>>AI CONF 2026

Quantum game theory - new approaches to optimization

Davide Colazingari

Senior Software Developer
Crif S.p.a.

improove

Kudos++

Executive

beSharp. we make IT run.

LEXROOM

REPLY STORM

Gold

seeweb

FASTWEB + vodafone

>>AI CONF 2026

improove

Motivation

- Quantum computing (QC) research grew from physics: Feynman (1982) suggested using quantum machines to efficiently simulate quantum systems.
- QC can outperform classical computing on specific problems — the idea of "quantum advantage".
- Relevant problem classes: simulation of Hamiltonians (e.g. quantum chemistry), combinational optimization (e.g. scheduling), and machine learning primitives.
- Scope of the talk: how quantum game-theoretic models and quantum optimization methods offer new approaches to hard optimization tasks.

Classical bit vs quantum bit (qubit)

- Classical state: an element of a finite set $S = \{\clubsuit, \diamondsuit, \spadesuit, \heartsuit\}$
- Classical bit: $S = \{0, 1\}$, abstracts analog phenomena
- Quantum state: a linear combination of elements of S of length 1

$$|\psi\rangle = \alpha_1 |\clubsuit\rangle + \alpha_2 |\diamondsuit\rangle + \alpha_3 |\spadesuit\rangle + \alpha_4 |\heartsuit\rangle , \qquad \alpha_i \in \mathbb{C},\ \sum_i |\alpha_i|^2 = 1$$

- Qubit: a two-dimensional complex vector of length 1, abstracts quantum phenomena

$$|\psi\rangle = \alpha |0\rangle + \beta |1\rangle , \qquad \alpha, \beta \in \mathbb{C},\ |\alpha|^2 + |\beta|^2 = 1$$

Logical vs physical qubits

- Logical qubit: the abstract ideal qubit used in algorithms and theory.
- Physical qubit: the actual implementation (trapped ions, spin systems, ...).
- Noise and decoherence affect physical qubits; quantum error correction encodes one logical qubit into many physical qubits.
- Current devices are noisy intermediate-scale quantum (NISQ): limited qubit counts, imperfect gates. Loading the initial state is not trivial.
- In this talk: we will always talk about logical qubits, while keeping in mind realistic expectation of current and near-term technology.

Tensor product and multi-qubit states

- The classical state $S_1 \times S_2$ described by the cartesian product becomes the tensor product $S_1 \otimes S_2$ between the respective quantum state spaces.
- Example: $\{\clubsuit, \diamondsuit, \spadesuit, \heartsuit\} \otimes \{1, \ldots, 10, J, Q, K\}$
- Multi-qubit state space grows exponentially: n qubits live in a 2^n-dimensional tensor space.
- Product state: $|\psi\rangle \otimes |\phi\rangle$, often abbreviated $|\psi\rangle\,|\phi\rangle$ or $|\psi\phi\rangle$.
- Example: $|\clubsuit K\rangle = |\clubsuit\rangle \otimes |K\rangle$.
- Exponential state-space is a source of both power and difficulty (simulation cost on classical machines).

Measurement, collapse and entanglement

- Measurement projects a qubit onto a basis. The qubit $\alpha|0\rangle + \beta|1\rangle$ will be measured as 0 with probability $|\alpha|^2$.
- After measurement result 0, the post-measurement state is $|0\rangle$ (collapse); α and β cannot be accessed directly
- If the system is in the state $\alpha|01\rangle + \beta|10\rangle + \gamma|11\rangle$, the first qubit would be measured as 1 with probability $|\beta|^2 + |\gamma|^2$
- Yet, if the second qubit is measured as 0 (with probability $|\beta|^2$), then the first qubit is measured as 1 with certainty.
- This is an example of entangled state, i.e. the state cannot be written as a pure tensor $|\phi\psi\rangle$, where ϕ and ψ are classical states.
- Entanglement is a fundamental resource in most quantum algorithms.

Unitary evolution and Hamiltonian

- Quantum gates = unitary operators U with $U^\dagger U = I$ (reversible).
- Time evolution is generated by the Hamiltonian H (Hermitian operator) - you might recognize Schrödinger equation!:

$$i\hbar \frac{d}{dt} |\psi(t)\rangle = H |\psi(t)\rangle \,, \qquad |\psi(t)\rangle = e^{-iHt/\hbar} |\psi(0)\rangle \,.$$

- Many problems map to finding ground states or low-energy properties of a Hamiltonian (quantum chemistry, combinatorial optimization via Ising-type Hamiltonians).

