QUANTUM ANNEALING AND TENSOR NETWORKS: A POWERFUL COMBINATION TO SOLVE OPTIMIZATION PROBLEMS

Author: Miquel Albertí Binimelis

Bachelor's final thesis

Under the Guidance of
Dr. Nahuel Norbeto Statuto
Dr. Luis Víctor Dieulefait

UNIVERSITAT DE BARCELONA

Facultat de Matemàtiques i Informàtica

Dept. of Mathematics and Computer Science, UB

Table of Contents

Project scope

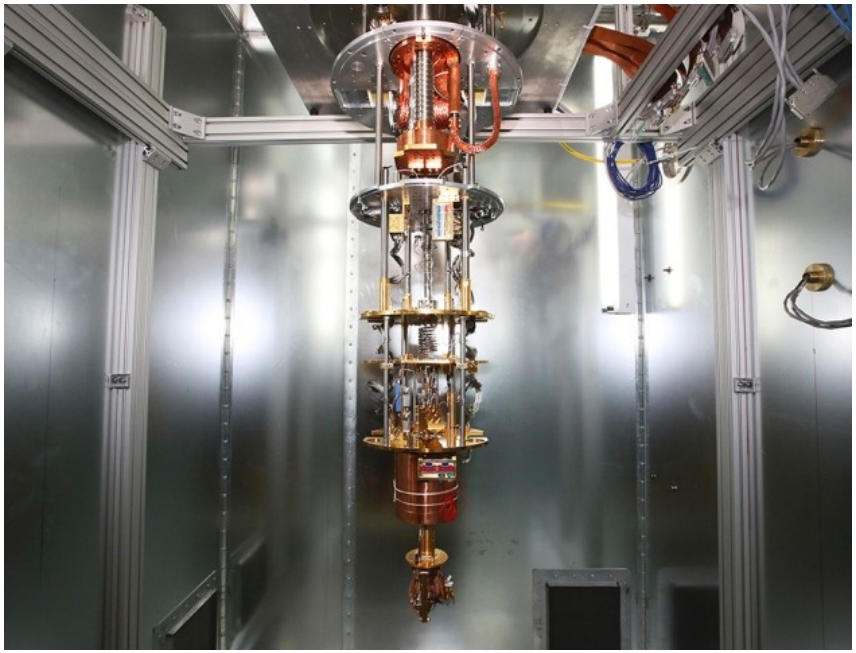


Figure: D-Wave's Quantum Annealer.

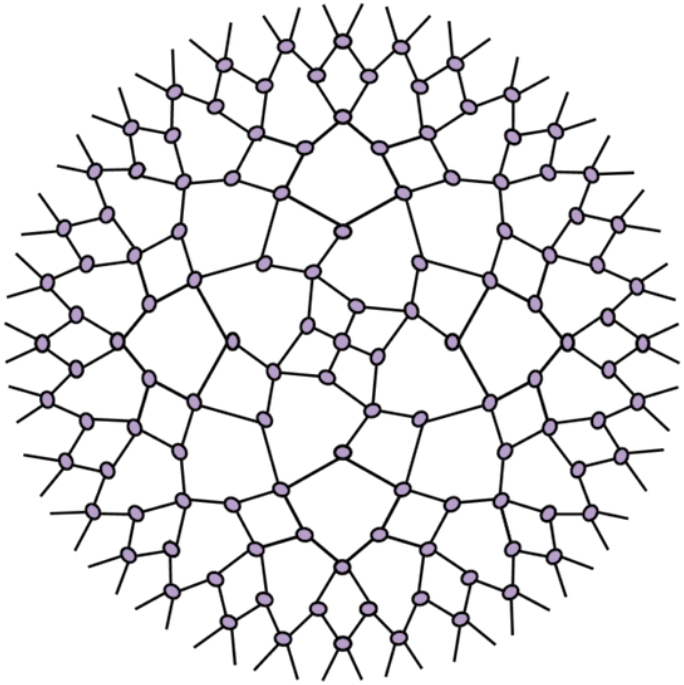


Figure: Tensor network.

Project scope

Adiabatic quantum computing (AQC)

- Algorithm: Quantum annealing (QA).

Tensor networks (TN)

- Algorithm: The Density matrix renormalization group (DMRG).
- Creation of the Matrix Product Operator (MPO) using finite automata.

Application: quadratic knapsack problem (QKP)

- Quadratic Unconstraint Binary Optimization (QUBO) formulation of the problem.
- Additional solvers: dynamic programming (DP), brute force.

Organization

	Summer	//	Feb H1	Feb H2	Mar H1	Mar H2	Apr H1	Apr H2	May H1	May H2	Jun H1
Study on AQC, QA, and QUBO		//									
Study on TN, MPS, and MPO		//									
Study on the DMRG		//									
Research on MPO and finite automata		//									
Research on the QKP and DP		//									
Problem formulation and implementation		//									
Run experiments		//									
Write the memory		//									

Table of Contents

Hilbert space

Ket notation: states

$$|0\rangle := \begin{pmatrix} 1 \\ 0 \end{pmatrix} \qquad |1\rangle := \begin{pmatrix} 0 \\ 1 \end{pmatrix}$$

$$|\psi\rangle = \alpha_0|0\rangle + \alpha_1|1\rangle = \begin{pmatrix} \alpha_0 \\ \alpha_1 \end{pmatrix} \; ; \; \alpha_0, \alpha_1 \in \mathbb{C}, \; |\alpha_0|^2 + |\alpha_1|^2 = 1$$

$$\mathbb{P}(|0\rangle) = |\alpha_0|^2 \qquad \mathbb{P}(|1\rangle) = |\alpha_1|^2$$

Hilbert space

System with N qubits

$$\{|i_1, \ldots, i_N\rangle := |i_1\rangle \otimes \ldots \otimes |i_N\rangle \mid i_1, \ldots i_N \in \{0, 1\}\}$$

$$|00\rangle := |0\rangle \otimes |0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \end{pmatrix} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

$$|01\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 0 \\ 0 \end{pmatrix}$$

...

Dimension: 2^N, where N is the numbe of particles

Bra notation: Dual vector

Complex conjugate of the elements in the transpose vector (row vector).

Ket: $|\psi\rangle = \sum_i^n \alpha_i |\psi_i\rangle$ Bra: $\langle\psi| = \sum_i^n \alpha_i^* \langle\psi_i|$

- Inner product: $\langle\psi|\phi\rangle$
- Norm: $\||\phi\rangle\| := \sqrt{\langle\phi|\phi\rangle}$

Operators: Hamiltonians

2nd postulate of quantum mechanics

The time evolution of the state of a closed quantum system is described by the time-depentent Schrödinger equation

$$i\hbar \frac{d|\phi(t)\rangle}{dt} = H(t)|\phi(t)\rangle$$

If H does NOT depend on time, we can solve the differential equation to obtain the time-independent Schrödinger equation

$$H|\psi_i\rangle = E_i|\psi_i\rangle \ ,$$

where H is a Hermitian operator.

Operators: Hamiltonians

if $|\phi\rangle$ is not an eigenstate

$$\begin{aligned} H|\phi\rangle &= H\left(\alpha_1|\psi_1\rangle + \ldots + \alpha_N|\psi_N\rangle\right) \\ &= \alpha_1 H|\psi_1\rangle + \ldots + \alpha_N H|\psi_N\rangle \\ &= \alpha_1 E_1|\psi_1\rangle + \ldots + \alpha_N E_N|\psi_N\rangle \ . \end{aligned}$$

Expected energy:

$$\begin{aligned} \langle\phi|H|\phi\rangle &= \left(\sum_i^n \alpha_i^* \langle\psi_i|\right)\left(\sum_i^n \alpha_i E_i|\psi_i\rangle\right) \\ &= \alpha_1^*\alpha_1 E_1\langle\psi_1|\psi_1\rangle + \ldots + \alpha_N^*\alpha_N E_N\langle\psi_N|\psi_N\rangle \quad \text{(orthogonality)} \\ &= |\alpha_1|^2 E_1 + \ldots + |\alpha_N|^2 E_N \ . \quad \text{(orthonormality)} \end{aligned}$$

Operators: Pauli matrices

The Pauli group:

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} \quad Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}$$

The Z-basis:

$$Z|0\rangle = |0\rangle \qquad Z|1\rangle = -|1\rangle$$

QUBO formulation

Quadratic Unconstrained Binary Optimization (QUBO)

Combinatorial optimization problem that can be formulated as

$$\min_x x^T Q x$$

where $x = (x_1, \ldots, x_N)^T \in \{0,1\}^N$ is a vector containing the N binary variables of the problem and $Q = \{q_{ij}\}_{i,j \in \{1,\ldots,N\}}$ is a constant square matrix that depends on the problem formulation.

- x $\rightarrow$ decision variables (binary).
- $x^T Q x$ $\rightarrow$ cost function.

