Multiple Query Optimization on the D-Wave 2X Adiabatic Quantum Computer

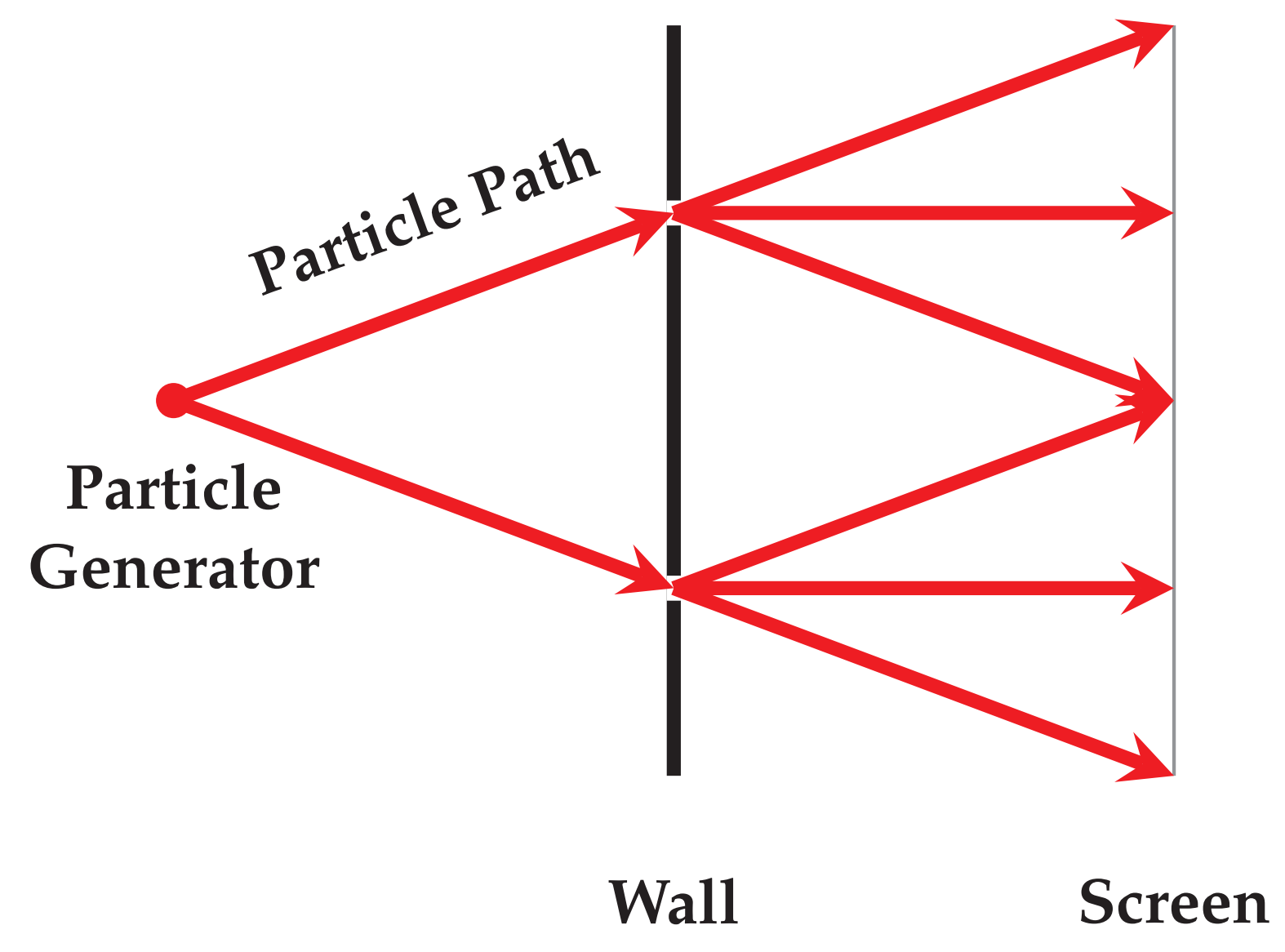
ecocloud
an EPFL research center

Immanuel Trummer, Christoph Koch
immanuel.trummer@epfl.ch, christoph.koch@epfl.ch

The associated paper appears on VLDB 2016.