What "computing" means in QC

- Circuit model: initialize qubits, apply sequence of gates (unitaries), measure outputs.
- Adiabatic/annealing model: encode problem into a time-dependent Hamiltonian and evolve slowly to the ground state.
- Variational (hybrid) algorithms: classical optimizer chooses parameters for a quantum circuit; quantum device evaluates cost function.
- Output is probabilistic — algorithms typically require many runs to estimate probabilities/expectation values.

Complexity and quantum advantage

- BQP: class of problems efficiently solvable on a quantum computer (bounded-error quantum polynomial time).
- Known quantum speedups:
  - Shor's algorithm: exponential speedup for integer factorization (impact on crypto).
  - Grover's algorithm: quadratic speedup for unstructured search.
  - Quantum simulation: exponential benefits expected for many quantum many-body problems.
- Caveats: quantum advantage is problem-dependent and requires careful resource accounting (qubits, depth, error rates).
- Polynomial speedups, while theoretically interesting, might not be sufficient to be relevant to the industry.

Four approaches (CC / CQ / QC / QQ)

- CC — Classical data / Classical computing
  - Classical algorithms that operate on classical data.
  - This includes quantum-inspired ML methods derived from research in quantum computing.
- CQ — Classical data / Quantum computing
  - We are here!
- QC — Quantum data / Classical computing
  - Classical algorithms process data that originate from quantum experiments (state tomography, property estimation).
  - Active area of research, exponentially complex.
- QQ — Quantum data / Quantum computing
  - Fully quantum workflows: quantum states processed by quantum circuits.
  - *Research frontier*
  - Speculation: quantum data might arise from economic data (e.g. AML).

“Practical” guidance for choosing an approach

- Start with CC (or quantum-inspired) to set performance baselines and understand problem structure.
- Use CQ (hybrid) when a Hamiltonian/Ising mapping or linear-algebra kernel suggests potential quantum advantage and when short-depth circuits suffice.
- Explore QC when working with quantum experiments or when preprocessing/classification of quantum-generated data is needed.
- Consider QQ for theoretical exploration of quantum strategies; real-world applications are still far.

Game theory — informal introduction

- Game theory studies strategic interaction between agents (players) whose outcomes depend on each other's choices.
- Two broad perspectives:
  - Cooperative (collaborative): players can form binding agreements / coalitions; focus on value division and fair allocation.
  - Non-cooperative (adversarial / strategic): players act individually and strategically; focus on equilibria (e.g., Nash equilibrium).

- A normal-form game = $(\mathrm{N}, (\mathrm{S}_i)_{i \in N}, (u_i)_{i \in N})$:
  - N = set of players;
  - S_i = finite set of strategies for player i
  - $u_i : S_1 \times \cdots \times S_n \to \mathbb{R}$ = payoff for player i.

	L	R
U	$(3, 2)$	$(0, 0)$
D	$(1, 1)$	$(2, 4)$

Key concepts: Nash equilibrium: no player can unilaterally increase their payoff by deviating.

Question

If we consider the features of an optimization problem as players, are you aware of a game-theoretic method in ML?

SHAP — game theory meets ML explainability

- Problem: How much does each feature contribute to a model's prediction?
- Game-theoretic answer: Apply the **Shapley value** from cooperative game theory.

$$\phi_i(v) \;=\; \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N|-|S|-1)!}{|N|!} \left(v(S \cup \{i\}) - v(S)\right)$$

  where players = features; $v(S)$ = expected model output when only features in S are known.
- Intuition: each feature's contribution is its **average marginal contribution** across all orders in which features can be added.

SHAP — algorithms and considerations

- Exact computation is exponential ($2^{|N|}$ coalition evaluations) — approximations required:
  - KernelSHAP: model-agnostic estimator (weighted regression on random coalitions).
  - TreeSHAP: exact and efficient for tree ensembles (e.g., Random Forest, XGBoost).
  - DeepSHAP, PartitionSHAP: gradient-based approximations for neural networks.
- Key design choice: how to define the value function $v(S)$?
  - Marginal expectation: $v(S) = \mathbb{E}[\text{model}(S, \text{random other features})]$ — fast but can inflate importance of correlated features.
  - Conditional expectation: $v(S) = \mathbb{E}[\text{model}(S) \mid \text{context}]$ — more principled but computationally harder.