Ising model

Through a change of variables

$$x_i = \frac{1 - s_i}{2}$$

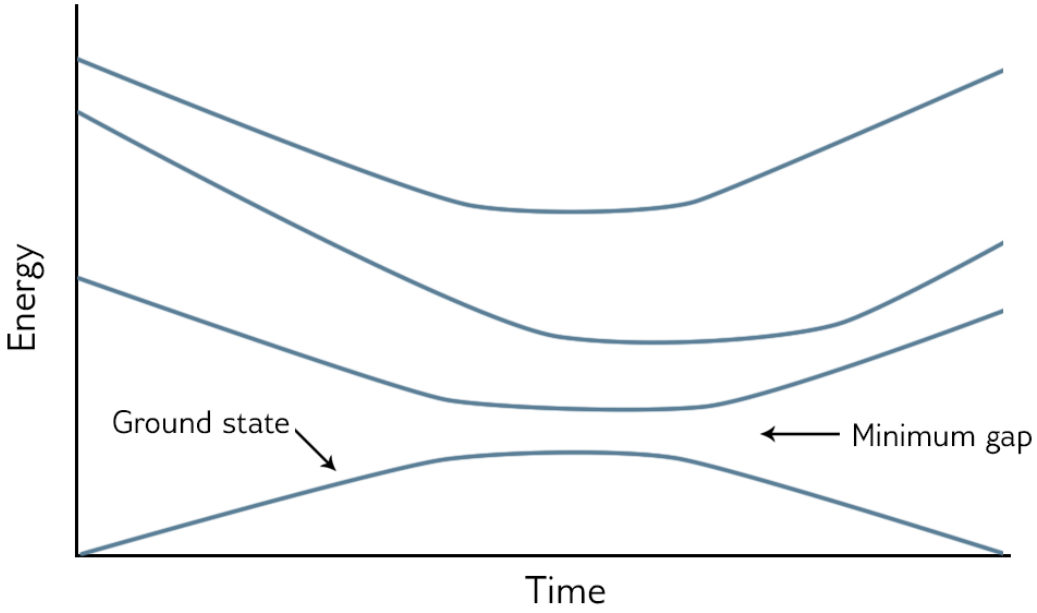
we obtain the Hamiltonian of an Ising model

$$H = -\sum_{i,j} J_{ij} Z_i Z_j - \sum_j h_i Z_i \ , \quad J_{ij}, h_i \in \mathbb{R}$$

where

$$Z_i := I \otimes \ldots \otimes Z_i \otimes \ldots \otimes I$$
$$Z_i Z_j := I \otimes \ldots \otimes Z_i \otimes \ldots \otimes Z_j \otimes \ldots \otimes I$$

Adiabatic quantum computing



$$g_{min} = \min_{s \in [0,1]} (E_1(s) - E_0(s))$$

Time-dependent Hamiltonian

$$H(t) = (1 - s(t))H_0 + s(t)H_p$$

$$t \in [0, T], \ T \in \mathbb{R}_{>0} \qquad s(0) = 0, \ s(T) = 1$$

Quantum annealing

$$H(t) = (1 - s(t))H_0 + s(t)H_p$$

$$t \in [0, T], \ T \in \mathbb{R}_{>0} \qquad s(0) = 0, \ s(T) = 1$$

$$H_0 = -\sum_{i=1}^{N} X_i$$

$$H_p = -\sum_{i,j} J_{ij} Z_i Z_j - \sum_{j} h_i Z_i, \quad J_{ij}, h_i \in \mathbb{R}$$

Annealing Hamiltonian

$$H(t) = (1 - s(t)) \left[-\sum_{i=1}^{N} X_i \right] + s(t) \left[-\sum_{i,j} J_{ij} Z_i Z_j - \sum_{j} h_i Z_i \right]$$

$$t \in [0, T], \ T \in \mathbb{R}_{>0} \qquad s(0) = 0, \ s(T) = 1$$

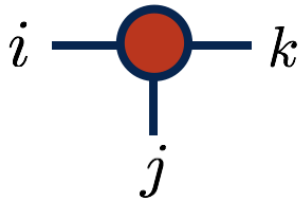
Table of Contents

Introduction to TN

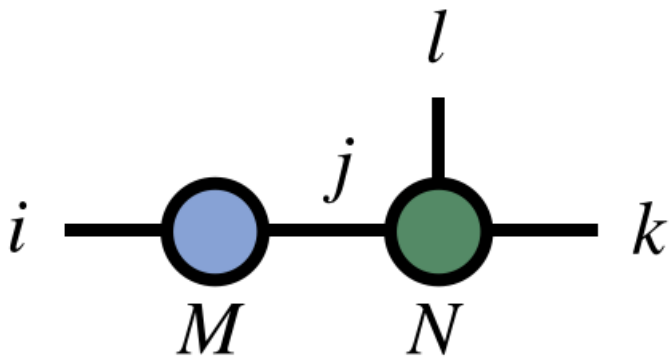
v_j

M_{ij}

T_{ijk}



Contractions



$$= \sum_j M_{ij} N_{jkl}$$

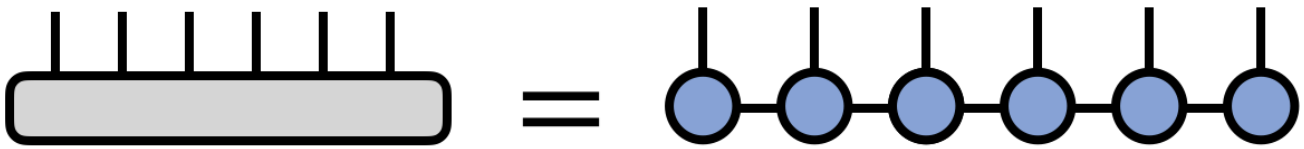
Cost: $O(d^4)$

Matrix Product States (MPS)

Each particle k can be in d_k possible states ($d_k = 2$ if it is a qubit)

$$|\psi\rangle = \sum_{i_1 i_2 \ldots i_N} C_{i_1 i_2 \ldots i_N} |i_1 \ldots i_N\rangle \ , \qquad i_k = 0, \ldots, d_k - 1, \quad k = 1, \ldots, N$$

has $d_1 d_2 \ldots d_N$ parameters.



$$|\psi\rangle = \sum_{i_1 i_2 \ldots i_N} M_{i_1} \ldots M_{i_N} |i_1 \ldots i_N\rangle \ , \qquad i_k = 0, \ldots, d_k - 1, \quad k = 1, \ldots, N$$

has $d_1 m_1 + m_1 d_2 m_2 + \ldots + m_{N-2} d_{N-2} m_{N-1} + m_{N-1} d_N$ parameters.

Conversion from a general tensor to its MPS form

$$C_{i_1,(i_2,\ldots,i_N)} = \sum_{a_1} U_{i_1,a_1} S_{a_1,a_1} V^{\dagger}_{a_1,(i_2\ldots i_N)} \ .$$

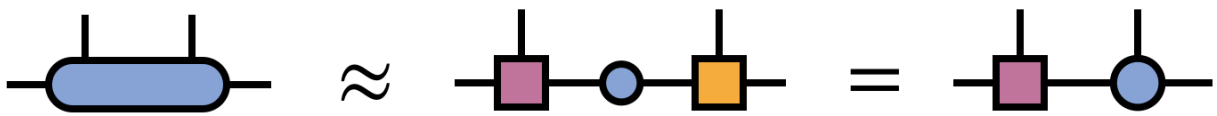
Then, we define

$$M_{i_1} := U_{i_1,a_1} \qquad\qquad C_{a_1,(i_2,\ldots,i_N)} := S_{a_1,a_1} V^{\dagger}_{a_1,i_2\ldots i_N} \ .$$

2nd iteration

$$C_{(a_1,i_2),(i_3\ldots,i_N)} = \sum_{a_2} U_{(a_1,i_2),a_2} S_{a_2,a_2} V^{\dagger}_{a_2,(i_3\ldots i_N)}$$

$$M_{i_2} := U_{a_1,i_2,a_2} \qquad\qquad C_{a_2,(i_3,\ldots,i_N)} := S_{a_2,a_2} V^{\dagger}_{a_2,i_3\ldots i_N} \ .$$



Matrix Product Operators (MPO)

$$H = \sum_{\{i\}} \sum_{\{i'\}} C_{i_1,\dots i_L}^{i'_1 \dots i'_L} |i_1 \dots i_L\rangle\langle i'_1 \dots i'_L|$$

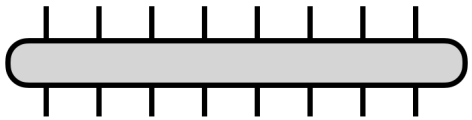
$H =$ 

Figure: Initial tensor.