Quantum Mechanics

Classical physics does not accurately describe physical processes at small scales. The figure below shows the double-slit experiment in which single particles are shot towards a wall with two slits. Particles passing through one of the slits will be registered on a screen. Classical physics predicts cumulation of particles in the middle of the screen as the middle can be reached through both slits.



Quantum mechanics explains why the middle of the screen remains empty: each particle enters both slits *at the same time* (this effect is called *superposition*). Multiple paths leading to the middle of the screen cancel each other out.

Quantum Computing

Quantum computers operate on qubits instead of bits. Qubits can adopt a superposition of states (one *and* zero) that is admissible only according to the laws of quantum mechanics. With a simplifying intuition, superposition allows quantum computers to explore multiple computational paths in parallel. This may lead to speedups compared to classical computers.

D-Wave Computer

The adiabatic quantum annealer by the Canadian company D-Wave exploits a limited form of quantum computing to solve NP-hard optimization problems. Its Niobium circuits are placed in a vacuum and cooled down to a temperature close to absolute zero. Under those conditions, Niobium becomes superconducting and quantum effects appear.

D-Wave Controversy

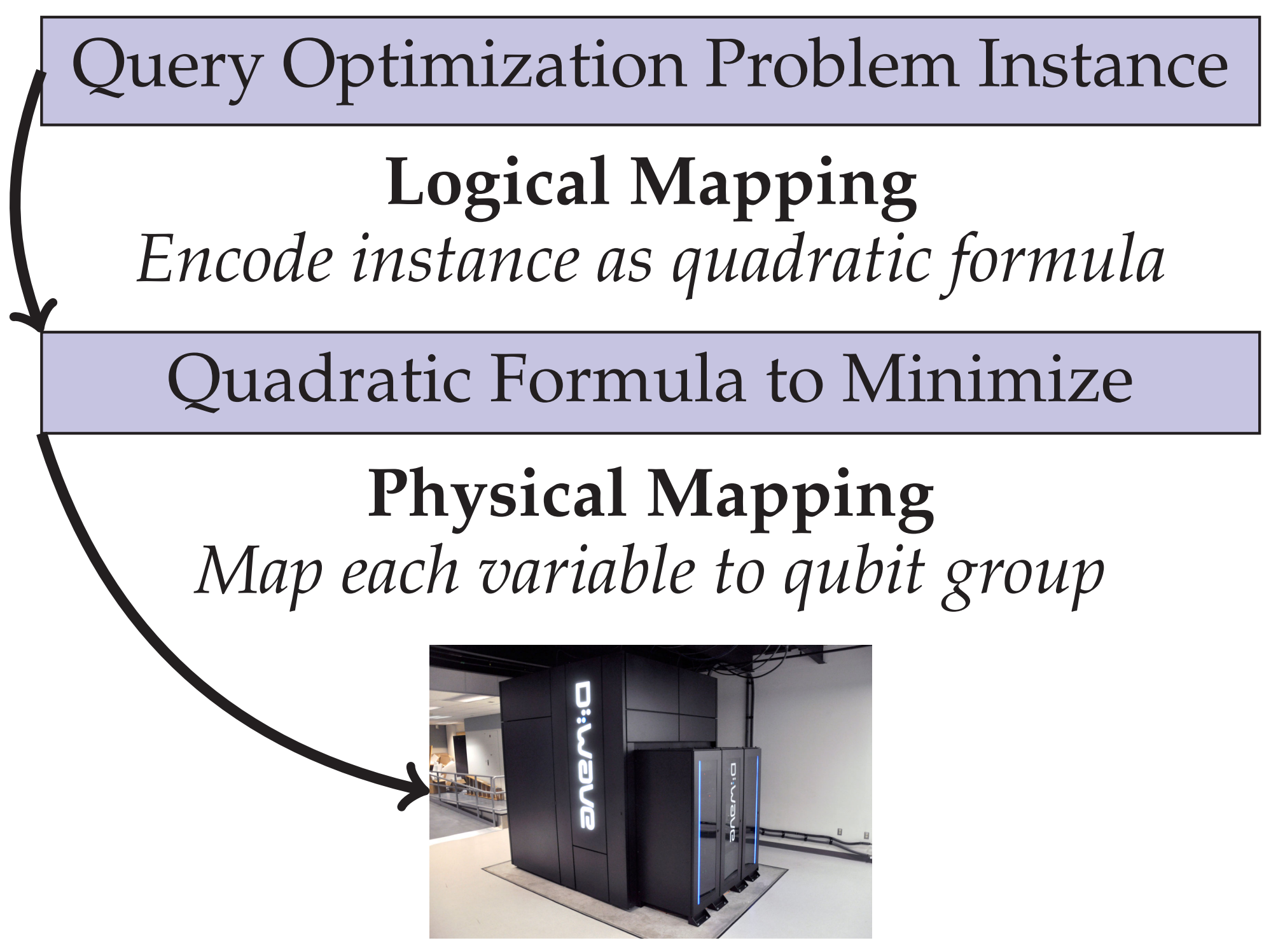
The D-Wave computer has been controversially discussed: do quantum effects have significant impact on its operation? Experimental evidence supporting a positive answer has become overwhelming. Scott Aaronson and other experts agree nowadays that *"this completely nails down the case for computationally relevant collective quantum tunneling in the D-Wave machine"*.

