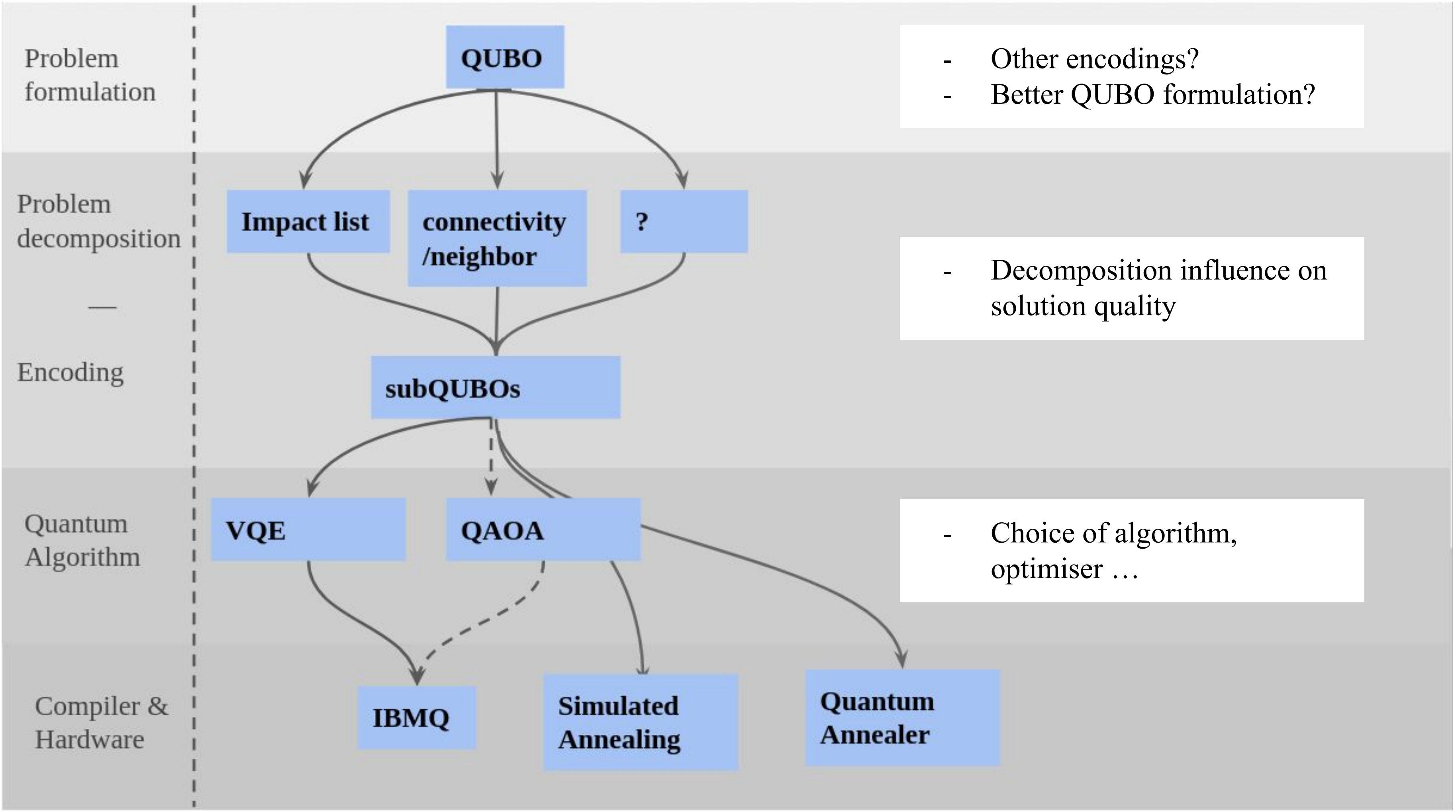
Quantum Computing Meeting

15.05.2023


Problem formulation
QUBO
- Other encodings?
- Better QUBO formulation?
Problem decomposition
—
Encoding
Impact list
connectivity /neighbor
?
- Decomposition influence on solution quality
subQUBOs
Quantum Algorithm
VQE
QAOA
- Choice of algorithm, optimiser …
Compiler & Hardware
IBMQ
Simulated Annealing
Quantum Annealer

QUBO: learned couplings?