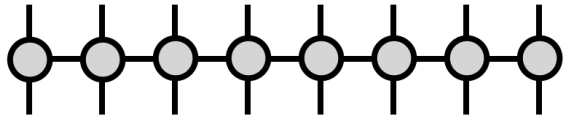
$H =$ 

Figure: MPO.

$$H = \sum_{\{i\}} \sum_{\{i'\}} M_{i_1}^{i'_1} \dots M_{i_N}^{i'_N} |i_1 \dots i_L\rangle\langle i'_1 \dots i'_L|$$

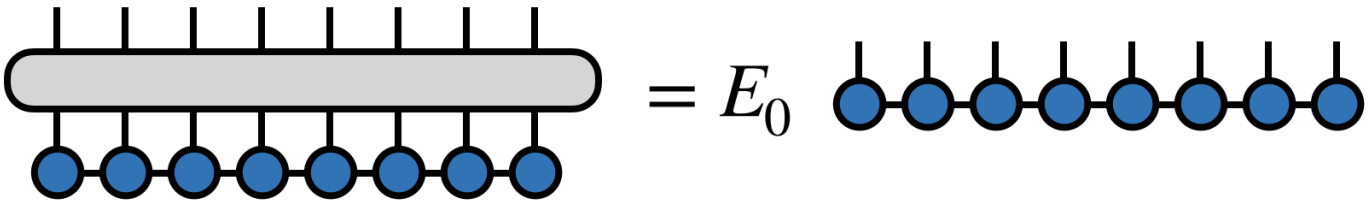
Density Matrix Renormalization Group (DMRG)

Goal: Find the state with minimum energy (ground state).

Remember the 2nd postulate of quantum mechanics:

$$H|\psi\rangle = E_0|\psi\rangle.$$

The equivalent TN would be:



$$\langle\psi|H|\psi\rangle / \langle\psi|\psi\rangle = E_0$$

1st iteration:

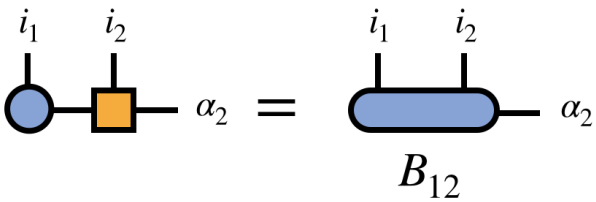


Figure: Two sites to be optimized.

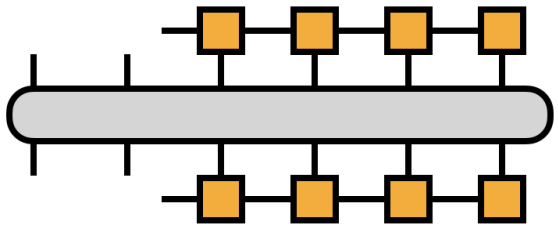
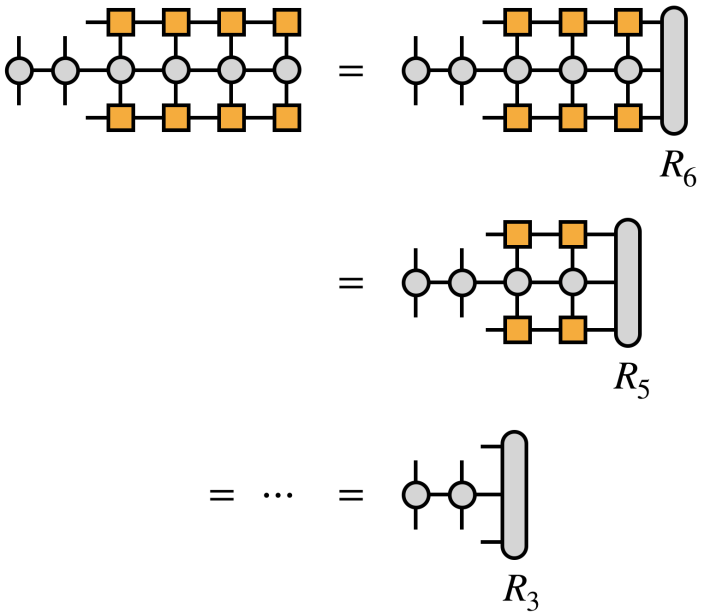


Figure: The remaining structure is fixed during the optimization.

1st iteration:



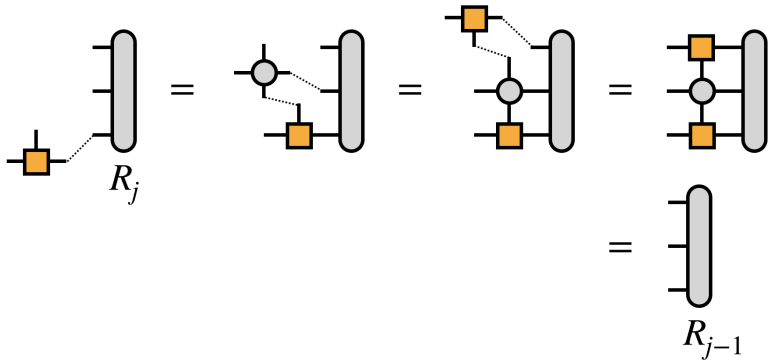


Figure: Ordering for efficient contraction.

1st iteration:

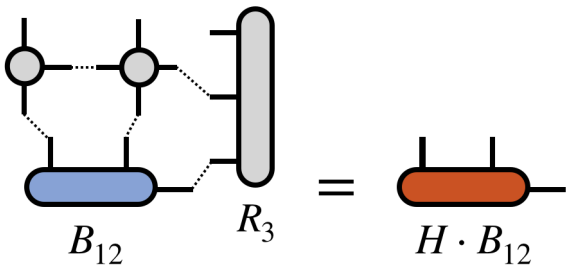


Figure: Optimization step: Find the lowest eigenvalue of H. Algorithm employed: **Lanczos method**.

Figure: Restore the MPS form.

2nd iteration:

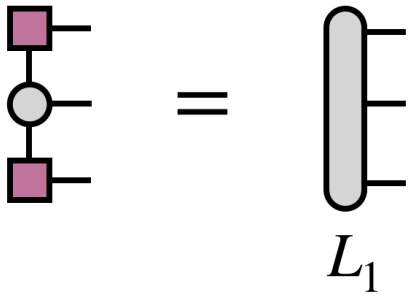


Figure: Fix the first tensor.

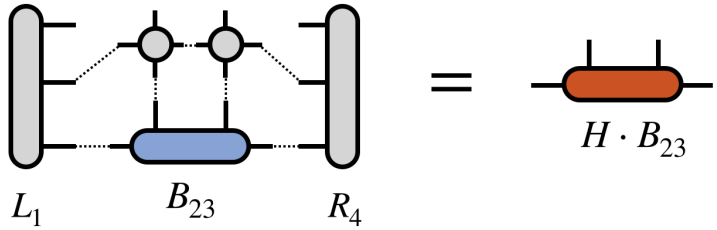


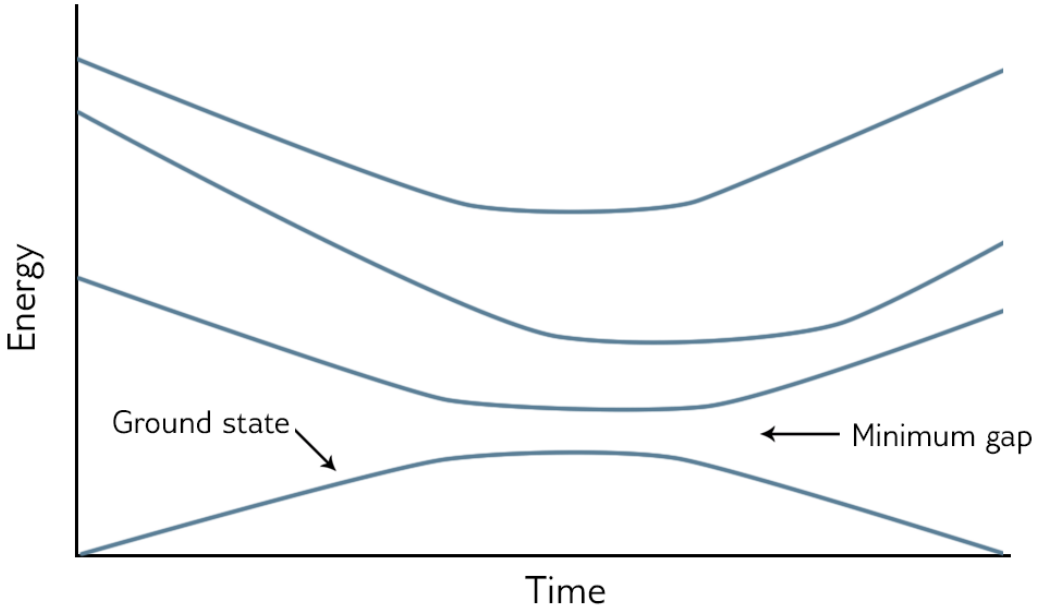
Figure: Optimization step for the 2nd and 3rd tensors.

