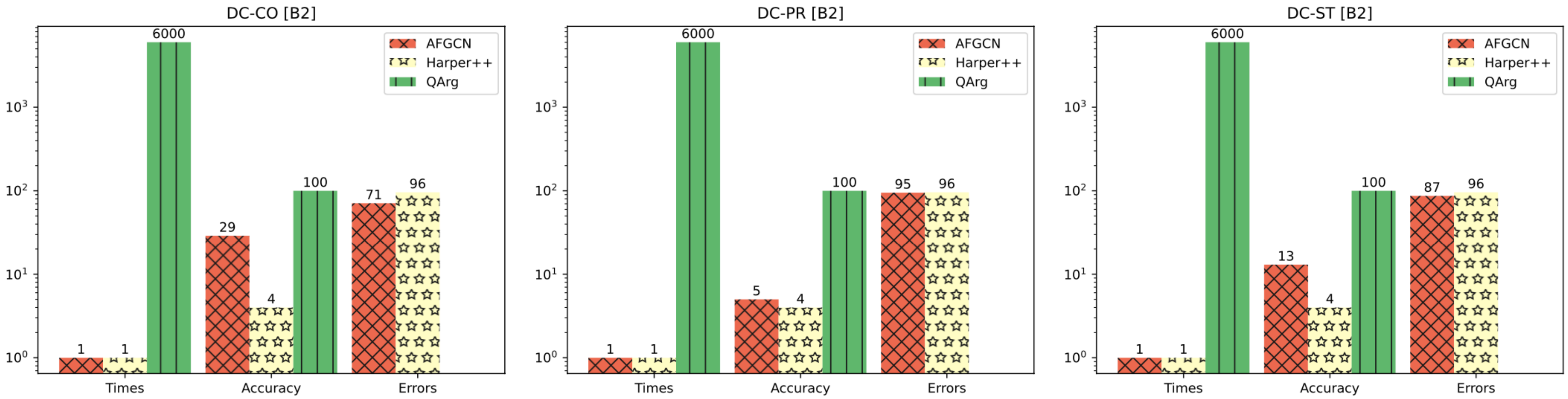
$$P_{co} = x_1x_2 + x_2x_3 + x_3x_4 + x_4x_5 + x_2 + x_5 + x_3 - 2x_3x_5 + 1 - x_1$$

Tests using Simulated Annealing



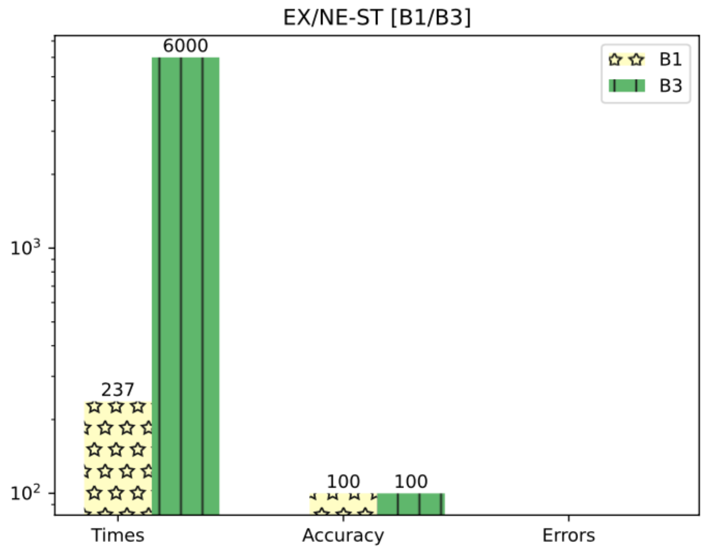
Tested on 100 –YES– problems taken from ICCMA19 benchmark