QUBO — quadratic unconstrained binary optimization

- Canonical form (minimization):

$$\text{minimize} \quad f(x) = x^\top Q x \quad \text{subject to} \quad x \in \{0,1\}^n,$$

  where Q is an $n \times n$ real (typically symmetric) matrix.
- Many combinatorial problems map to QUBO: graph coloring, scheduling, feature selection, and portfolio optimization.
- Complexity: QUBO is NP-hard in general; exact solution scales poorly with n.
- Relevance to quantum: QUBO $\iff$ Ising spin model (change of variables: $s_i \in \{-1, +1\}$, $x_i = (1 + s_i)/2$) used in quantum annealers.

QUBO $\iff$ Ising model and potential games

- Link to game theory: interpret each binary variable x_i as a player choosing strategy 0 or 1.
- A *potential game* has a potential function Φ such that any unilateral player deviation changes payoff by the same amount as Φ changes:

$$u_i(a'_i, a_{-i}) - u_i(a_i, a_{-i}) = \Phi(a'_i, a_{-i}) - \Phi(a_i, a_{-i}).$$

- For QUBO (with appropriate payoff construction), the objective $f(x)$ is a potential function.
- Consequences:
  - Pure-strategy Nash equilibria = coordinate-wise local optima of $f(x)$ (no single variable flip improves it).
  - Local search / best-response dynamics converge to a Nash equilibrium (a local optimum).
  - Global optimizers (e.g. quantum annealing) must escape local optima to find the global minimum.

Credit scoring — classification vs regression

- Classification: predict discrete outcome (default / no-default). Typical output used for decisioning and accept/decline thresholds.
- Regression / scoring: estimate continuous quantity (e.g. probability of default, loss given default). Allows ranking, PD curves and portfolio aggregation.
- Common modelling steps: feature engineering, handling imbalance, cross-validation, calibration and explainability checks.

German Credit Data — small benchmark

- Instances: 1,000 (classic, often used for teaching and benchmarking).
- Features: 20 original (7 numeric, 13 categorical, e.g., credit history, purpose, amount, duration, age, employment).
- After one-hot encoding: $\sim$48 binary feature variables.
- Label: binary credit risk (good / bad).
- Use case: quick prototyping, feature selection exploration, model explainability demos.

Taiwan Credit Card Default — larger, realistic scale

- Instances: ~30,000 clients (modern benchmark, more realistic size).
- Features: demographics, credit limit, 6 months of repayment status (ordinal), monthly bill amounts, monthly payment amounts — ~22–24 variables.
- Label: default payment next month (binary, imbalanced ~22% positives).
- Use case: realistic scale for ML experimentation; temporal structure in repayment history enables engineered features.
- Key insight: **dataset scale matters**. Classical deep learning (neural nets) dominates on German (small); on Taiwan, classical ML (tree ensembles) competes or wins.

Dataset comparison — why Taiwan is instructive

- German Credit (1k samples, 20 features):
  - Small dataset $\rightarrow$ deep learning overfits easily.
  - Feature selection is valuable to reduce noise and improve interpretability.
  - Tree ensembles (RF, XGB) and logistic regression often win.
- Taiwan Credit (30k samples, 22–24 features):
  - Larger dataset and realistic imbalance $\rightarrow$ deep learning can leverage scale.
  - Yet classical ML (XGBoost, Random Forest) often **matches or exceeds** neural net performance on AUC, often with better calibration.
  - Imbalanced classification is hard: sampling, reweighting, and threshold tuning are critical.
- Takeaway for the talk: not all performance gains come from fancier models; problem structure and dataset characteristics matter. QUBO-based feature selection is most relevant on structured, interpretable problems (like credit scoring).

Case study: Quantum annealer for feature selection (Milne et al., 2017)

- Problem: Select a subset of features that are both **influential** (correlated with target) and **independent** (low cross-correlation).
- Approach: Formulate as QUBO with parameter α balancing influence vs. independence.

$$f(x) = -\alpha \sum_{j} x_j |\rho_{Vj}| - (1-\alpha) \sum_{j \neq k} x_j x_k |\rho_{jk}|$$

- Dataset: German Credit (1000 samples, 20 features $\rightarrow$ 48 after binarization).
- Solver: Quantum annealer (D-Wave) + logistic regression classifier.
- Key advantage: QUBO naturally encodes both objectives simultaneously; classical alternatives (RFE) optimize sequentially.

QUBO formulation for feature selection

- Binary variables: $x_j \in \{0, 1\}$ indicates whether feature j is selected (presence in the model).
- Objective combines two desiderata: **influence** (high correlation with label) and **independence** (low pairwise correlation between features).
- Using Spearman correlations — let ρ_{Vj} = corr(feature j, label), and ρ_{jk} = corr(feature j, feature k):

$$f(x) \; = \; -\alpha \sum_{j} x_j |\rho_{Vj}| - (1-\alpha) \sum_{j \neq k} x_j x_k |\rho_{jk}|$$

  Negated form for minimization; parameter $\alpha \in [0, 1]$ trades off influence vs independence.
- Cardinality control (optional): add penalty $\lambda(\sum_j x_j - K)^2$ to encourage subsets of size K.
- This is a true QUBO (quadratic in binary variables), suitable for quantum annealers and QAOA.