Convergence criterion:

$$\text{var(E)} = \langle\psi|H^2|\psi\rangle - (\langle\psi|H|\psi\rangle)^2$$

Estimation of the minimum gap

Modified DMRG: Finds the second lowest energy of the system.



$$g_{min} = \min_{s \in [0,1]} (E_1(s) - E_0(s))$$

Minimum gap estimate using DMRG

$$\tilde{g}_{min} = \min_{i \in \{0,\dots,n\}} (\tilde{E}_1(t_i) - \tilde{E}_0(t_i)) \ .$$

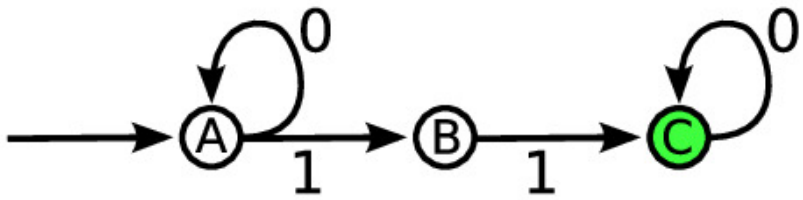
Table of Contents

Construction of MPSs and MPOs

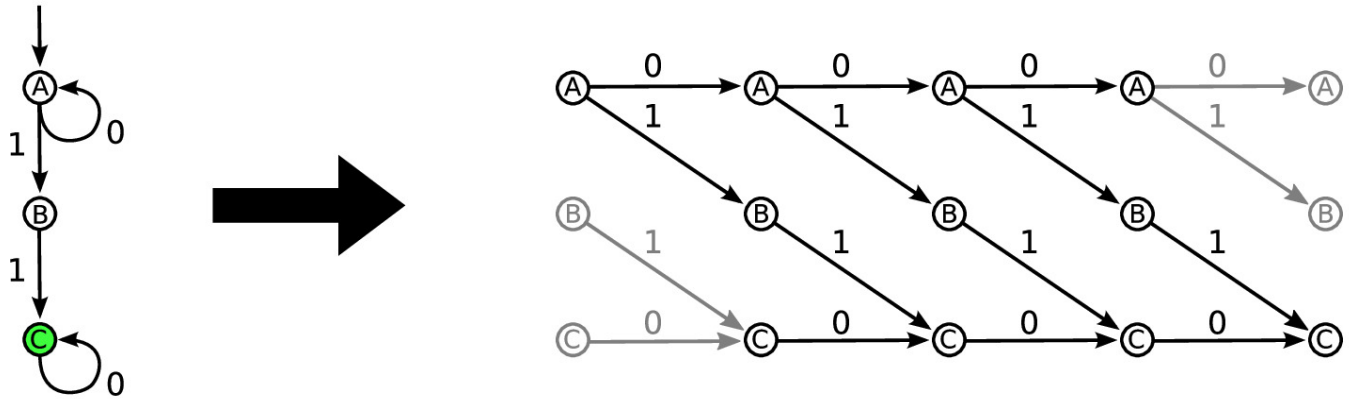
Let's say we want the MPS for the state

$$|11000\ldots\rangle + |01100\ldots\rangle + |00110\ldots\rangle + \ldots$$

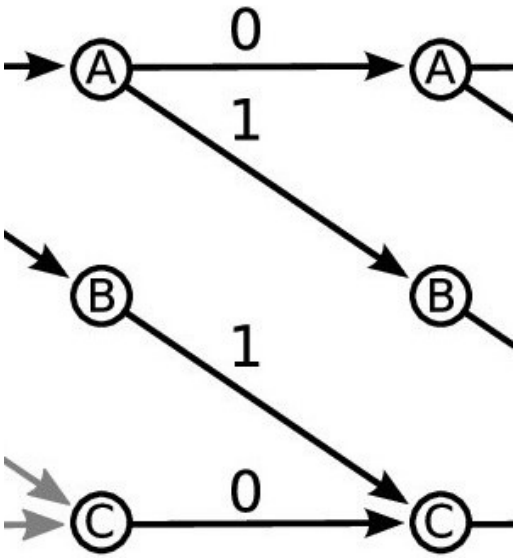
The summands form the language of the automaton



Which can be translated into an MPS diagram



Construction of MPSs and MPOs



$$\begin{bmatrix} \downarrow & \uparrow & 0 \\ 0 & 0 & \uparrow \\ 0 & 0 & \downarrow \end{bmatrix}$$

Final MPS (for 4 qubits):

$$\begin{bmatrix} \downarrow & \uparrow & 0 \end{bmatrix} \cdot \begin{bmatrix} \downarrow & \uparrow & 0 \\ 0 & 0 & \uparrow \\ 0 & 0 & \downarrow \end{bmatrix} \cdot \begin{bmatrix} \downarrow & \uparrow & 0 \\ 0 & 0 & \uparrow \\ 0 & 0 & \downarrow \end{bmatrix} \cdot \begin{bmatrix} 0 \\ \uparrow \\ \downarrow \end{bmatrix} = \downarrow\downarrow\uparrow\uparrow + \downarrow\uparrow\uparrow\downarrow + \uparrow\uparrow\downarrow\downarrow$$

To build an MPO, just use operators instead of states $\uparrow$, $\downarrow$.

Our contribution: construction of the MPO for our use case

Annealing Hamiltonian

$$H(t) = (1 - s(t)) \left[-\sum_{i=1}^{N} X_i \right] + s(t) \left[-\sum_{i,j} J_{ij} Z_i Z_j - \sum_j h_i Z_i \right]$$

$$t \in [0, T], \ T \in \mathbb{R}_{>0} \qquad s(0) = 0, \ s(T) = 1$$

We found in the literature an explicit construction of an MPO for the Hamiltonian

$$H = \sum_{k=1}^{N} X'_k + \sum_{k<l} c_{kl} Z_k \otimes Z_l$$

and applied this change of variables:

$$X'_i = -(1 - s(t)) \cdot X_i - s(t) \cdot h_i Z_i$$
$$c_{kl} = -s(t) \cdot J_{kl} \quad , J_{kl} \in \mathbb{R}$$

Our contribution: construction of the MPO for our use case

Furthermore,

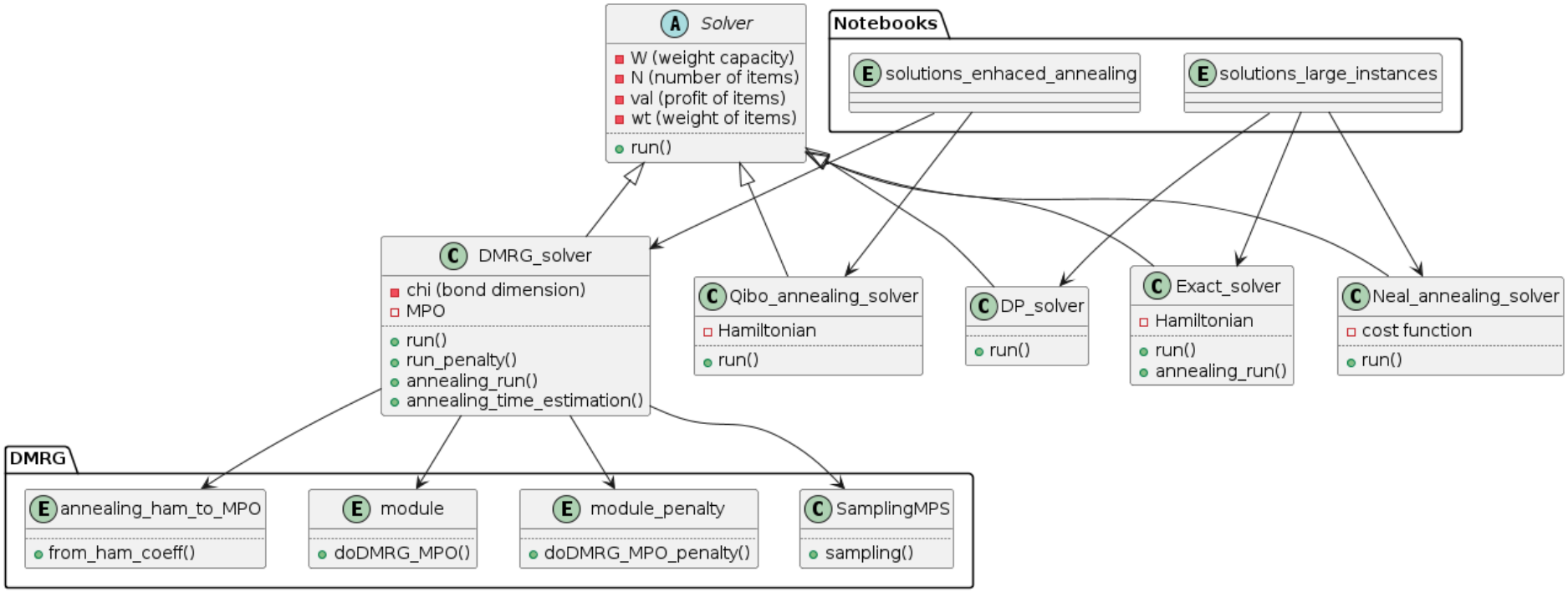
- Our implementation does not rely on $N-1$ auxiliary matrices they used to encode the coefficients.

$$c^{j_1,\ldots,j_N}_{i_1,\ldots,i_N} = A^{[1]}_{i_1 j_1} T^{[1]} A^{[2]}_{i_2 j_2} T^{[2]} \ldots T^{[N-1]} A^{[N]}_{i_N j_N}$$

- We corrected two errors in their implementation.
- We extended the code to handle any given N.

