TenSolver.jl

A tensor network based QUBO solver with JuMP integration

Iago Leal de Freitas

joint work with

David Bernal Neira

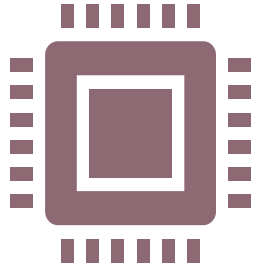
João Victor Paim de Cerqueira Melo Souza

JuMP-dev 2026

Edinburgh, UK

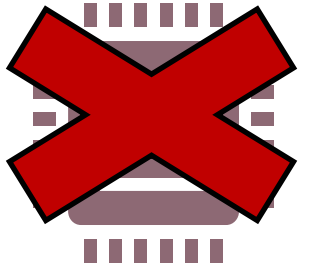
June 1st 2026

Motivation



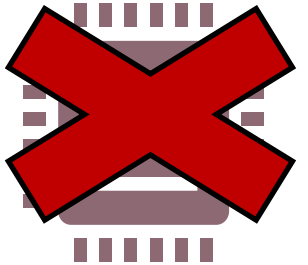
MIP Scalability

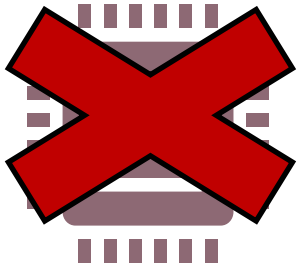
- MIP is **NP-hard**
- Classical algorithms are **CPU bound**
- Hard to **parallelize**



Want

- Parallel solver
- Take advantage of novel hardware (GPU, TPU, QPU)





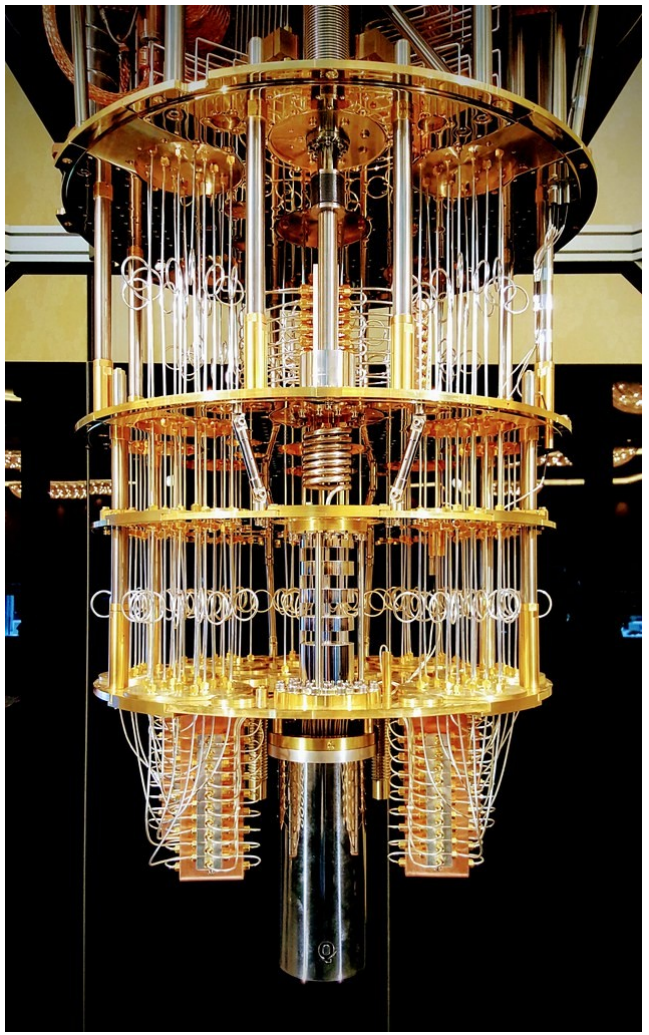
Idea

- Focus on **specific** family of optimization problems

Novel Hardware – Quantum Computing

Concretely:

- **Quantum Annealer**
  - Solves a **specific problem** (QUBO)
- Requires preprocessing
- Results are **probability distributions** over actual solutions

