1st Working Symposium on Quantum Software Engineering

Programming Quantum Annealers

Dr. Elisabeth Lobe

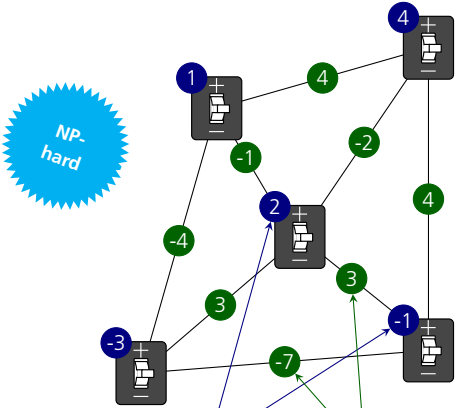
High-Performance Computing
Institute for Software Technology
DLR German Aerospace Center

15th December 2022

DLR

Optimizer for Ising Problems

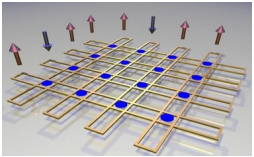
- Minimization
  - over spin variables
  - of a quadratic objective function
  - without further constraints
- "Programming"
  - providing the weights $W \in \mathbb{R}^V$ and strengths $S \in \mathbb{R}^E$ for graph $G = (V, E)$



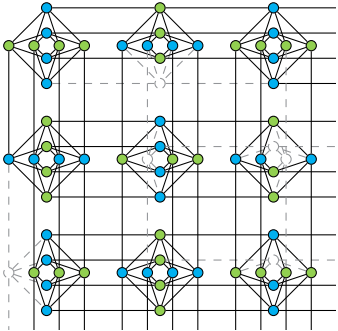
$$\min \sum_{v \in V} W_v \, s_v + \sum_{vw \in E} S_{vw} \, s_v s_w$$

$$\text{s.t. } s \in \{-1, +1\}^V$$

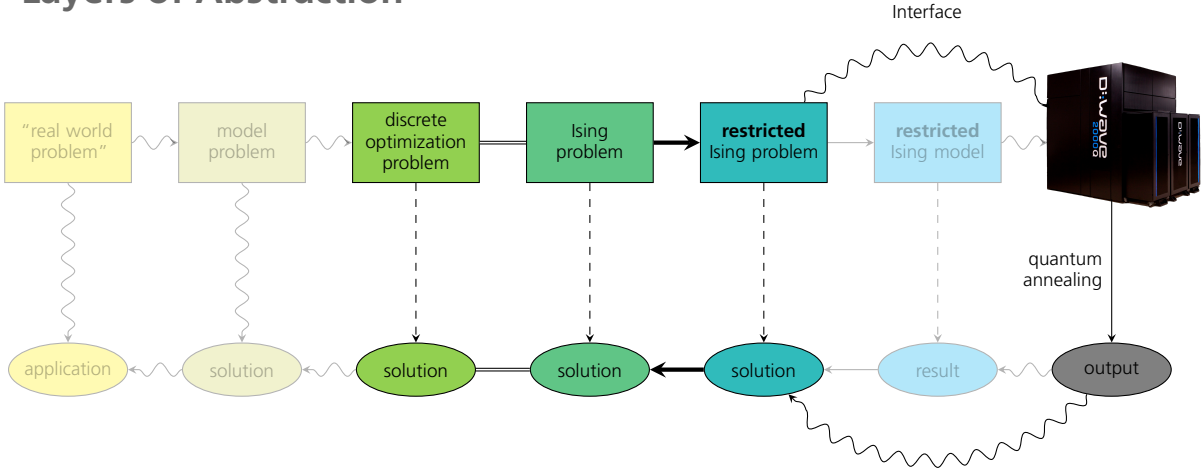
Restricted Hardware



- No ideal realization of underlying physical concept
  - heuristic solver
  - no guarantee for optimal solution
- Hardware structure of the quantum chip
  - formed by overlapping superconducting loops
  - realizes specific graphs
    (Chimera in `DW2000Q`, Pegasus in `Advantage5`)
  - with certain "broken" qubits due to calibration
- Resolution of input parameters
  - limited due to analog control circuits
  - parameter distribution decisively influences success probability



Layers of Abstraction



Goals

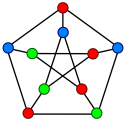
- Easily reproducible experiments
  - reformulate arbitrary optimization problems to Ising problems
  - by automated transformation steps
  - based on a parameterized formulation of the problem instances
  - generate several problem instances with the same structure
  - store or load intermediate data at every stage of the transformation process
- Allow analysis of the machine behaviour and experimental results
  - provide hints whether problems are suitable to be solved with the annealers
  - provide solutions of classical solver for comparison
- Modularity for flexible usage
  - adapt to different hardware architectures and thus restrictions
  - base for further algorithmic implementations

Our Software Library

Application Layer

problem parameters →

quapps



$$\min \sum_{c \in C} \sum_{nm \in E} x_{nc} x_{mc}$$

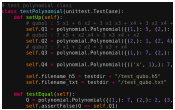
$$\text{s.t.} \sum_{c \in C} x_{nc} = 1 \ \forall n \in N,$$

$$x \in \{0,1\}^{N \times C}$$

Transformation Layer

problem description (cost function, constraints) →

quark



```
class TestPolynomial(unittest.TestCase):
    def setUp(self):
        # cost = 3 * x + 6 x2 + 1 x1 x2 + x4 + 1 X2 x4 +
        self.Q1 = polynomial.Polynomial({(1,): 3, (2,(
        # cost = 3 * x1 x2 + 6 x1 x2 +
        self.Q2 = polynomial.Polynomial({(1,2): 1, (2, 4)
        # cost = x1 x2 + x3 +
        self.Q3 = polynomial.Polynomial({(1,): 7, (2,3)

        self.Q4 = polynomial.Polynomial({('a', 1),1): 7,

        self.filename_h5 = testdir + "/test_qubo.h5"
        self.filename_txt = testdir + "/test_qubo.txt"

    def test_Equal(self):
        Q = polynomial.Polynomial({(1,): 7, (2,): 2, (3,
        self.assertFalse(Q == self.Q1)
```

$$\min \sum_{v \in V} W_v s_v + \sum_{vw \in E} S_{vw} s_v s_w$$

$$\text{s.t.} \ s \in \{-1,+1\}^V$$

QC Interface Layer

restricted Ising problem →

← check validity

quokka

QA

roque

QAOA

Questions?

Dr. Elisabeth Lobe

High-Performance Computing
Institute for Software Technology
DLR German Aerospace Center

elisabeth.lobe@dlr.de

Code

`https://gitlab.com/quantum-computing-software/quark`

DLR