Table of Contents

Simplified UML diagram of the project



General formulation

- W : weight capacity of our knapsack.
- N : number of available items to put in the bag.
- Each item $i = 1, \ldots, N$ has a weight w_i and a value v_{ii}.
- v_{ij} : extra profit obtained if items i, j are both in the solution.
- x_i : binary decision variables.

Total weight:

$$\mathcal{W} = \sum_{i=1}^{N} w_i x_i$$

Total value:

$$\mathcal{V} = \sum_{i=1}^{N} \sum_{j=1}^{i} v_{ij} x_i x_j$$

Goal: maximize $\mathcal{V}$ subject to the constraint that $\mathcal{W} \leq W$.

Imposing inequality constraints: unbalanced penalization

Given the constraint

$$h(x) = W - \sum_i w_i x_i \geq 0.$$

Idea: minimize $-\mathcal{V} + e^{-h(x)}$. In a QUBO formulation

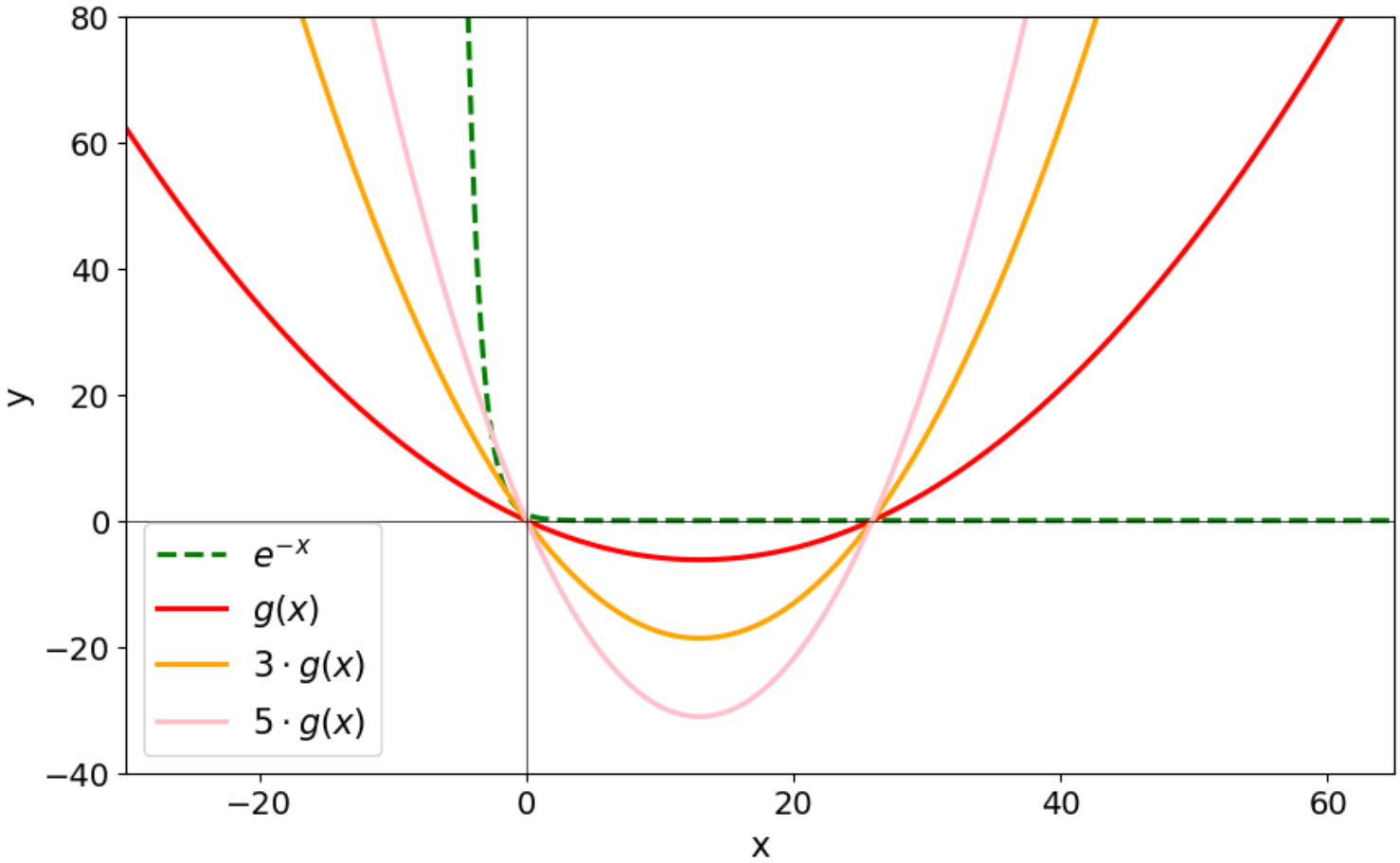
$$e^{-h(x)} \approx 1 - h(x) + \frac{1}{2} h(x)^2.$$

Instead, minimize $-\mathcal{V} + \lambda g(h(x))$, where

$$g(h(x)) := -0.9603 \cdot h(x) + 0.0371 \cdot h(x)^2.$$

Imposing inequality constraints: unbalanced penalization

$$\min_x -\mathcal{V} + \lambda g(h(x))$$



Dynamic programming approach

Algorithm 2 DP approach for QKP

Require: v_{ij} to be a triangular matrix

Initialize $V(k) \leftarrow 0$ for $k = 1, \dots W$ ▷ stores optimal profit for each weight

Initialize $S(k) \leftarrow \emptyset$ for $k = 1, \dots W$ ▷ stores solution items for each weight

for $k = 1, \dots, N$ **do**

  for $r = W, \dots, 1$ **do**

    if $w_k < r$ **then**

      $v \leftarrow V(r - w_k) + v_{ii}$

      for $i \in S(r - w_k)$ **do**

        $v \leftarrow v + v_{max(i,k),min(i,k)}$

      end for

      if $V(r) < v$ **then** ▷ if item k improves solution, update V and S

        $V(r) \leftarrow v$

        $S(r) \leftarrow S(r) \cup \{k\}$

      end if

    end if

  end for

end for

Output $V(W)$ and $S(W)$

Dynamic programming approach

Knapsack problem

- $O(N \cdot W)$.
- Returns optimal solution.

Quadratic knapsack problem

- $O(W \cdot N^2)$.
- It does not obey Bellman's principle of optimality.

$$V_k(x_k) = max_{d_k}\{g(x_k, d_k) + V_{k+1}(x_{k+1})\}, \quad k = 1, \ldots, N$$

Overview

Optimization problem
QUBO formulation

$$\sum_{i=1}^{N-1}\sum_{j>i}^{N} q_{ij}x_i x_j + \sum_{i=1}^{N} q_{ii}x_i$$

Ising Hamiltonian

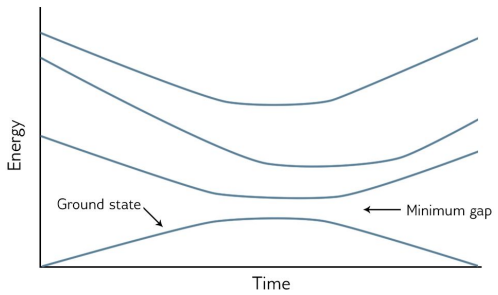
$$H = -\sum_{i,j} J_{ij}Z_i Z_j - \sum_j h_i Z_i$$