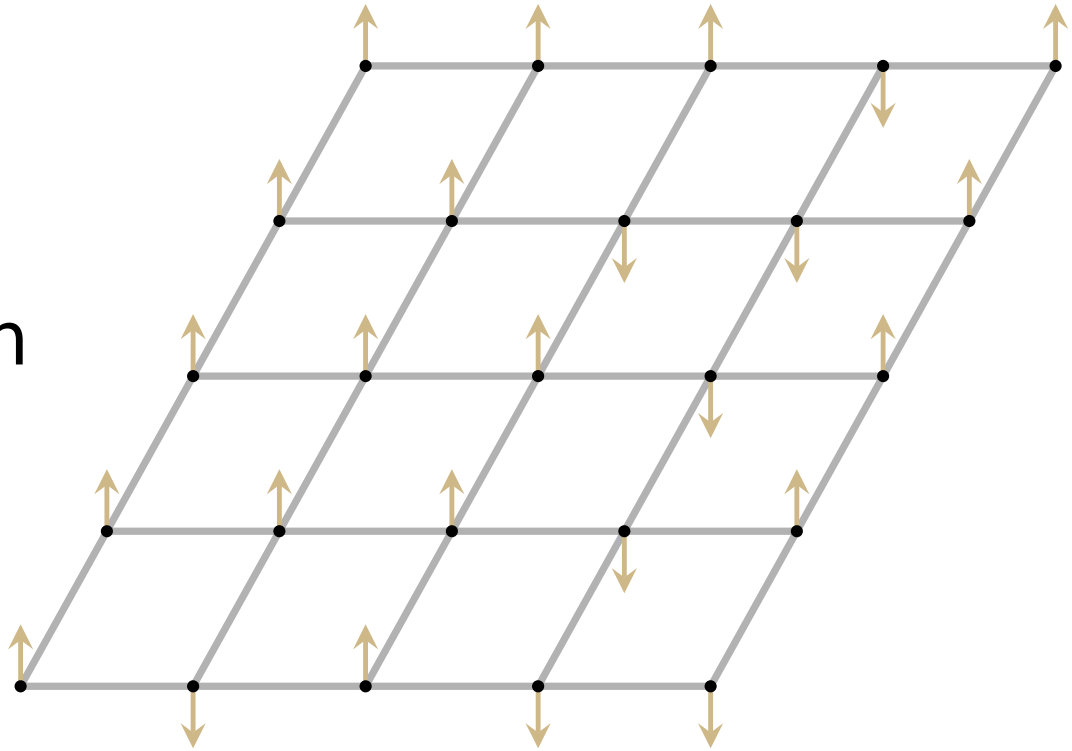


What is a QUBO?

- **Quadratic Unconstrained Binary Optimization**

$$\begin{aligned} \min_x \quad & x^\top Q x \\ \text{s.t.} \quad & x \in \{0,1\}^N \end{aligned}$$

- Can represent any discrete problem and approximate any mixed one
- Equivalent to the **Ising model**
- Great for **Quantum Computers**



1. Thermalisation and Relaxation of Quantum Systems - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Schematic-representation-of-a-configuration-of-the-2D-Ising-model-on-a-square-lattice_fig2_321920877 [accessed 13 Sept 2024]

Example conversion: ILP to QUBO

ILP

$$\min c\mathbf{x}$$
$$A\mathbf{x} = b$$
$$\mathbf{x} \in \{0,1\}^n$$

Quadratic Penalization

$$\min_{\mathbf{x}} c\mathbf{x} + \rho(A\mathbf{x} - b)^T(A\mathbf{x} - b)$$
$$\mathbf{x} \in \{0,1\}^n$$

MILP formulation

ILP

$$\min c\mathbf{x}$$
$$A\mathbf{x} = b$$
$$\mathbf{x} \in \{0,1\}^n$$

for example

$$\min_{\mathbf{x}} 2x_0 + 4x_1 + 4x_2 + 4x_3 + 4x_4 + 4x_5 + 5x_6 + 4x_7 + 5x_8 + 6x_9 + 5x_{10}$$
$$s.t. \begin{bmatrix} 1 & 0 & 0 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 1 \\ 0 & 1 & 0 & 1 & 0 & 1 & 1 & 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 & 1 & 1 & 1 \end{bmatrix} \mathbf{x} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$
$$\mathbf{x} \in \{0,1\}^{11}$$

QUBO formulation

QUBO

$$\min_{\mathbf{x}} \; \mathbf{x}^T Q \mathbf{x} + c \qquad \mathbf{x} \in \{0,1\}^n$$

same solution

$$= \min_{\mathbf{x} \in \{0,1\}^{11}} \mathbf{x}^T \begin{bmatrix}
-46. & 0. & 0. & 48. & 48. & 48. & 0. & 48. & 48. & 48. & 48. \\
0. & -44. & 0. & 48. & 0. & 48. & 48. & 0. & 48. & 48. & 48. \\
0. & 0. & -44. & 0. & 48. & 0. & 48. & 48. & 48. & 48. & 48. \\
48. & 48. & 0. & -92. & 48. & 96. & 48. & 48. & 96. & 96. & 96. \\
48. & 0. & 48. & 48. & -92. & 48. & 48. & 96. & 96. & 96. & 96. \\
48. & 48. & 0. & 96. & 48. & -92. & 48. & 48. & 96. & 96. & 96. \\
0. & 48. & 48. & 48. & 48. & 48. & -91. & 48. & 96. & 96. & 96. \\
48. & 0. & 48. & 48. & 96. & 48. & 48. & -92. & 96. & 96. & 96. \\
48. & 48. & 48. & 96. & 96. & 96. & 96. & 96. & -139. & 144. & 144. \\
48. & 48. & 48. & 96. & 96. & 96. & 96. & 96. & 144. & -138. & 144. \\
48. & 48. & 48. & 96. & 96. & 96. & 96. & 96. & 144. & 144. & -139.
\end{bmatrix} \mathbf{x} + 144$$

Results – Dwave Quantum Annealer

- Quantum computer
- **Minimizes** the energy of **Ising** spin system
- Exxonmobil + IBM efforts VRP data

Largest problem we could solve on a quantum computer!

Vehicle Routing Problem	D-Wave 2000Q
Number of variables	11
Fraction of optimal (feasible) solutions	0.24 (0.81)
Annealing time 1k samples [s]	**0.379**

1. https://www.dwavesys.com/tutorials/background-reading-series/introduction-d-wave-quantum-hardware