Experimental setup

- Preprocessing:
  - German Credit: 20 features $\rightarrow$ one-hot binarization $\rightarrow$ $\sim$48 binary feature variables.
  - Numeric features scaled (z-score). Categorical groups encoded as $k-1$ indicators.
  - Label: $0 =$ good, $1 =$ bad credit.
- Solver: quantum annealer (D-Wave) and classical heuristics (RFE, random search).
- Classifier: logistic regression (wrapper) trained on selected features and evaluated on held-out test set.
- Evaluation protocol:
  - StratifiedShuffleSplit cross-validation with many shuffles (1000–3000) and 20% test share.
  - Report mean accuracy and F1 score with 95% confidence intervals.

Experimental results — German Credit

Method	Accuracy	F1 score	# Features
QUBO Feature Selection	0.764 ± 0.049	0.54 ± 0.1	24
RFE (target 28)	0.767 ± 0.050	0.55 ± 0.1	28
RFECV	0.764 ± 0.050	0.54 ± 0.1	31
Random (10k samples)	0.762 ± 0.050	0.54 ± 0.1	35

Key observation: accuracies are comparable; **QUBO returns the smallest subset** (24 features vs 28–35) without sacrificing accuracy.

Results (cont.) — Local optima and sensitivity

- Role of α parameter:
  - As α increases (favoring influence over independence), subset size increases.
  - Small α favors feature independence but can collapse to trivial solutions ($\sim$70% baseline accuracy).
  - Empirical peak: $\alpha \approx 0.977$ achieves highest accuracy (24 features).
- Local search robustness: QUBO solutions show resistance to small perturbations (one/two feature flips), indicating a good local optimum in the search landscape.
- Implication: QUBO approach balances influence and independence in a principled way; multimodal landscape suggests multiple good subsets exist, hence careful CV is essential.

Key takeaway from the case study

- The paper demonstrates a concrete **QUBO encoding for feature selection** and proves a **proof-of-concept on German Credit Data**: smaller subset (24 features) with comparable accuracy.
- This is **not a proof of quantum advantage**; rather, an exploration of QUBO as a principled modelling tool and the potential of hybrid (quantum + classical) solvers for combinatorial problems.
- Performance is modest on a small dataset — gains are meaningful but **hardware-dependent and problem-specific**.
- One (non-peer reviewed!) experiment shown empirically that further subdivision of the QUBO might have an exponential speedup over classical methods for certain specific QUBO problems.

Takeaways and next steps

- Game theory provides conceptual tools for optimization: **potential games** illuminate why local search gets stuck at local optima.
- SHAP shows how game-theoretic axioms (Shapley value) solve real ML problems (feature attribution); a bridge between abstract theory and practice.
- Quantum and hybrid optimization methods (QUBO, QAOA, annealers) offer new angles on combinatorial problems, but gains are problem-dependent and require careful benchmarking.

Further reading

- Classical textbooks: Nielsen&Chuang ('90); Manenti&Motta (2022)
- Quantization schemes for games ('90):
    - EWL: https://doi.org/10.1103/PhysRevLett.83.3077
    - Meyer: https://doi.org/10.1103/PhysRevLett.82.1052
    - Marinatto & Weber: https://doi.org/10.1016/S0375-9601(00)00441-2
- Credit risk:
    - Mapfumo & Shongwe (review of classical methods, 2026): https://doi.org/10.3390/jrfm19030210
    - Optimal feature selection in credit scoring and classification using a quantum annealer (white paper on quantum annealers): https://1qbit.com/whitepaper/optimal-feature-selection-in-credit-scoring-classification-using-quantum-annealer/
- Quantum game theory: "Mapping the frontier" by Sanz-Martín et al. (2025, has hallucinated citations): https://doi.org/10.1007/s11128-025-04913-4
- Quantum Computing – Strategic Recommendations for the Industry (2026): https://arxiv.org/abs/2601.08578v1

Enjoyed the talk?

Scan the QR code to rate this session

Connect with me:

GitHub: /dcolazin **LinkedIn** **Website**

Questions?

Thank you!

👉 slides & videos: https://www.improove.tech/videos

>>AI CONF 2026

improove