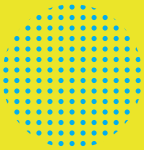
Hybrid Solver for Hard Optimization Problems

Timotej Hrga[1] Roman Kužel[1] Janez Povh[1] Beno Zupanc[1]

[1]Rudolfovo – Science and Technology Centre Novo Mesto, Slovenia

November 5th, 2026

RUDOLFOVO
Znanstveno
in tehnološko
središče
Novo mesto



A Motivating Problem: Which Projects to Fund?

We have five research projects, but limited resources.

Project	A	B	C	D	E
Benefit	8	6	5	7	4
Cost	4	3	2	4	2

For every project, introduce a binary decision:

$$x_i = \begin{cases} 1, & \text{if project } i \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases}$$

A Motivating Problem: Which Projects to Fund?

We have five research projects, but limited resources.

Project	A	B	C	D	E
Benefit	8	6	5	7	4
Cost	4	3	2	4	2

Interactions matter

- Projects A and C create a synergy of $+3$.
- Projects B and D compete, causing a penalty of -4.

For every project, introduce a binary decision:

$$x_i = \begin{cases} 1, & \text{if project } i \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases}$$

A Motivating Problem: Which Projects to Fund?

We have five research projects, but limited resources.

Project	A	B	C	D	E
Benefit	8	6	5	7	4
Cost	4	3	2	4	2

For every project, introduce a binary decision:

$$x_i = \begin{cases} 1, & \text{if project } i \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases}$$

Interactions matter

- Projects A and C create a synergy of $+3$.
- Projects B and D compete, causing a penalty of -4.

$$\begin{aligned} \max \quad & 8x_A + 6x_B + 5x_C \\ & + 7x_D + 4x_E \\ & + 3x_A x_C - 4x_B x_D \end{aligned}$$

Key observation

The value of one decision may depend on another decision:

$$\boxed{x_i x_j}$$

From a Decision Problem to QUBO

Suppose that we want to select exactly two projects:

$$x_A + x_B + x_C + x_D + x_E = 2.$$

Instead of keeping this as an explicit constraint, we penalize its violation:

$$P(x) = \lambda \left(x_A + x_B + x_C + x_D + x_E - 2\right)^2 .$$

From a Decision Problem to QUBO

Suppose that we want to select exactly two projects:

$$x_A + x_B + x_C + x_D + x_E = 2.$$

Instead of keeping this as an explicit constraint, we penalize its violation:

$$P(x) = \lambda \left(x_A + x_B + x_C + x_D + x_E - 2\right)^2.$$

A single unconstrained objective

$$\begin{aligned}\min_{x \in \{0,1\}^5} H(x) = & -\Big(8x_A + 6x_B + 5x_C + 7x_D + 4x_E \\ & + 3x_A x_C - 4x_B x_D\Big) \\ & + \lambda \left(x_A + x_B + x_C + x_D + x_E - 2\right)^2.\end{aligned}$$

From a Decision Problem to QUBO

Suppose that we want to select exactly two projects:

$$x_A + x_B + x_C + x_D + x_E = 2.$$

Instead of keeping this as an explicit constraint, we penalize its violation:

$$P(x) = \lambda \left(x_A + x_B + x_C + x_D + x_E - 2\right)^2.$$

A single unconstrained objective

$$\begin{aligned}\min_{x \in \{0,1\}^5} H(x) = & -\Big(8x_A + 6x_B + 5x_C + 7x_D + 4x_E \\ & + 3x_A x_C - 4x_B x_D\Big) \\ & + \lambda \left(x_A + x_B + x_C + x_D + x_E - 2\right)^2.\end{aligned}$$

The complete objective is quadratic: $\min_{x \in \{0,1\}^n} x^\top Q x$

Quadratic Unconstrained Binary Optimization

RU DOL FO VO

QUBO

Given symmetric matrix $Q \in \mathbb{R}^{n \times n}$, QUBO (Quadratic Unconstrained Binary Optimization) problem is

$$\begin{aligned} \min \quad & z^T Q z \\ \text{subject to} \quad & z_i \in \{0, 1\}. \end{aligned}$$

In general, QUBO is an NP–Hard problem.

Why QUBO

Why QUBO

- If we can solve QUBO, we can solve any NP-complete problem.
- QUBO has wide range of apps. from finance & economics to machine learning.
- Many combinatorial optimization problems can be naturally formulated as QUBO: SAT, graph coloring, max clique, maximum cut.
- QUBO fits perfectly to quantum annealeres.