Real-world optimization

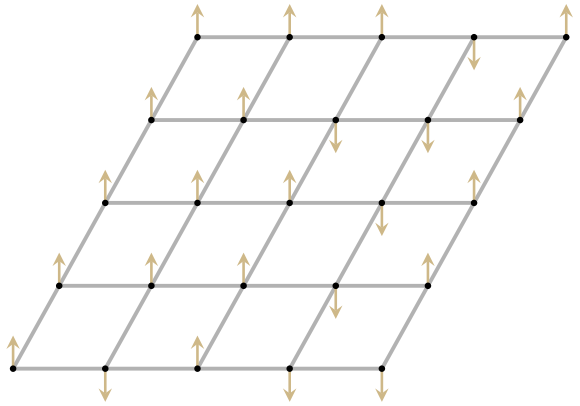
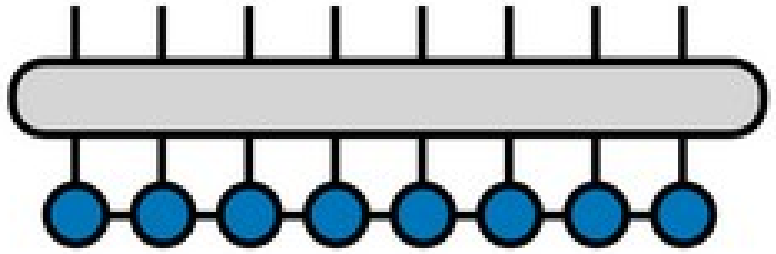
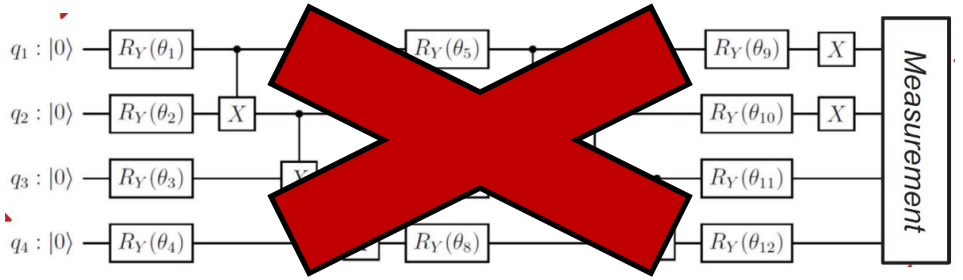
Problems

- Quantum optimization is a technology for **the future**
- They are not ready for current-scale models

Ideas!

- Use **GPUs** instead of QPUS
- Some quantum systems are easy to simulate
- Look for **good representations** for quantum states / probability distributions

From Classical to Quantum and Back Again

	Classical	Quantum
Model	$\min_x \quad c^\top x$ $\text{s.t.} \quad Ax = b$ $x \geq 0$ $x \in \mathbb{R}^n \times \{0,1\}^k$	
Method		

Quantum crash course

- A normalized quantum state is a 2-norm probability distribution over bitstrings
- Binary vectors becomes basis terms
- Tensor product is joint distribution

$$\{0,1\}^N \longrightarrow \bigotimes_{i=1}^{N} \mathbb{C}^2$$

Distributions

$$|\psi\rangle = \sum_{\omega \in \{0,1\}^N} \psi_\omega |\omega\rangle$$

“probabilities”

Born’s Rule

$$\frac{\langle\psi|A|\psi\rangle}{\langle\psi|\psi\rangle} = \mathbb{E}[A]$$

2-norm probability

$$|\psi_i|^2 = \text{prob}(i)$$

QUBO Hamiltonian

Remap a QUBO from a **discrete problem in N binary variables** into a **N-particle quantum system**

QUBO

$$\min_{x \in \mathbb{B}^N} \sum_{i \geq j} Q_{ij} x_i x_j \quad \text{s.t.}$$

$$Q \in \mathbb{R}^{N \times N}, \quad \mathbb{B} = \{0, 1\}$$

Hamiltonian

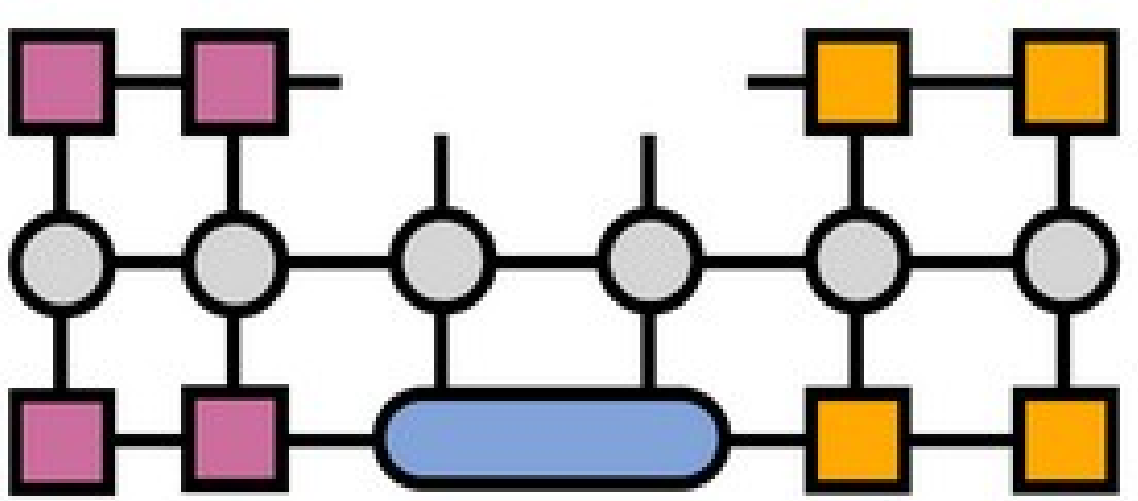
$$H\text{:} \quad \bigotimes_{i=1}^{N} \mathbb{C}^2 \rightarrow \bigotimes_{i=1}^{N} \mathbb{C}^2, \quad H = \sum\nolimits_{i \geq j} Q_{ij} D_i D_j$$

