INFINITY|Q⟩

Bridging the Gap in Combinatorial Optimization: A Quantum-Inspired Approach

Technology Brief | 28.09.24

Abstract

InfinityQ's TitanQ represents a breakthrough in solving NP-Hard combinatorial optimization problems. It surpasses traditional solvers and quantum computing limitations by leveraging quantum-inspired algorithms, enabling faster solutions for larger problems on classical hardware.

Context

Combinatorial optimization problems of NP-Hard complexity emerge across many domains. From scheduling of complex supply chains to optimization of financial portfolios, to discovery of new protein folding structures, these problems are ubiquitous in our daily lives. Many of these problems are or map to non-convex quadratic programming problems, which are NP-Hard, and generally necessitate a complex optimization solver to try to approach an optimal solution.

Traditionally, the resolution of non-convex problems has been tackled using variants of branch and bound, and brand and cut algorithms. This approach operates by systematically partitioning the solution space into increasingly smaller subsets, identifying the optimal local solution within each subset, and subsequently proposing the best solution found as a potential global solution.

Despite the guarantee that this method will ultimately find the true global solution given sufficient time, it is notably resource-demanding and struggles with scalability as the problem size increases. There are many solvers utilizing this algorithm: Gurobi, CPLEX, FICO XPRESS and others [3][4].

On the other hand, quantum-computing based approaches have been promising to eventually revolutionize this field for decades now. Unfortunately, the current quantum computing based methods are very limited in terms of the size and kinds of problems they can solve, while also being very expensive and thus not yet ready to answer real business needs.

The Ising Model, a quantum to classical link

The Ising model was first introduced by Wilhem Lenz and Ernst Ising in 1925 as a model to understand ferromagnetism in materials. It was introduced to understand dipolar interactions between atoms in a lattice, where each atom could have an up spin or a down spin.

While it started with ferromagnetism in classical systems, the quantum Ising model is used as a quantum mechanical model for magnetic spins in a lattice. These models are highly related, and used variously in modeling of statistical physics, quantum physics, materials science and many other fields.

Quantum Transverse Ising Model | Classical Ising Model

$$\boldsymbol{H(\sigma) = \sum J_{ij}\sigma_i^z\sigma_j^z + \sum h_i\sigma_i^z + g\sum \sigma_i^x} \rightleftharpoons \boldsymbol{E(s) = \sum J_{ij}s_is_j + \sum h_is_i}$$

Recently the model of computation around the Ising model has been used in quantum annealing, by companies such as D-Wave. Quantum annealing leverages the Ising model to solve complex optimization problems by evolving a quantum system toward its ground state, as demonstrated by D-Wave Systems' quantum processors [1]. Additionally, adiabatic quantum computing (AQC), which slowly transitions a system from an initial Hamiltonian to a final one often represented by an Ising Hamiltonian, utilizes the model to encode and solve computational problems [2].

Although quantum methods have been used to solve Ising model problems, the quantum system has not been proven to provide any major asymptotic performance gains beyond the quadratic gains created through Grover's algorithm and its derivatives. In most cases, it's not necessary to have a quantum computer to solve Ising model problems, and its not clear that there will ever be a major benefit to using a quantum computer to solve these problems.

Minimizing a function vs. Sampling from a distribution

Generally, the goal of optimization is to find the minimum (or maximum) configuration of a function. This is usually done using methods for local search, and only focusing on improving the function in small areas rather than looking at the overall system.

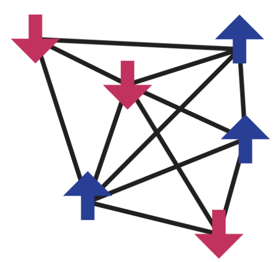
However, this leaves some understanding of the underlying structure of the system out. Instead, we will generally follow a probabilistic model where we can't exactly prove optimality but we can have guarantees of convergence through detailed balance conditions [3].

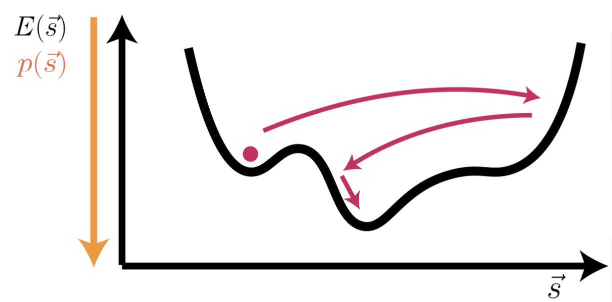
To start with, we will be looking at functions over binary variables to start simply. This same model works for integers and continuous variables as well when extended further. We can call the function we are trying to minimize the "energy" function, denoted by $E(s)$. A solution state is a string of binary values denoted by s. Put together we have the description described below:

$$argmin_s E(s) \;\; s \in \{0,1\}^N$$

To model this as a probability model, rather than a minimization function, we need to ensure that probabilities for all the states are positive (we can't have a negative probability) and the sum of probabilities adds up to 1 (probabilities need to be normalized). This gives us the probability model described below:

$$p(s) = \frac{1}{Z} e^{-E(s)}, \;\; Z = \sum_s e^{-E(s)}$$





$$E(\vec{s}) = \frac{1}{2} \sum_i \sum_j w_{ij} s_i s_j + b_i s_i$$

$$p(\vec{s}) \propto e^{-E(\vec{s})}$$

In this case we call Z the “partition function”, and it is the normalizing constant for the distribution. The partition function comes from thermodynamics, where it describes all the statistical properties of a system of non-interacting particles. The partition function ensures that the energies map to a normalized probability, such that $p(s) \in [0,1]$.