Constrained QBO vs. QUBO

RU
DOL
FO
VO

Contraints

We can penalise linear and quadratic constraints, and still obtain QUBO. This gives QUBO additional modelling power.

Ground truth

“What is truth?”

Pontius Pilate, John 18:38



Ground truth

“What is truth?”

Pontius Pilate, John 18:38



In optimization, truth (ground truth) is the global optimum.

Ground truth

"What is truth?"

Pontius Pilate, John 18:38



In optimization, truth (ground truth) is the global optimum.

An algorithm returns a solution with objective value

$$\boxed{f(x) = 920}.$$

But how good is this solution?

Ground truth

"What is truth?"

Pontius Pilate, John 18:38



In optimization, truth (ground truth) is the global optimum.

An algorithm returns a solution with objective value

$$\boxed{f(x) = 920}.$$

But how good is this solution?

Ground truth	Gap	Interpretation
$f^\star = 930$	1.1%	Excellent
$f^\star = 1150$	20.0%	Not so good
$f^\star = ?$	$?$	**We do not know**

The key message

A solution tells us **what we found**.

The global optimum tells us **how well we did**.

The key message

A solution tells us **what we found**.

The global optimum tells us **how well we did**.

Global optimum $\longleftrightarrow$ ground truth $\longleftrightarrow$ ground state

The central problem

Find global optimum of QUBO:

$$\begin{aligned} \min \quad & z^T Q z \\ \text{subject to} \quad & z_i \in \{0, 1\}. \end{aligned}$$

The Quantum (Boiling) Frog Principle



The Quantum (Boiling) Frog Principle

RU DOL FO VO

The boiling frog metaphor

If the environment changes only gradually, the frog continuously adapts to the changing conditions.



The Quantum (Boiling) Frog Principle



The boiling frog metaphor

If the environment changes only gradually, the frog continuously adapts to the changing conditions.

The quantum annealing analogy

QUBO $\rightarrow$ Ising model $\rightarrow$ problem Hamiltonian.

- Start with an initial simple QUBO for which your (quantum) system easily finds ground state of related Hamiltonian.

The Quantum (Boiling) Frog Principle



The boiling frog metaphor

If the environment changes only gradually, the frog continuously adapts to the changing conditions.

The quantum annealing analogy

QUBO $\rightarrow$ Ising model $\rightarrow$ problem Hamiltonian.

- Start with an initial simple QUBO for which your (quantum) system easily finds ground state of related Hamiltonian.
- Slowly transform initial QUBO to the actual QUBO such that your quantum system remains in the ground state.
- Read out the ground state.

The Quantum (Boiling) Frog Principle



The boiling frog metaphor

If the environment changes only gradually, the frog continuously adapts to the changing conditions.

The quantum annealing analogy

QUBO $\rightarrow$ Ising model $\rightarrow$ problem Hamiltonian.

- Start with an initial simple QUBO for which your (quantum) system easily finds ground state of related Hamiltonian.
- Slowly transform initial QUBO to the actual QUBO such that your quantum system remains in the ground state.
- Read out the ground state.

Theorem

If the change is sufficiently slow, the quantum system can remain close to the ground state (global optimum).

D-Wave QPU

- D-Wave quantum processors implement quantum annealing on specialized hardware using superconducting qubits.

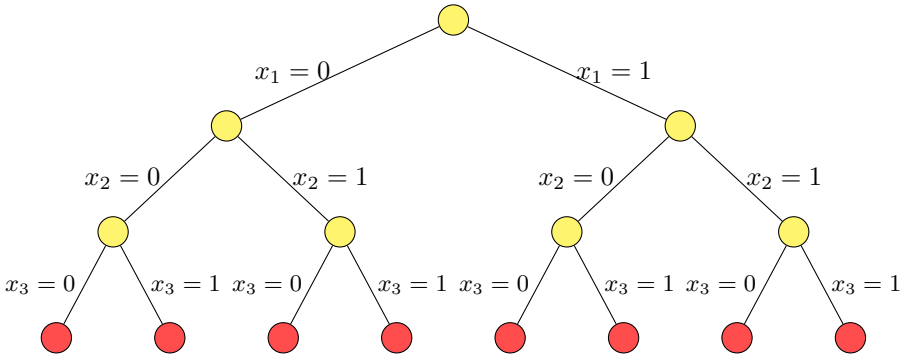
D-Wave QPU

- D-Wave quantum processors implement quantum annealing on specialized hardware using superconducting qubits.
- Qubits are connected according to specific graph topologies:
  - Chimera: D-Wave 2000Q, 2000+ qubits (degree 6)
  - Pegasus: Advantage, 5000+ qubits (degree 15)
  - Zephyr: Advantage2, 4400+ qubits (degree 20)