Least eigenvalue of H is the **optimal solution** to the QUBO

ISSUE: requires 2^N x 2^N coefficients

Tensor Networks

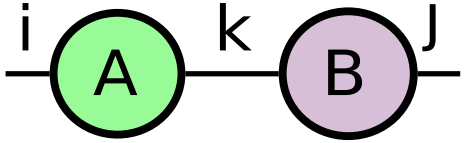
- **Classical** simulation of Quantum many-body systems
- Graphical language
- Amenable to **GPU**s
- Small representations for sparse tensors



- Standard tool for overturning quantum supremacy claims

1. https://www.simonsfoundation.org/2026/05/21/quantum-dynamics-breakthrough-overturns-claim-of-quantum-supremacy-opens-new-research-directions/

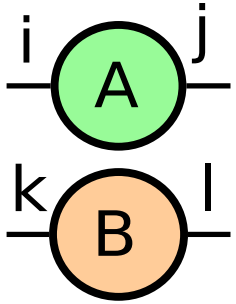
Graphical Language



$\Sigma A_i^k B_k^j$

Contraction

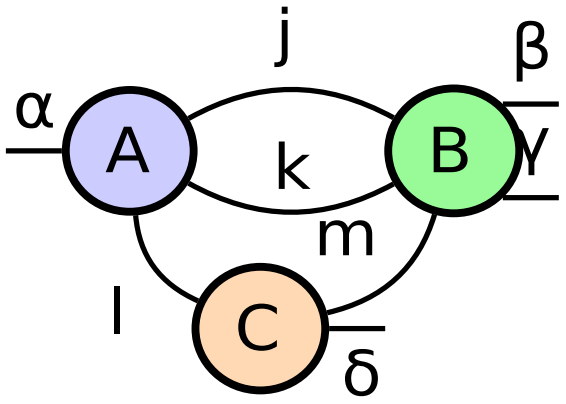
(Composition)



$A_i^j B_k^l$

Tensor

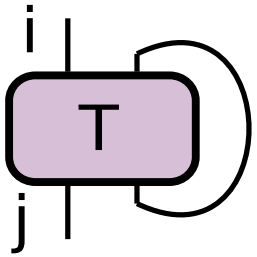
Product



$A_\alpha^{jkl} B_{jkm}^{\beta\gamma} C_l^{m\delta}$

Multi-Tensor

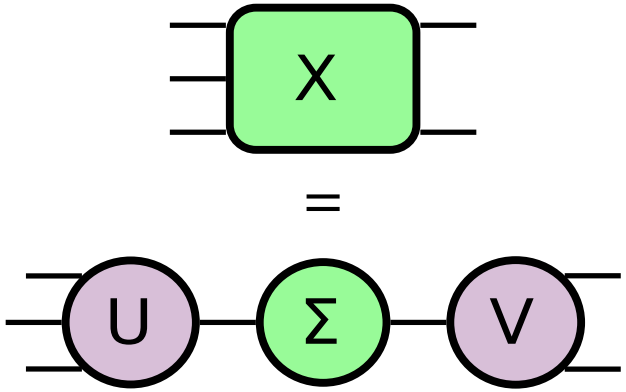
Contraction



T_{ik}^{jk}

Partial

Trace



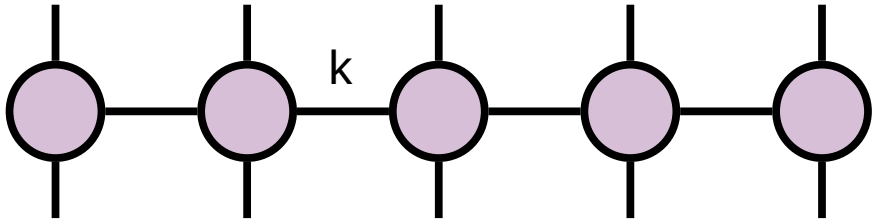
SVD

Decomposition

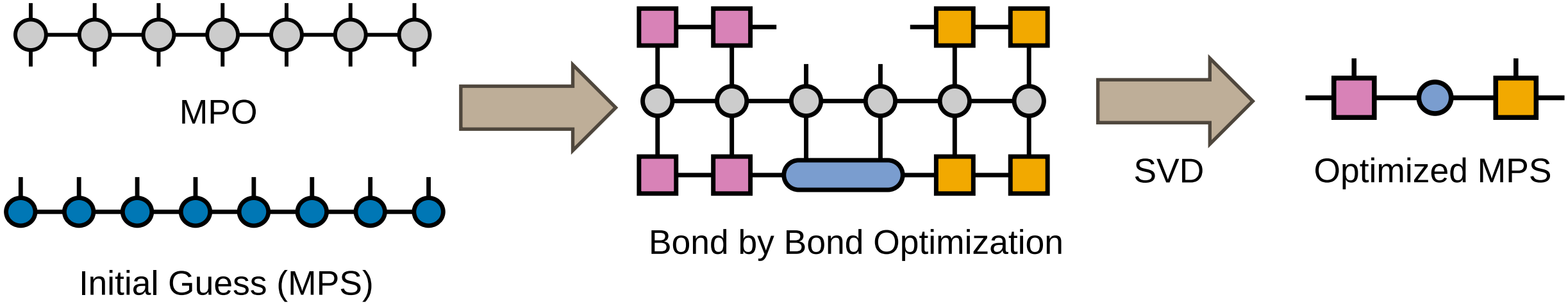
(Factorization)

Tensor Networks ~ Unevalued Tensor contractions

Matrix Product Operators



- Sequential compositions of rank 3 or 4-tensor
- **DMRG algorithm**: TN-specialized variational eigensolver (approximate solutions)
  - **$O(k^3N^4)$** computational complexity **$O(k^2N^2)$** memory complexity
- **Representation**: maximum $4N(N+1)^2$ coefficients required (better than 2^{2N})