The discerning reader will also note that the probability function that we are demonstrating here is the Boltzmann distribution traditionally used to model distributions of states based on the temperature of the system and the energy of the state. To sample from this distribution, we can sample from the distribution using the same framework that is used for statistical physics simulations.

Markov Chain Monte Carlo (MCMC) Sampling

Markov Chain Monte Carlo (MCMC) is a parallelizable method for sampling from complex distributions. This includes distributions such as the Ising Model distribution listed above. The MCMC family of algorithms are very flexible, and allows to sample from continuous, discrete, and integer variables as well as complex distributions with constraints.

There are a few methods to sample using Markov Chain Monte Carlo (MCMC), including Gibbs sampling [4], the Metropolis Hastings algorithm[5], Langevin dynamics [6], and much more. At InfinityQ we use a variety of sampling methods to find high quality solutions to the underlying problem.

Extending the Ising Model

While the Ising Model problem listed above is a good starting point, it does not support many optimization constructs that are important for modeling of complex systems. A few missing features for most Ising Model and quantum solvers are listed here:

- Linear Constraints
- Quadratic Constraints
- Mixed variable types (binary, integer continuous)
- Higher order variables

While many other Ising model solvers (quantum and otherwise) are restricted by their underlying hardware, our probabilistic framework is flexible enough to provide extensions to solve the problems listed above. InfinityQ's methods and hardware are extensible enough to be able to solve these difficult problems which arise in real world problems.

		Ising	INFINITY|Q⟩ Beyond-Ising
Variable Types	Binary/Bipolar	✓	✓
	Integer		✓
	Continuous		✓
Constraints	Native Equality Support		✓
	Native Inequality Support		✓
	Native Set Partitioning Support		✓
	Readiness to market		✓

One of the largest solvers for complex problems

The TitanQ system boasts one of the largest Ising Machines available, capable of handling more than 100,000 fully connected variables in a quadratic problem. This is much larger than traditional solvers can handle and allows the InfinityQ system to solve problems out of reach of traditional methods.

When we benchmarked the TitanQ system on 10,000 variable MaxCut problems, we found impressive performance differences. The TitanQ system was able to solve problems 800x faster, while also producing a 1.5% better solution quality when compared to the Gurobi v10 running on the same system (TitanQ used a GPU, while Gurobi used the CPU). When this problem was scaled to larger sizes, we found that Gurobi was no longer able to provide good solutions and would crash and run out of memory.

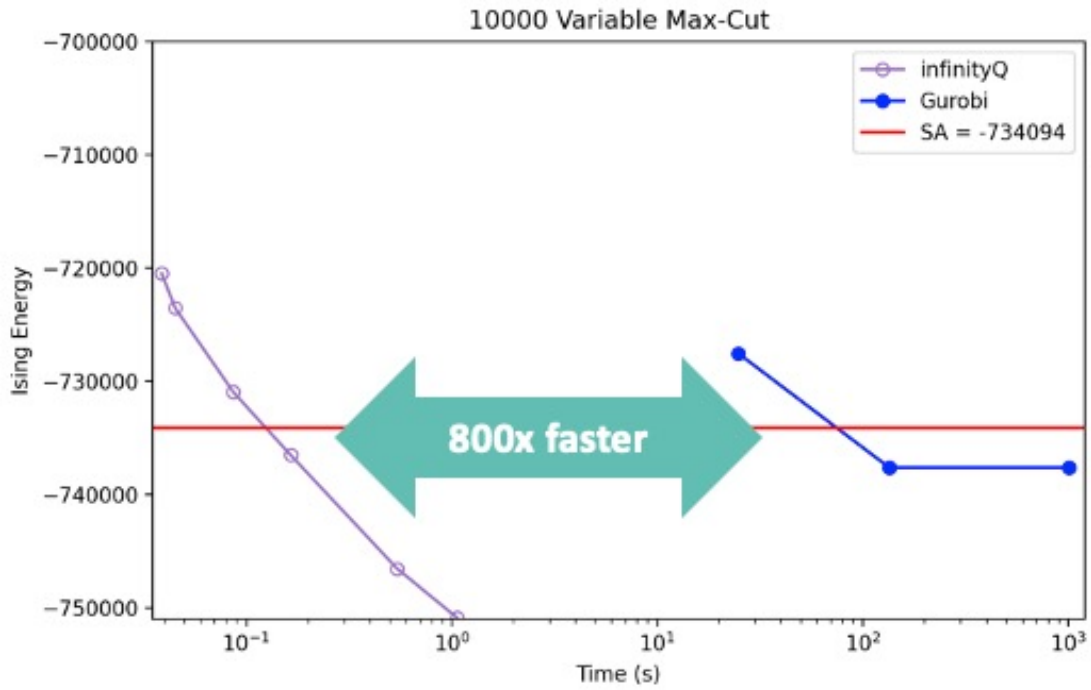


Figure 1: Performance on 10,000 variable MaxCut problem against the Gurobi solver. Gurobi uses Branch & Bound and Branch & Cut methods, which severely limits its ability to solve large quadratic problems.