Project Focus

We show how to use the D-Wave computer to solve a classical NP-hard optimization problem from the database domain and evaluate its performance. We address the multiple query optimization problem: the goal is to select one execution plan for each query to minimize execution cost, taking into account cost savings by work overlap between different plans.

Approach Overview

The D-Wave computer minimizes quadratic formulas that depend on qubit values. We transform query optimization problem instances into that representation in two steps.



Query Optimization Problem Instance

Logical Mapping
Encode instance as quadratic formula

Quadratic Formula to Minimize

Physical Mapping
Map each variable to qubit group

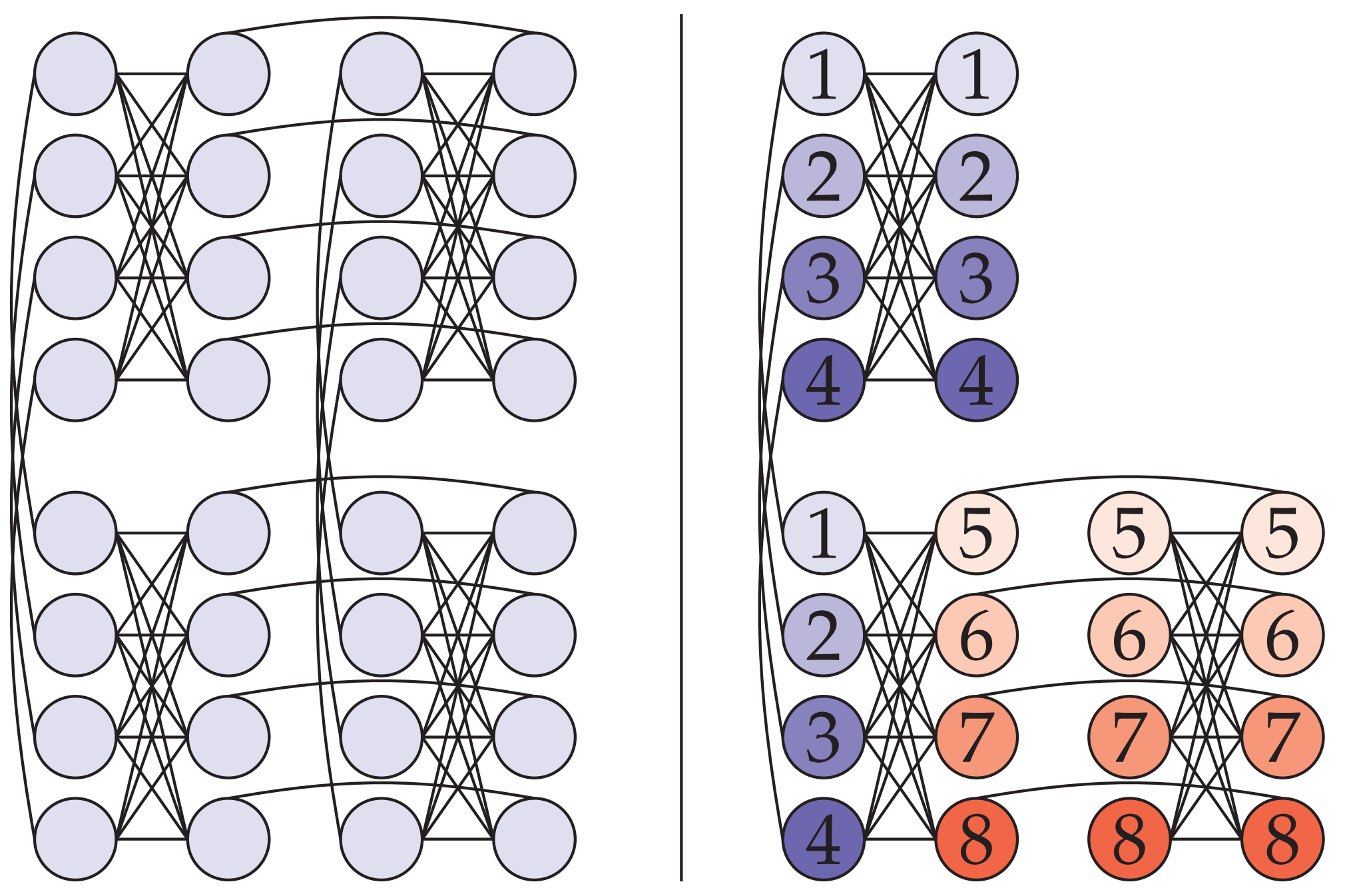
Logical Mapping

We translate a multiple query optimization problem instance into a quadratic formula that is minimized for an optimal solution. That formula has the following form

$$T_{cost} + T_{savings} + T_{select \geq 1} + T_{select \leq 1}$$

where T_{cost} represents plan execution costs, $T_{savings}$ the cost savings by work overlap between executed plans, $T_{select \geq 1}$ becomes minimal when selecting at least one plan per query, and $T_{select \leq 1}$ becomes minimal when selecting at most one plan per query. All four terms depend on binary variables representing whether specific plans are executed or not.

Physical Mapping



Qubits are sparsely connected in the D-Wave hardware (left side above shows connection pattern). Connected problem variables must be mapped to groups of qubits that share at least one hardware connection. The right side above shows how to map eight mutually connected problem variables to qubits: qubits with the same number (and color) represent the same problem variable. All qubits representing the same variable must be connected in a chain.

Complexity Analysis

The time complexity of the transformation into a quadratic formula is a low-order polynomial in the problem dimensions. There is currently no theoretical framework that allows us to analyze the time required by the D-Wave computer to minimize that formula. We can however analyze how the number of required qubits grows as a function of the problem dimensions.

After clustering queries such that queries in the same cluster can often share intermediate results, the asymptotic number of qubits is in