Classical Simulation

$$\min_{x \in \mathbb{B}^N} \sum_{i \geq j} Q_{ij} x_i x_j$$

Algorithm Steps

$H \leftarrow \mathbf{MPO}(Q)$

$|\psi\rangle \leftarrow |\psi_0\rangle$

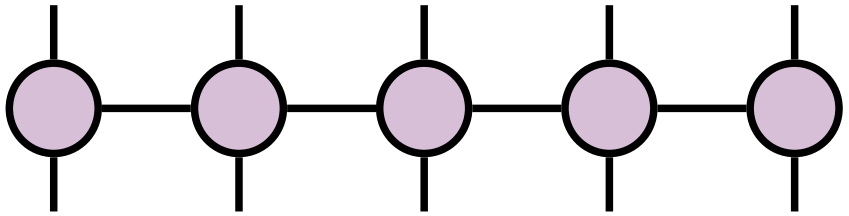
repeat

$E, |\psi\rangle \leftarrow \mathbf{DMRG}(\mathrm{H}, |\psi\rangle)$

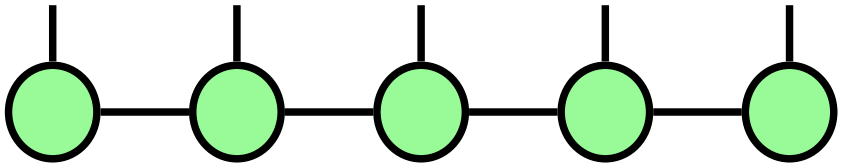
until $H|\psi\rangle \approx E|\psi\rangle$

$x^\star \leftarrow sample(|\psi\rangle)$

Represent



Eigensolve



Sample

$$[0,\ 1,\ 1,\ 0,\ 1]$$

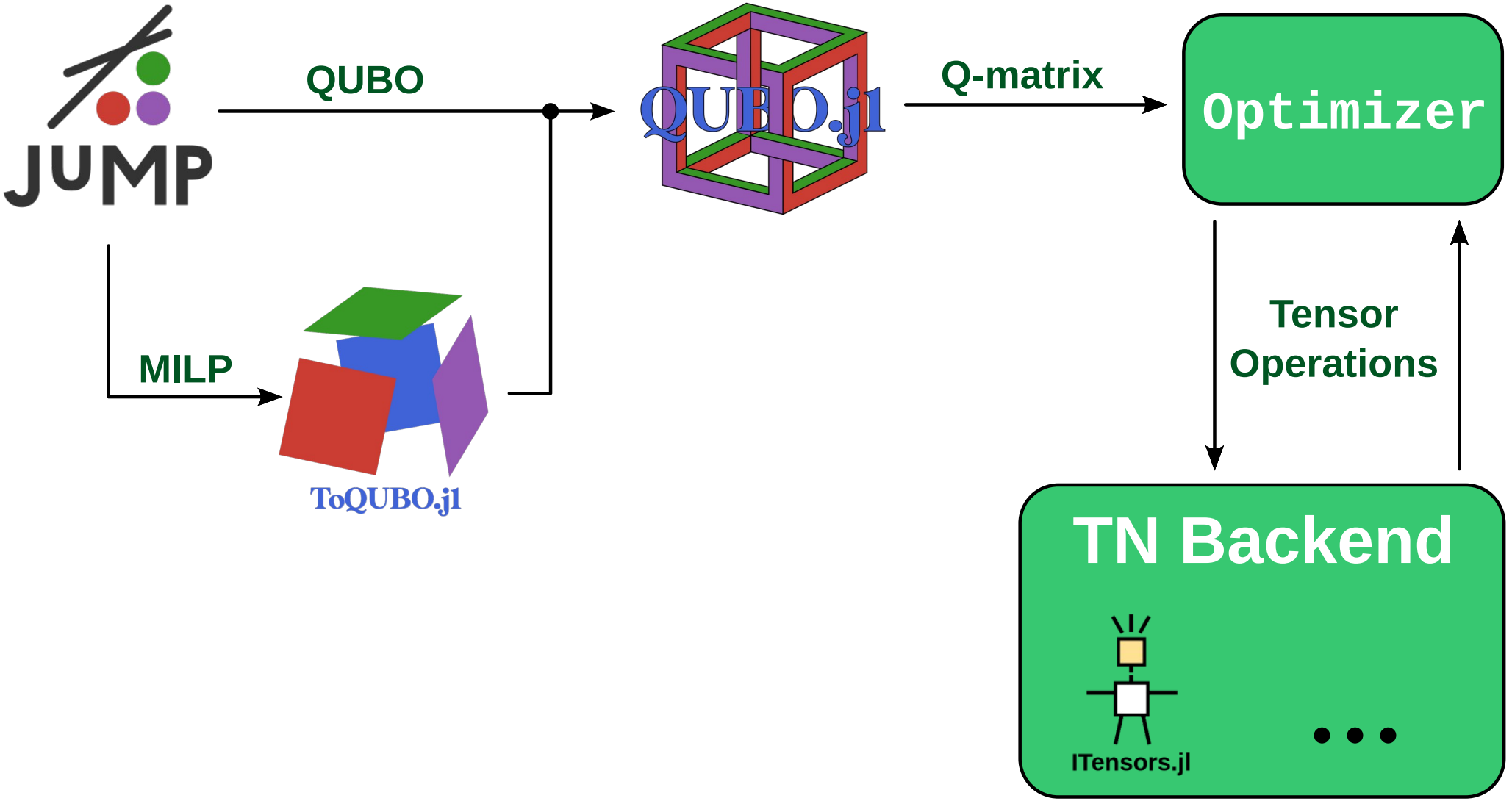
1. https://tensornetwork.org/diagrams/

TenSolver.jl

- Solves **QUBO** using **TN+DMRG**
- Runs on (multiple) **CPU** or **GPU**
- Tensor Network backend:
  - **ITensors.jl**
- **JuMP** allows abstracting all quantum away from the user

Visit the package!