Annealing Hamiltonian

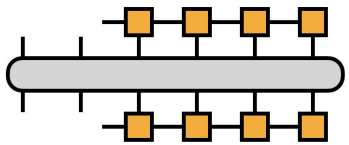
$$H(t) = (1 - s(t))H_0 + s(t)H_p$$

MPO

Adiabatic Quantum Computing (AQC)
Algorithm: quantum annealing



Tensor Networks (TN)
Algorithm: DMRG



QA results

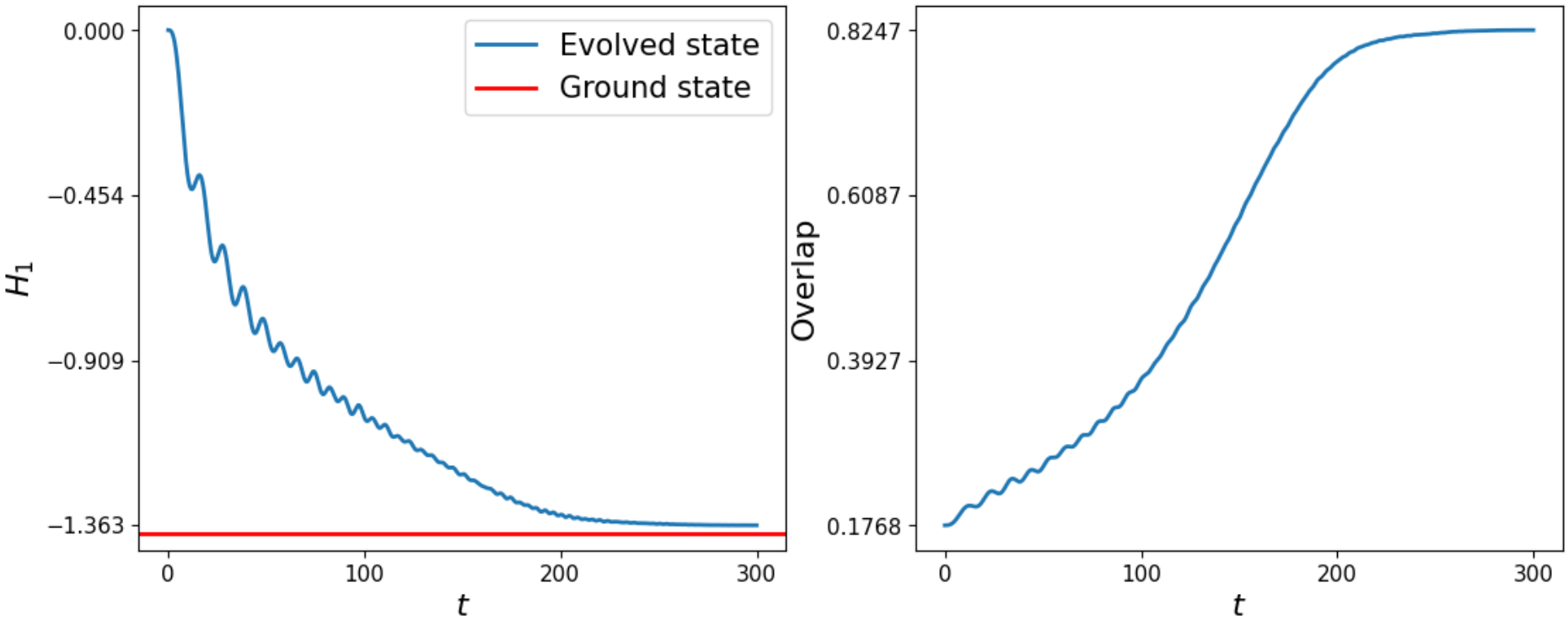
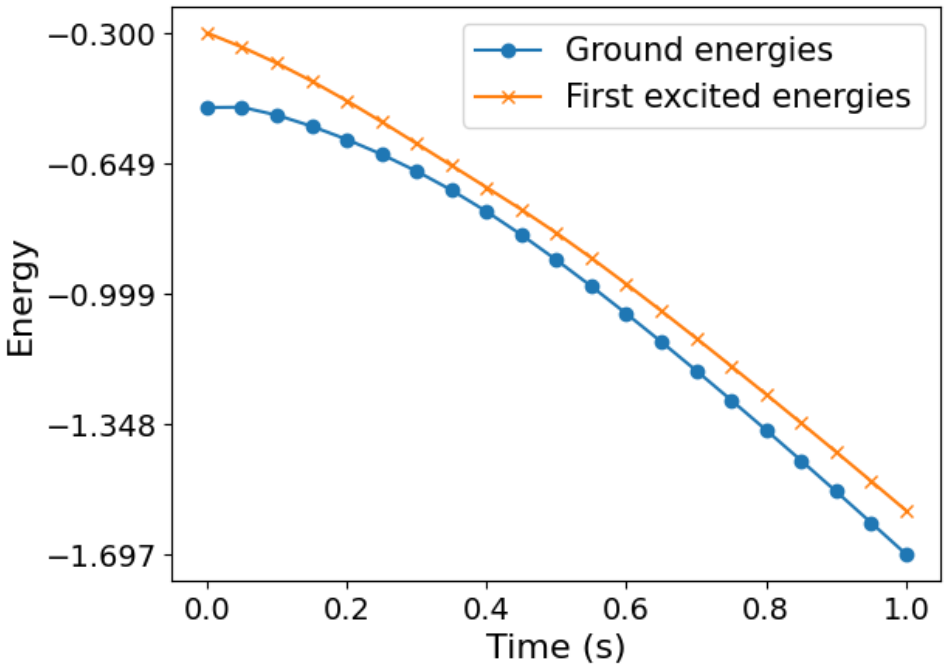
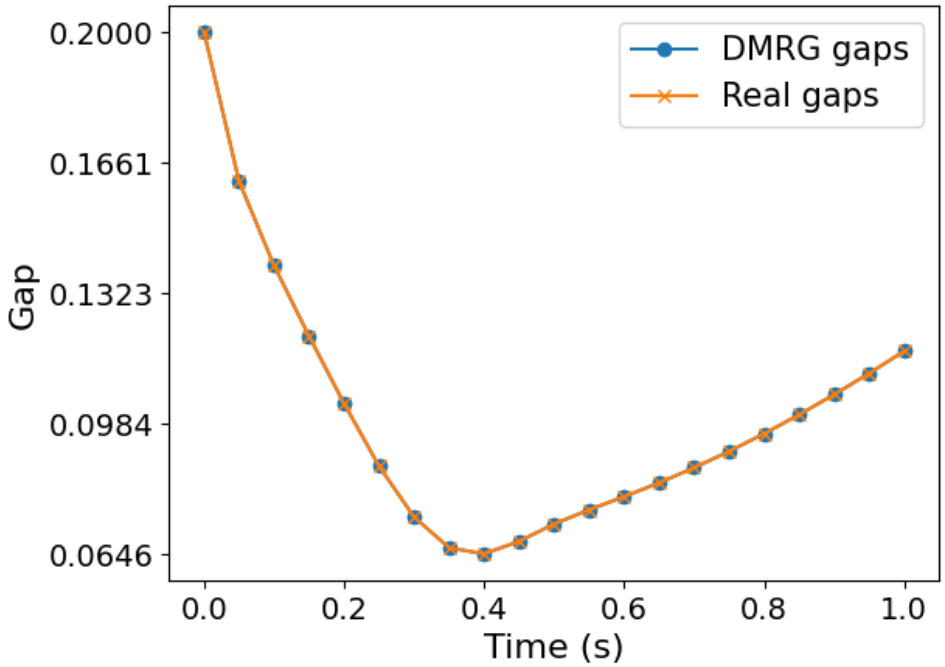


Figure: Plot of the simulated adiabatic evolution performed with the **Qibo library** for a QKP instance with **10 items**. The solution obtained matches the one derived using brute force.