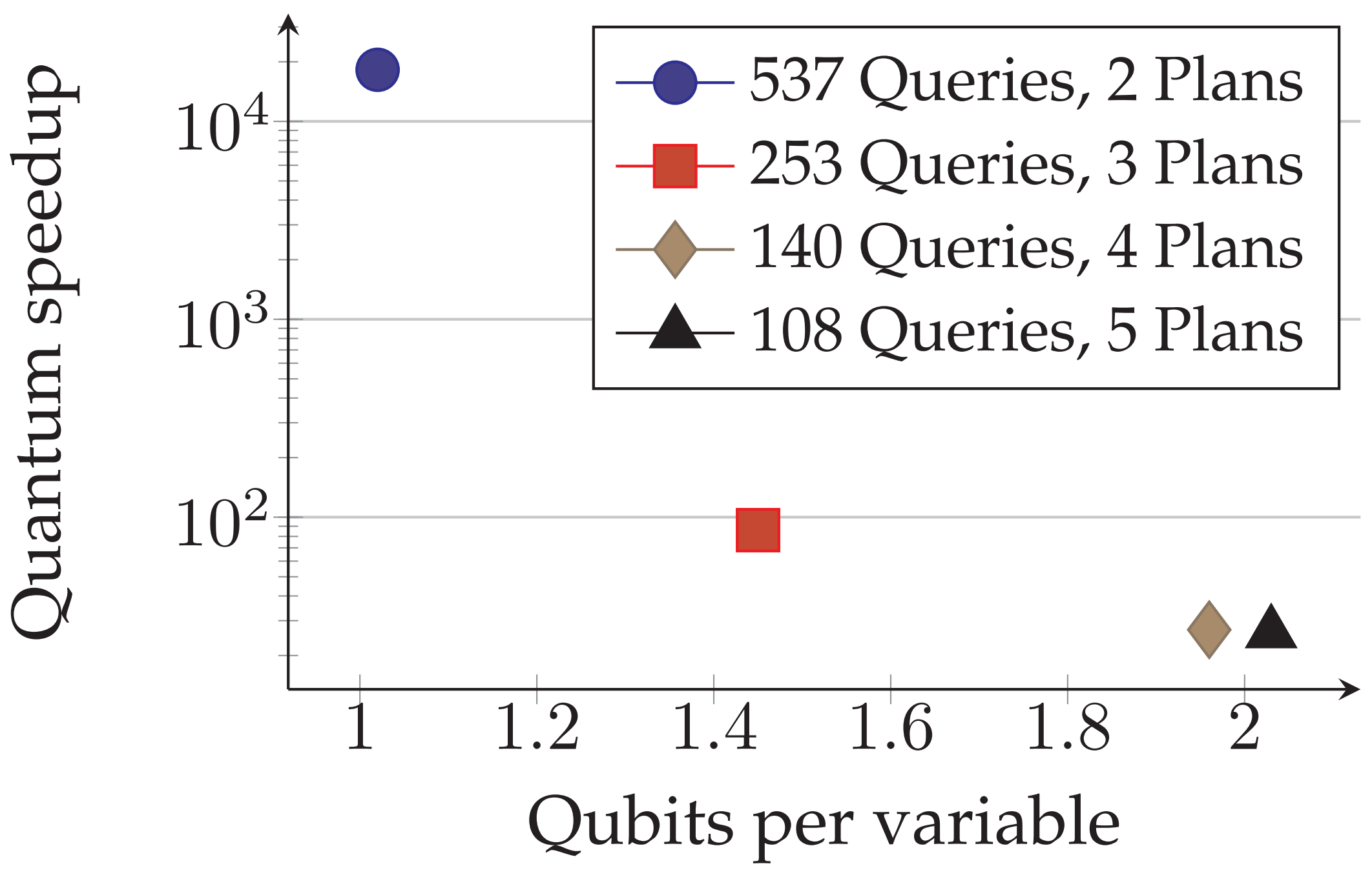
$$O(clusters \cdot (queries/cluster \cdot plans/query)^2)$$

Experimental Setup

We obtained access to a D-Wave 2X adiabatic quantum annealer with 1097 qubits, located at NASA Ames Research Center in California. We compare its performance (measured by optimization time) against multiple optimization algorithms on a commodity computer: a genetic algorithm, a greedy heuristic, a commercial linear programming solver, and a commercial quadratic programming solver.

Experimental Results

The following figure shows by how much the D-Wave computer is faster than the best classical algorithm at finding near-optimal solutions. We generate test cases in four classes that differ by the average number of qubits needed to represent one logical problem variable.



Significant speedups are possible but only if problem instances map well to qubits, i.e. if the number of qubits per variable is low.

Critical Discussion

We generate test cases that exploit nearly all available qubits. We measure pure optimization times but no pre-processing overheads. We only compare against classical algorithms that are commonly used for multiple query optimization. Therefore, the reported speedups correspond to upper bounds.

Acknowledgments

We acknowledge the support of the Universities Space Research Association Quantum AI Lab Research Opportunity Program.