Package Architecture



Code example

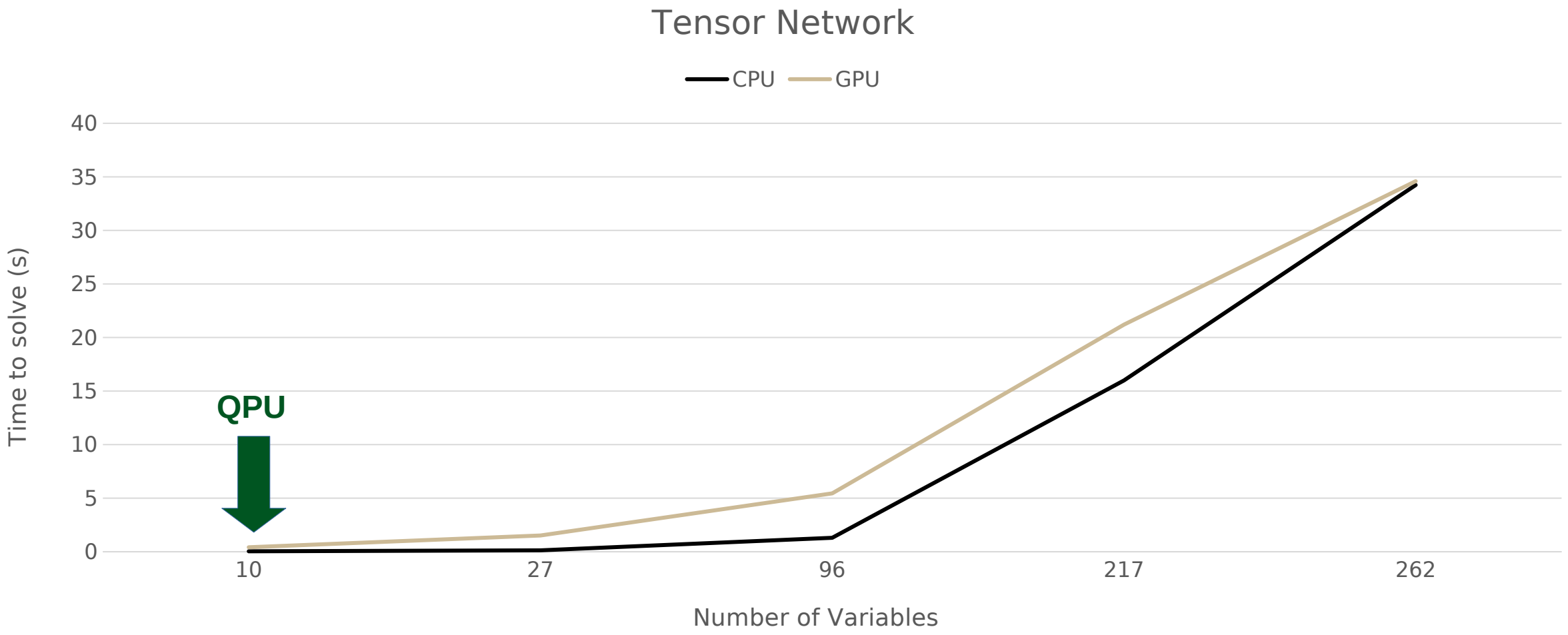
```
using JuMP, TenSolver
import CUDA, cuTENSOR

dim = 1000
Q   = ...     # Your matrix goes here

begin # Build QUBO and solve it using CUDA GPU
  m = Model(() -> TenSolver.Optimizer(device = CUDA.cu))

  @variable(m, x[1:dim], Bin)
  @objective(m, Min, x' * Q * x)

  optimize!(m)
end
```