One of the largest solvers for complex problems

We also compared the performance of the InfinityQ system against other quantum & quantum inspired systems. On a 100,000 variable challenge problem, the TitanQ system was able to produce solutions of the same or higher quality compared to Toshiba's Simulated Bifurcation Machine. Notably TitanQ only used a single GPU for this problem, while Toshiba used a cluster of 8 to solve the same problem, demonstrating an increase in efficiency for the TitanQ system.

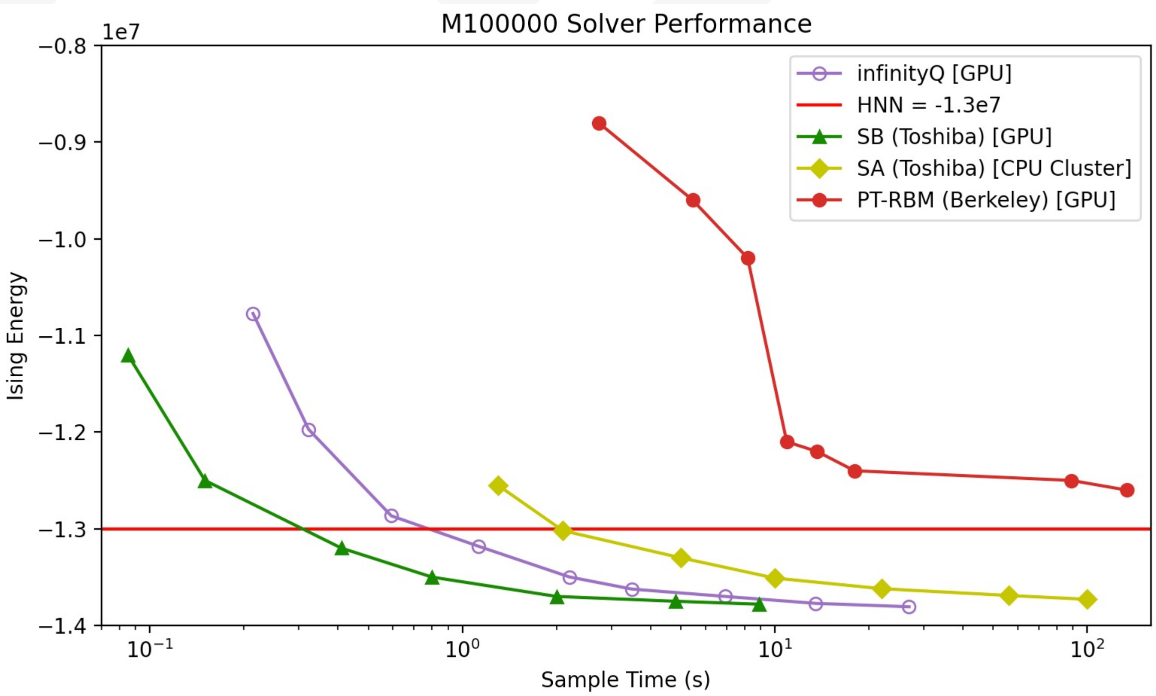


Figure 2: InfinityQ system performance on 100,000 variable dense MaxCut problem. Note InfinityQ uses 1 GPU for this system, while Toshiba uses a cluster of 8 GPUs.

Conclusion

InfinityQ uses a variety of probabilistic and sampling-based techniques to solve optimization problems in a method that is different from traditional optimization routines. We use methods that have been developed to solve physical & quantum simulations and reapply them to solve optimization problems. These methods can give large improvements on particularly hard problems (non-linear, NP-Hard) where traditional optimization routines may struggle.

References:

[1] Johnson, M. W., Amin, M. H. S., Gildert, S., Lanting, T., Hamze, F., Dickson, N., ... & Rose, G. (2011). Quantum annealing with manufactured spins. Nature, 473(7346), 194-198.

[2] Farhi, E., Goldstone, J., Gutmann, S., & Sipser, M. (2001). Quantum computation by adiabatic evolution. *arXiv preprint quant-ph/0001106*.

[3] https://personal.math.ubc.ca/~holmescerfon/teaching/asa22/handout-Lecture3_2022.pdf

[4] Casella, George, and Edward I. George. “Explaining the Gibbs sampler.” *The American Statistician* 46.3 (1992): 167-174

[5] Chib, Siddhartha, and Edward Greenberg. “Understanding the metropolis-hastings algorithm.” *The american statistician* 49.4 (1995): 327-335

[6] Girolami, Mark, and Ben Calderhead. “Riemann manifold langevin and hamiltonian monte carlo methods.” *Journal of the Royal Statistical Society Series B: Statistical Methodology* 73.2 (2011): 123-214.