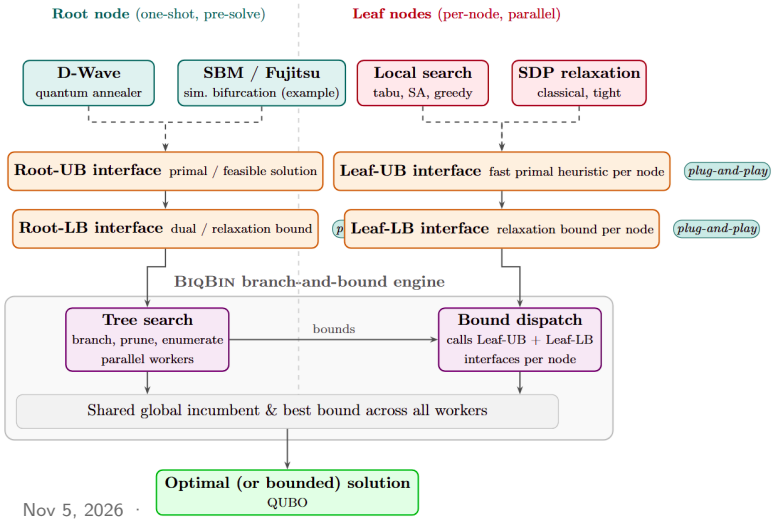
D-Wave QPU

- D-Wave quantum processors implement quantum annealing on specialized hardware using superconducting qubits.
- Qubits are connected according to specific graph topologies:
  - Chimera: D-Wave 2000Q, 2000+ qubits (degree 6)
  - Pegasus: Advantage, 5000+ qubits (degree 15)
  - Zephyr: Advantage2, 4400+ qubits (degree 20)
- Optimization problems are embedded onto the hardware graph, mapping logical variables to physical qubits.

Branch-and-Bound



QBIQ solver


Root node (one-shot, pre-solve)
Leaf nodes (per-node, parallel)
D-Wave
quantum annealer
SBM / Fujitsu
sim. bifurcation (example)
Local search
tabu, SA, greedy
SDP relaxation
classical, tight
Root-UB interface primal / feasible solution
Leaf-UB interface fast primal heuristic per node
plug-and-play
Root-LB interface dual / relaxation bound
Leaf-LB interface relaxation bound per node
plug-and-play
BiqBin branch-and-bound engine
Tree search
branch, prune, enumerate
parallel workers
bounds
Bound dispatch
calls Leaf-UB + Leaf-LB
interfaces per node
Shared global incumbent & best bound across all workers
Optimal (or bounded) solution
QUBO

Preliminary results on SIDON dataset

Table: Mean and median relative gap (%) against the best computed value, and exclusive wins, over 130 shared instances. QA = D-Wave quantum annealing; SA = simulated annealing; LHS = D-Wave Leap Hybrid Solver; BR = root solver.

Heuristic	Mean gap	Median gap	Excl. wins	Time (s, median)
QA	0.054	0.000	53	≈ 0.62 (flat)
LHS	0.066	0.014	6	3–6
SA	0.082	0.008	13	1–33
BR	0.079	0.004	37	0.3–34,000

Conclusions

- Quantum annealing can improve the search for the ground state, if used carefully.
- Studying the QPU topology is crucial for the efficient use of quantum annealers.
- Current quantum annealers operate serially, making parallelization challenging.
- How can digital quantum computers be integrated into Branch and Bound?

Thank You

1. Publications:
   - Vodeb, J., Eržen, V., Hrga, T., Povh, J. (2026). Accuracy and performance evaluation of quantum, classical and hybrid solvers for the Max-Cut problem: J. Vodeb et al. Quantum Information Processing, 25(7), 240.
   - Gusmeroli, N., Hrga, T., Lužar, B., Povh, J., Siebenhofer, M., Wiegele, A. (2022). BiqBin: a parallel branch-and-bound solver for binary quadratic problems with linear constraints. ACM Transactions on Mathematical Software (TOMS), 48(2), 1-31.

2. Funding:
   - Quantum solver for hard binary quadratic problems (QBIQ), L1-60136, Slovenian Research and Innovation Agency.
   - Quantum Excellence Center for Quantum Excellence Applications, G.A: 101194322, EuroHPC JU.