Quantum Annealing vs Tensor Networks

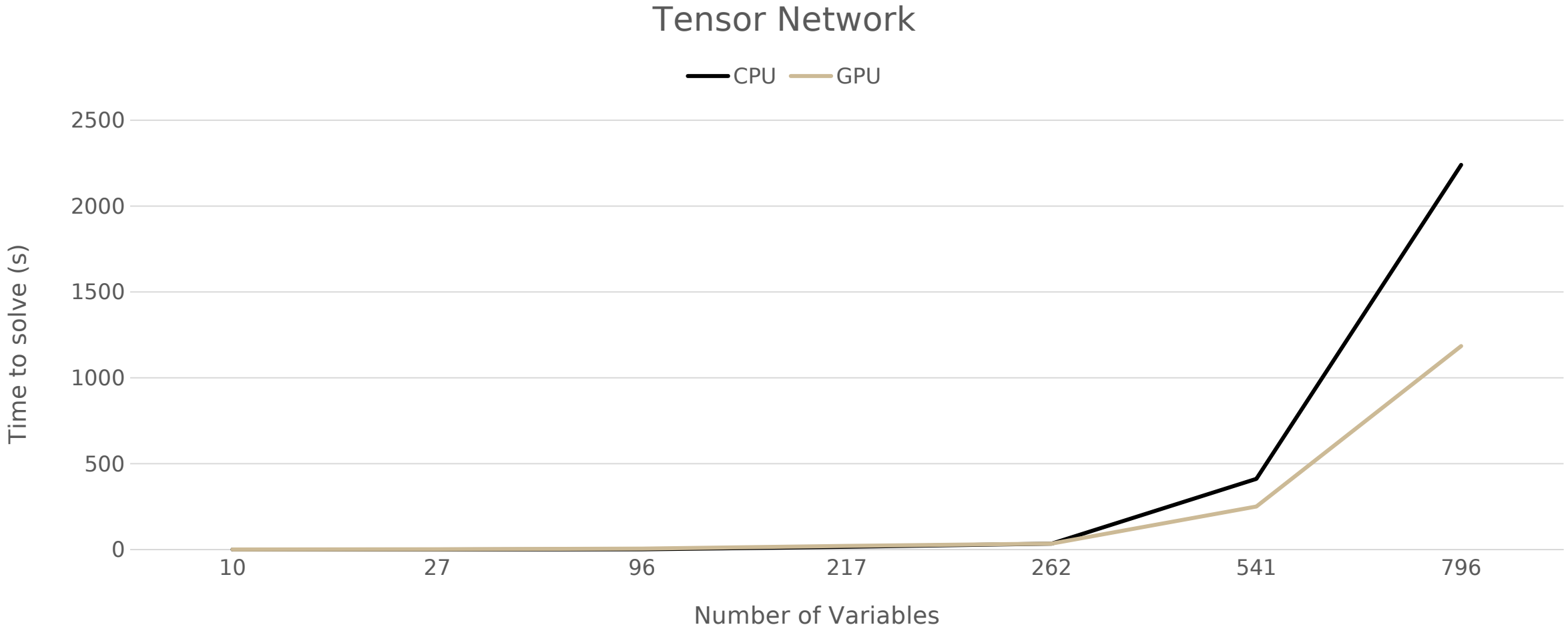
- Comparison for previous VRP problem **with 11 variables**

Vehicle Routing Problem	**Quantum**	**Classical (TenSolver.jl)**	
Hardware	**QPU**	**CPU**	**GPU**
	D-Wave 2000Q	AMD Epyc Milan	NVIDIA A100
Fraction of optimal (feasible) solutions	0.24 (0.81)	**1.00**	**1.00**
Time to solve [s]	0.379	**0.0484**	0.516