Tests

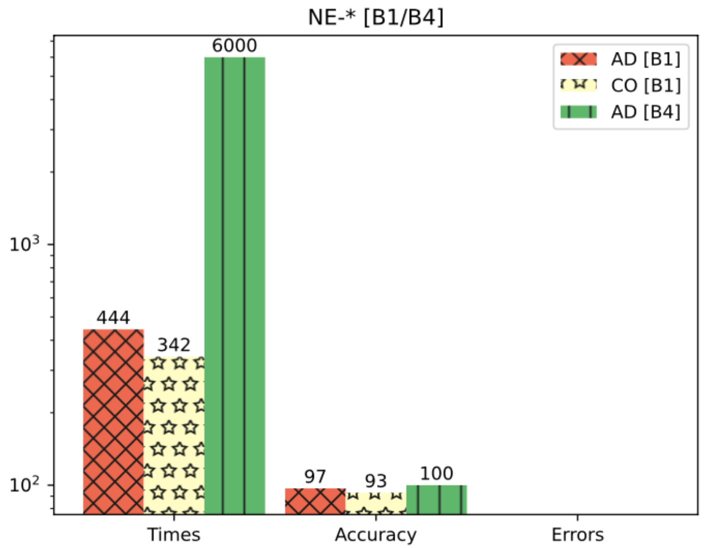


Tested on 100 –NO– problems taken from
ICCMA19 benchmark

Tests



B1 = 100 –YES–
B2 = 100 –NO–
Harper++ and AFGCN do not solve these problems



B1 = 100 –YES–
B2 = 100 –YES–
B4 = 100 –NO–
Harper++ and AFGCN do not solve these problems

Solving it with a quantum annealer

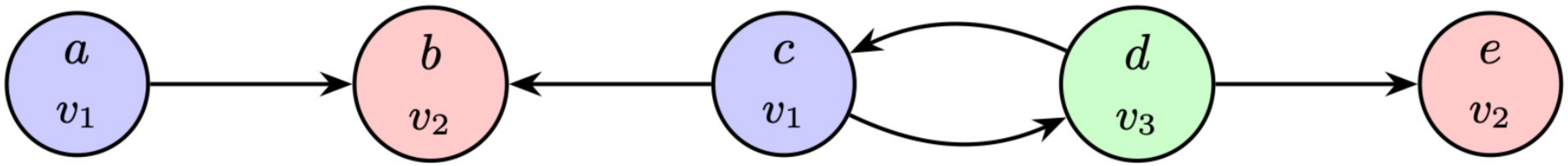
Leap™ Quantum Cloud Service

```
def dwave_sol(model):
  qubo, qubo_offset = model.to_qubo()
  sampler_kwargs = {"num_reads": 50, "annealing_time": 50, "
       num_spin_reversal_transforms": 4, "auto_scale": True, "chain_strength":
       2.0, "chain_break_fraction": True}
  dw_sampler = DWaveSampler(endpoint="https://cloud.dwavesys.com/sapi", token=
       MYTOKEN, solver=MYSOLVER)
  sampler = EmbeddingComposite(dw_sampler)
  sampleset = sampler.sample_qubo(qubo)
  decoded_samples = model.decode_sampleset(sampleset)
  best_sample = min(decoded_samples, key=lambda x: x.energy)
  return best_sample
```

Value-based AFs

- SBA is NP-Complete

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.



Conclusion and FW

- Conclusion
  - A new approximate approach to solve problems in Argumentation
  - To be used with Simulated Annealers and Quantum annealers
  - We solved this kind of problems with Quantum machines for the first time
  - Good results when compared to related algorithms

Conclusion and FW

- Conclusion
  - A new approximate approach to solve problems in Argumentation
  - To be used with Simulated Annealers and Quantum annealers
  - We solved this kind of problems with Quantum machines for the first time
  - Good results when compared to related algorithms

- Future work
  - Finding new Quantum machines
  - Ad-hoc Simulated Annealers
  - More optimization problems than numerical problems

Contacts:

francesco.santini@unipg.it