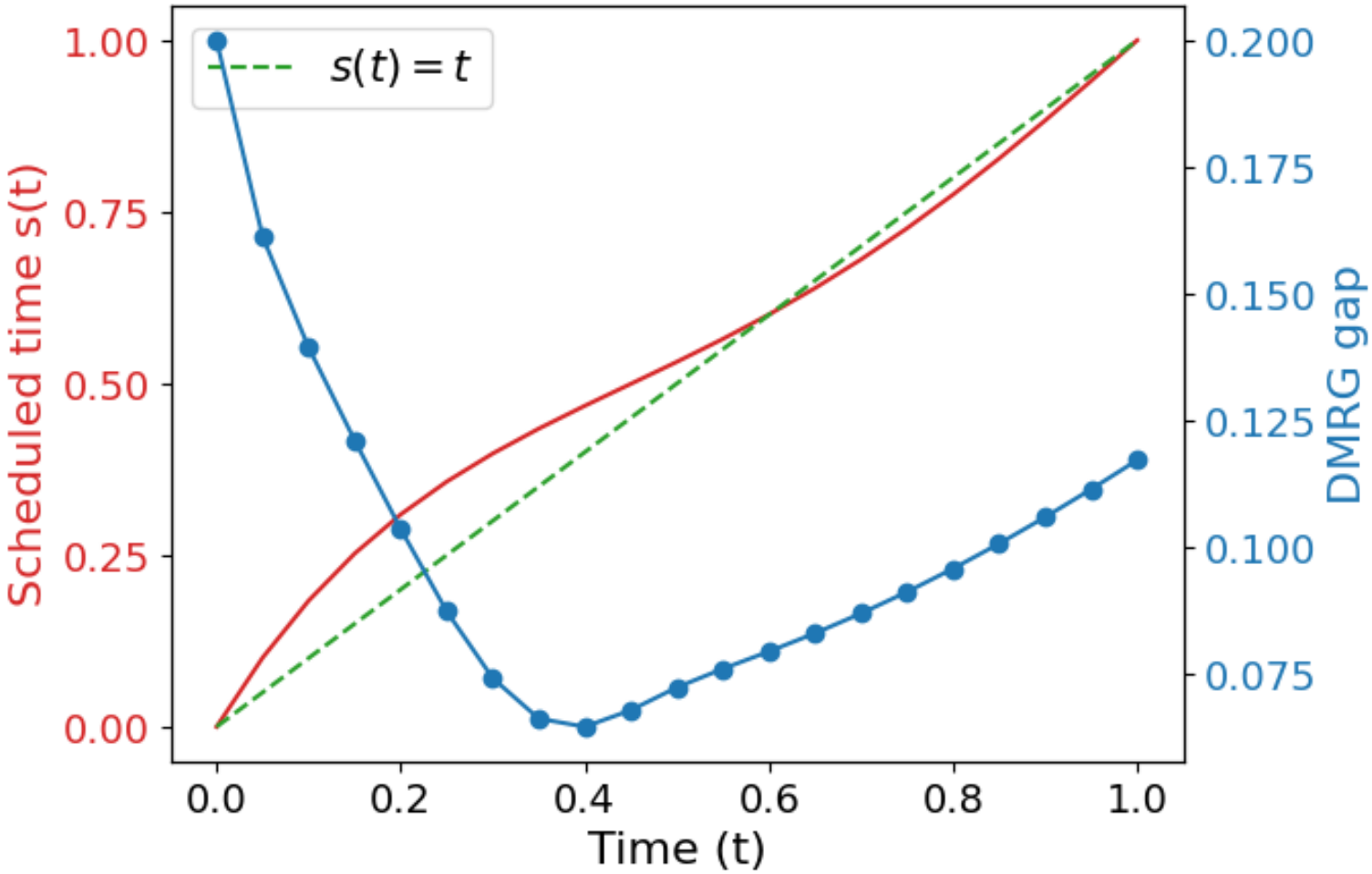
Computing the minimum gap with DMRG





Exploration of the annealing energy evolution.

Enhace QA with DMRG estimation



Performance on large-scale problem instances

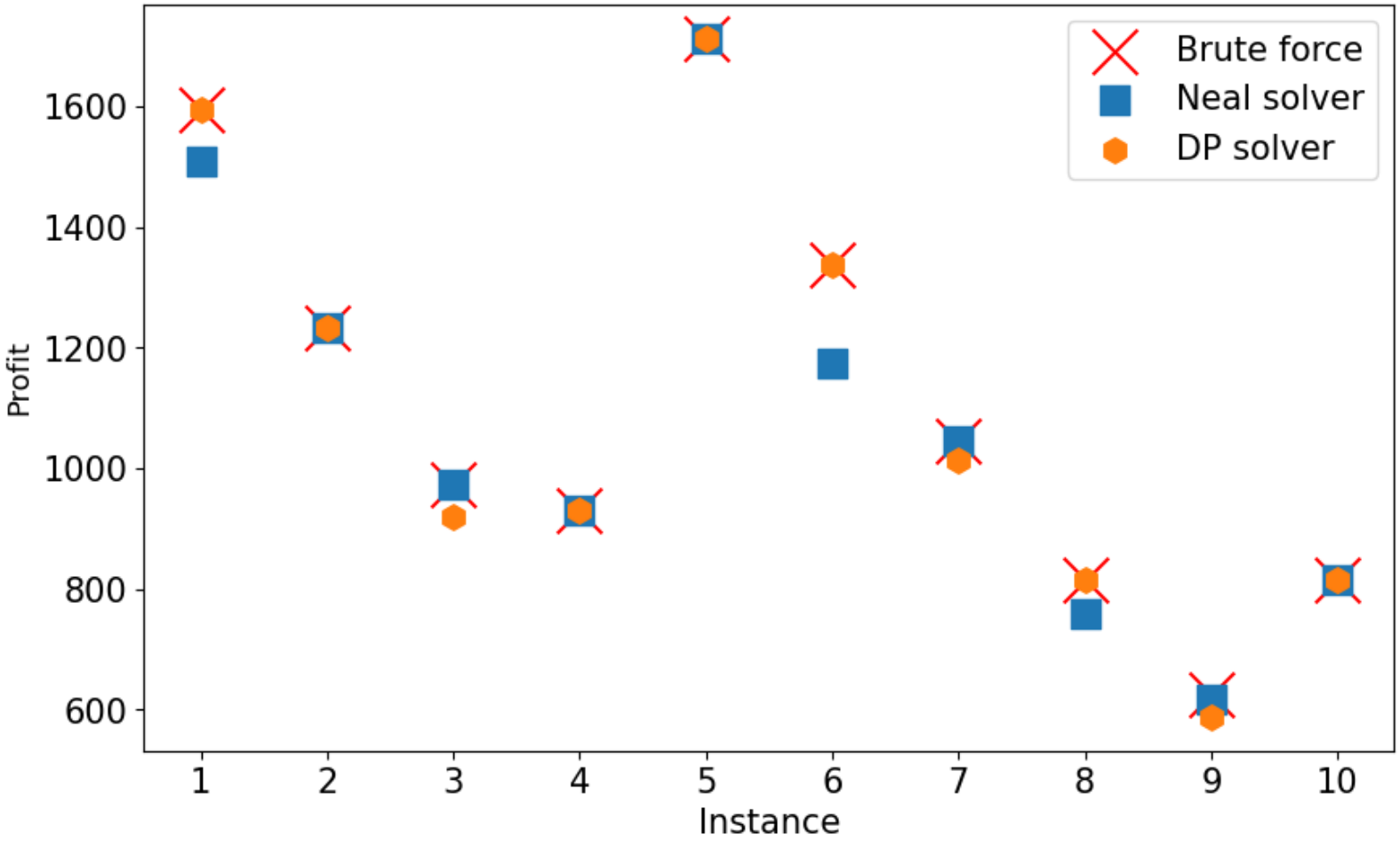


Figure: Random instances of the QKP involving **20 items** and a maximum capacity of 100.

Performance on large-scale problem instances

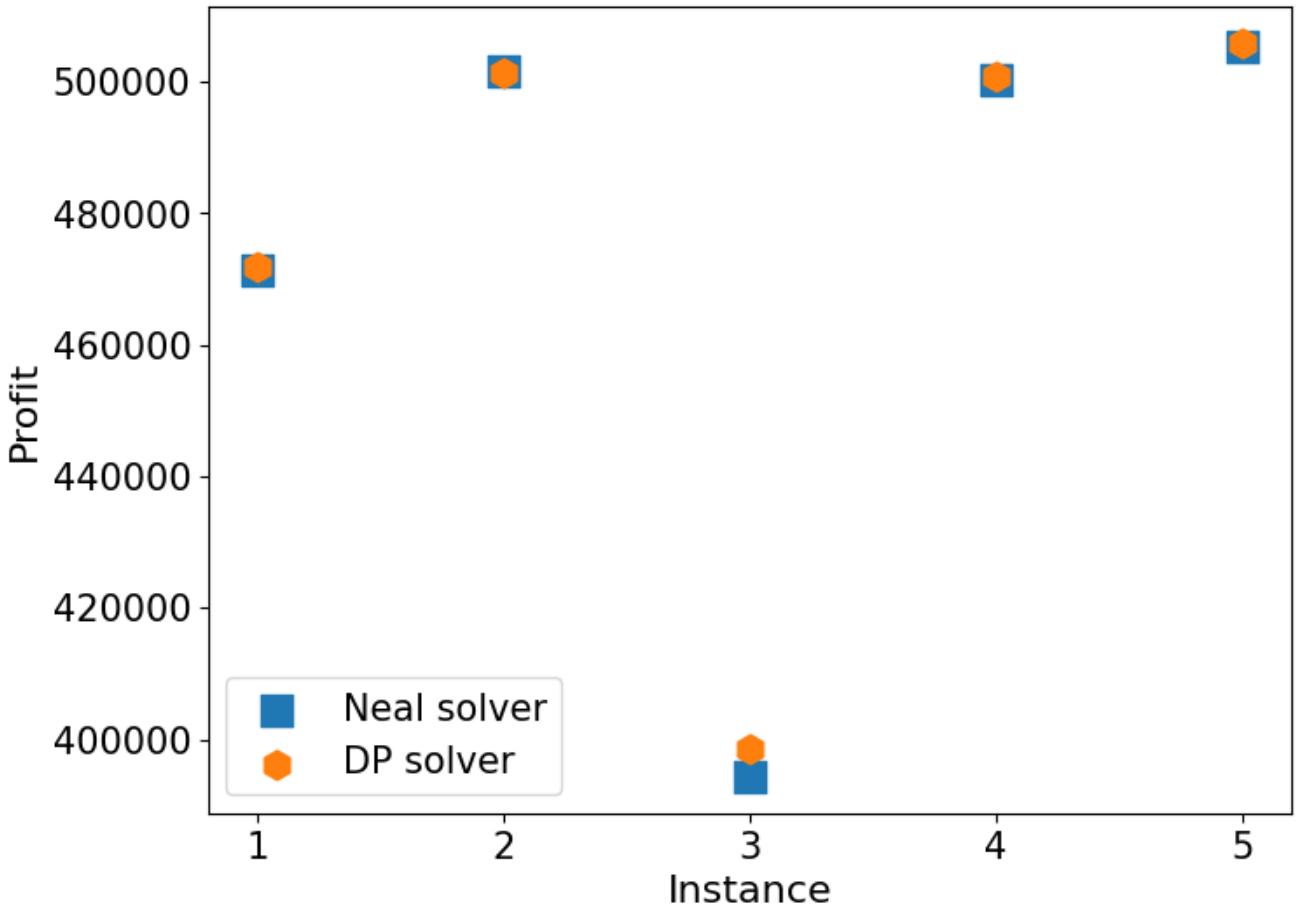


Figure: Random instances of the QKP involving **1000 items** and a maximum capacity of 1000.