Results - Larger Problems



Results - Larger Problems

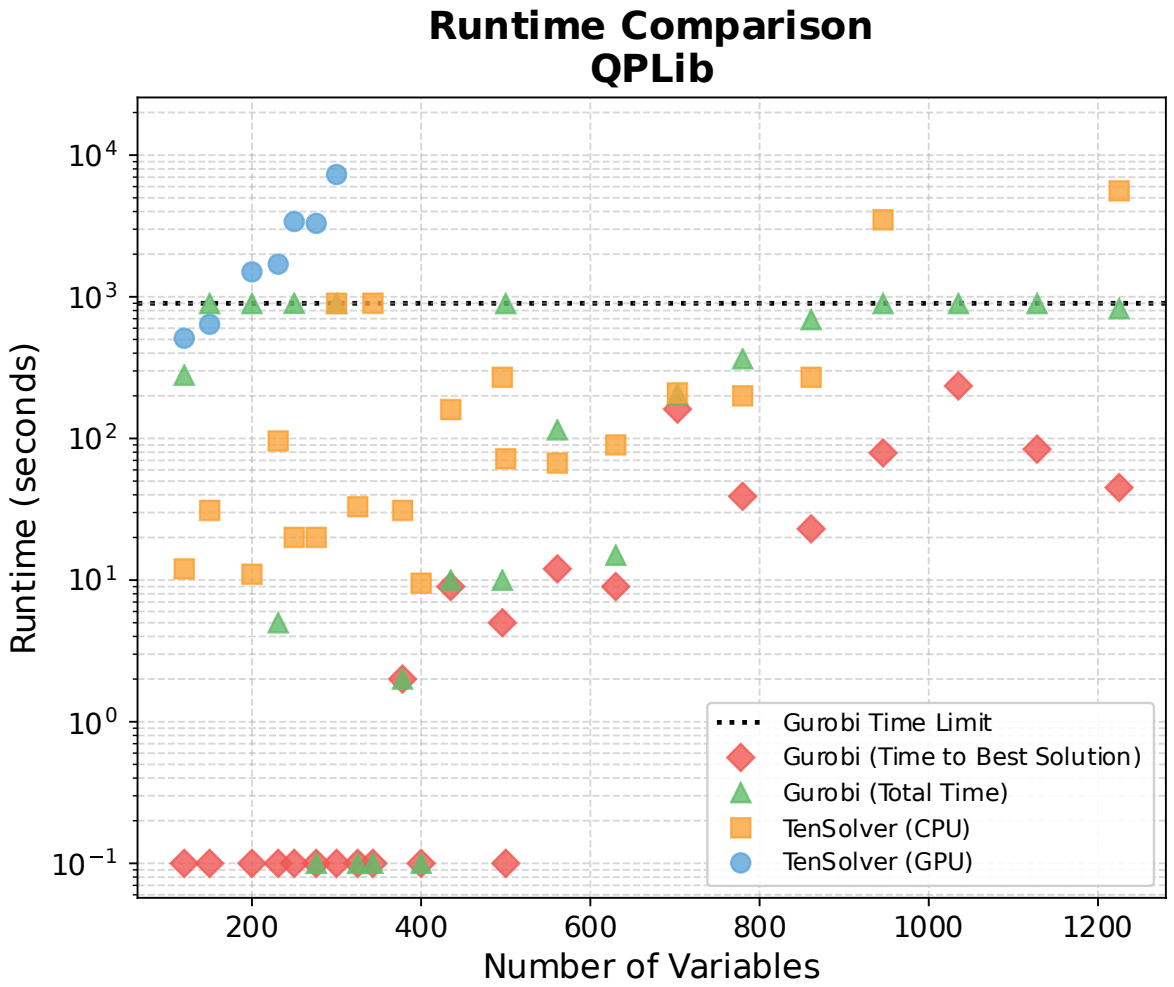


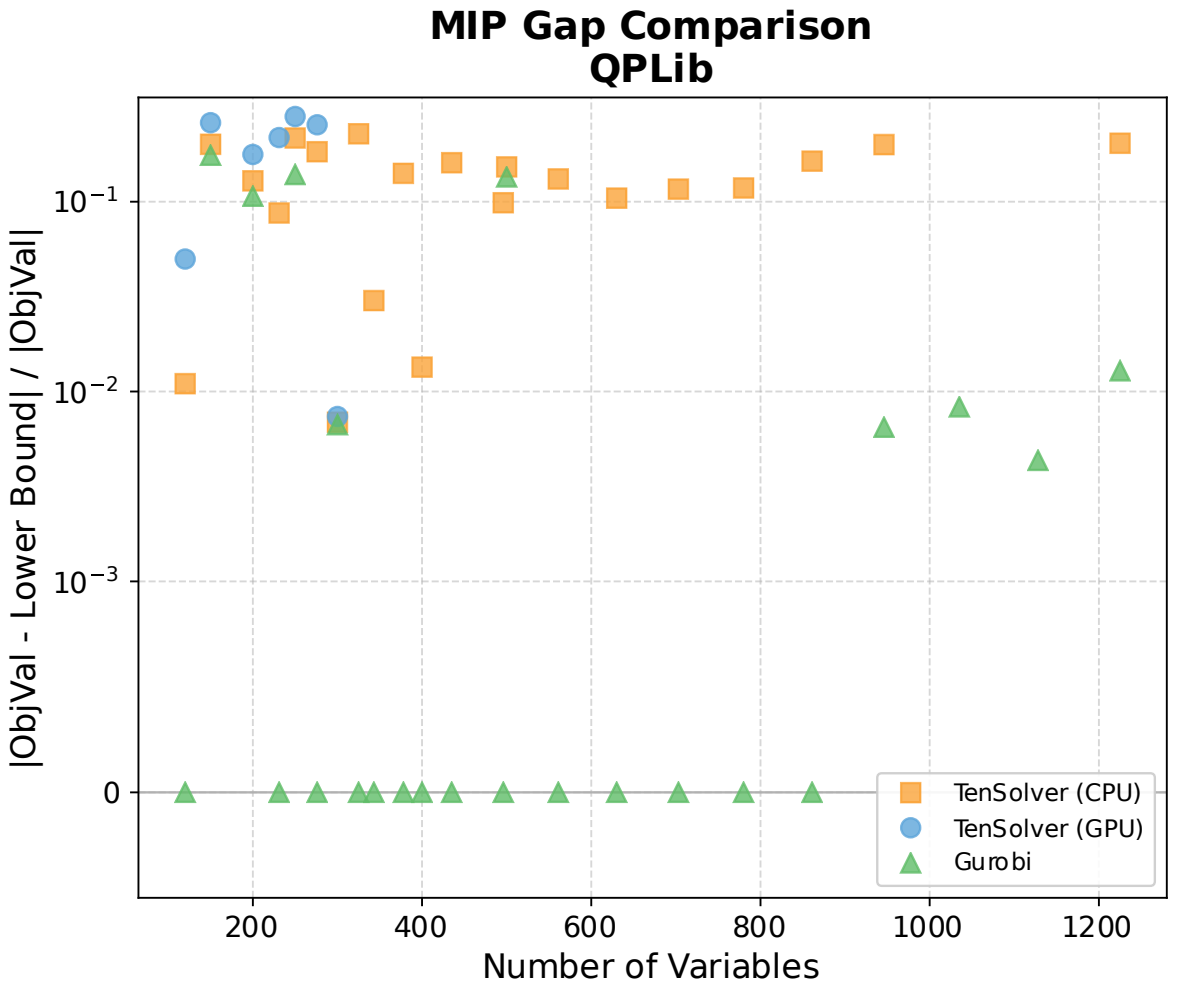
Further Studies

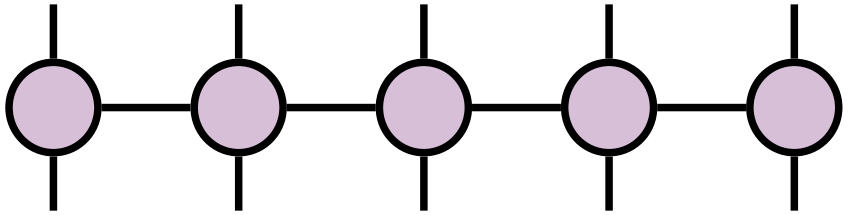
- **QPLib** dataset
- Problems ranging from **120** to **1225** variables
- **32 cores** CPUs

• **SVD Cut-off**	• **Max Bond Dimension**	• **Density Matrix Noise**	• **Convergence Threshold**	• **Initial State**
10^{-7}	• 40 (Initial)	• 10^{-5} (Initial)	• 10^{-4}	• Random MPS • max bond = 40

Results - QPLib







TenSolver.jl

Thank you!