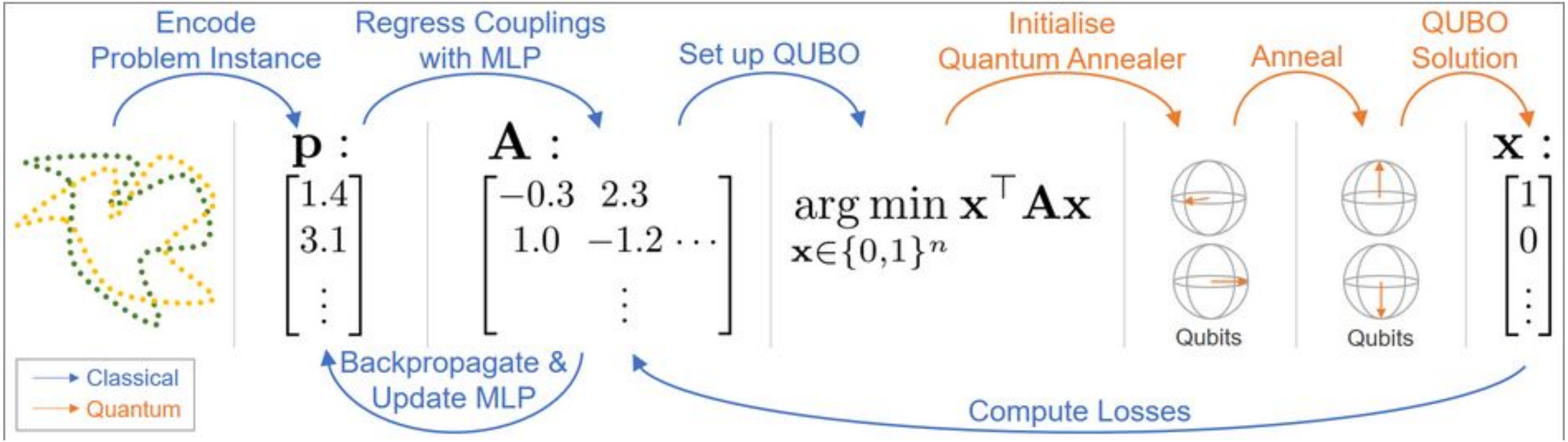
QuAnt: Quantum Annealing with Learnt Couplings

Marcel Seelbach Benkner[1] Maximilian Krahn[2,3] Edith Tretschk[2] Zorah Lähner[1] Michael Moeller[1] Vladislav Golyanik[2]

[1]University of Siegen [2]Max Planck Institute for Informatics, Saarland Informatics Campus [3]Aalto University

ICLR 2023 (Spotlight / Top 25%)

[Link to paper website]



Possible approaches for QUBO ML

Simulated annealing is fully implemented and can be used for the ML approach

- **QUANT paper:**
  - For each triplet, compute features (angles, ..)
  - Concatenate the features for each triplet into a single feature vector per triplet
  - Normalise and use similarity to compare the the feature vectors of different triplets /w each other
  - Flatten similarities into 1D description vector
  - Use description vector as input for ML → qubo encoding is learned
- **“Direct” supervised learning:**
  - We start with standard QUBO encoding (one triplet to one qubit)
  - QUBO matrix entries capture the features of triplets
  - How much individual features are weighted is learned, ie qubo entries are learned

Problem: creating QUBO is costly

Since we look at all triplet combinations, our complexity is ~ CKF

Proposal: hybrid approach

Only apply QUBO for highly difficult tracking area (complicated clusters) in parallel to CKF.

Resulting ambiguities are resolved classically afterwards. Questions arising:

1.) How do we measure the complexity of a cluster?
2.) When does a cluster become highly non-trivial, ie at what point should QUBO be used?
3.) Do we see an advantage in solution quality and/or time complexity if QUBO is used?

Proposal: use potential good/bad tracks inside the cluster to assess complexity

- Each hit is associated with a single, unique particle as of now.
- Assess the number of "bad" potential tracks within a cluster as part to evaluate its complexity (ie how many “branches” starting triplets have).
    - Measures to consider:
        - Average number of fake tracks
        - Conflicts to resolve
        - True-to-fake ratio
        - …
- Alternative approach:
    - Evaluate the area to resolve based on particle density instead of cluster information.
    - ..

Can the learned QUBO help with annealing errors?

- If part of the errors in the quantum annealer have consistent patterns, it is possible that the ML approach could learn to adjust the QUBO entries in a way that is more resilient to those errors.
- Counterargument: Errors might be too complex and difficult to predict, and might vary to much from run to run.
- One way to compare the performance of different approaches for mitigating errors could be to use simulated annealing with added errors.

Summary and Questions: context simulated/quantum annealing

QUBO approach can be used for complex problems inside detector, with CKF handling most cases. ML the entries or the encoding might assist with a more refined QUBO approach and may help with some errors.

- How do we classify cluster complexity?
- Is ML a suitable candidate (learned encoding or learned entries?) as a next step?
- What measure (solution quality, speedup, scalability …) can be applied to evaluate the ML approach?
- Are annealing errors too complex to be counteracted using ML?