Conclusions

Our technical contributions are:

- The QUBO formulation of the QKP without slack variables.
- The creation of an MPO that describes the annealing Hamiltonian of an Ising model.
- A proposal of an annealing scheduling time evolution for QA based on the study of the minimum gap with the DMRG.

Other relevant achievements:

- A mathematical explanation for people without a physics background covering quantum computing, quantum annealing, tensor networks, the DMRG algorithm, and how to create MPOs using finite automata.
- A Python implementation of all these algorithms, along with notebooks demonstrating their use.

Future work

- Further hyperparameter tuning of the QKP formulation and the DMRG algorithm parameters on a cluster is needed.
- Run quantum annealing on a real quantum annealer.
- Continue working on the custom annealing schedule timing.

End

Thank you for your attention

Table of Contents for the Appendix

Appendix A: Entanglement

$$|\psi\rangle = \left(\frac{1}{\sqrt{2}}, 0, 0, \frac{1}{\sqrt{2}}\right)^T = \frac{1}{\sqrt{2}}|00\rangle + \frac{1}{\sqrt{2}}|11\rangle$$

$$\mathbb{P}(|00\rangle) = 1/2 \qquad \mathbb{P}(|11\rangle) = 1/2$$
$$\mathbb{P}(|01\rangle) = 0 \qquad \mathbb{P}(|10\rangle) = 0$$

$$|\psi\rangle = (\alpha_0|0\rangle + \alpha_1|1\rangle) \otimes (\beta_0|0\rangle + \beta_1|1\rangle)$$

Postulate (unitary operator)

The time evolution of a closed quantum system is described by a unitary transformation. That is, the state $|\psi\rangle$ of the system at time t_1 is related to the state $|\psi'\rangle$ of the system at time t_2 by a **unitary operator** U which depends only on the times t_1 and t_2,

$$|\psi'\rangle = U|\psi\rangle$$

Unitary operators are the ones described by unitary matrices:

- Invertible, complex, and square.
- Satisfy $U^{-1} = U^\dagger$, where $U^\dagger$ is its conjugate transpose (Hermitian adjoint).

Hermitian adjoint satisfies: $U|\psi\rangle = |\phi\rangle \implies \langle\psi|U^\dagger = \langle\phi|$
Furthermore, if $U = U^\dagger$ they are known as Hermitian operators.

2nd postulate of quantum mechanics

The time evolution of the state of a closed quantum system is described by the time-depentent Schrödinger equation

$$i\hbar \frac{d|\phi(t)\rangle}{dt} = H(t)|\phi(t)\rangle$$

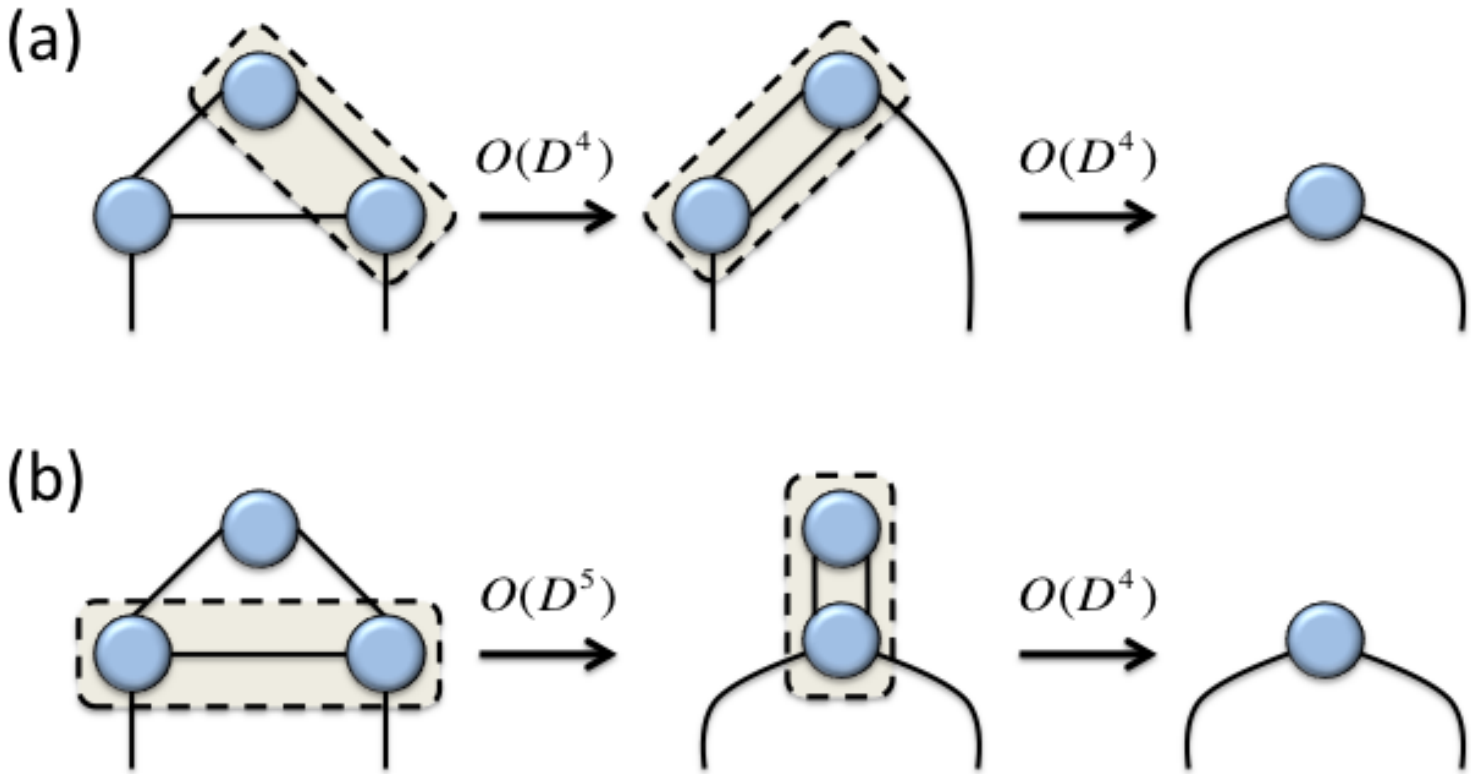
If H does NOT depend on time, we can solve the differential equation to obtain the time-independent Schrödinger equation

$$H|\psi_i\rangle = E_i|\psi_i\rangle \ ,$$

where

- $|\psi_i\rangle$ is an eigenstate. Since H is a Hermitian operator, an orthonormal basis of the Hilbert space composed of the eigenvectors of H exists.
- E_i is the energy of the state $|\psi_i\rangle$.

Appendix C: Contractions, the order matters



Appendix D: Gauge freedom and MPS orthogonality

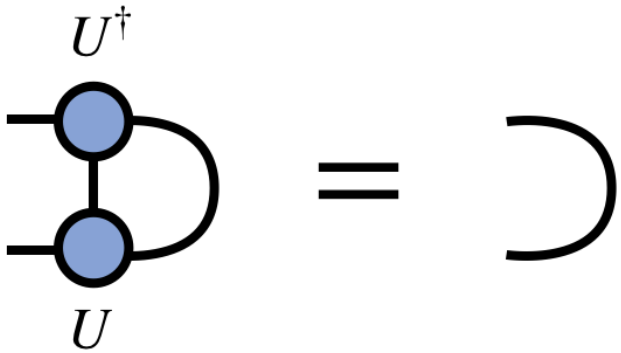


Figure: Right-orthogonal tensor in an MPS

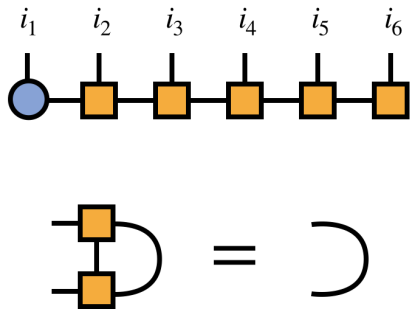
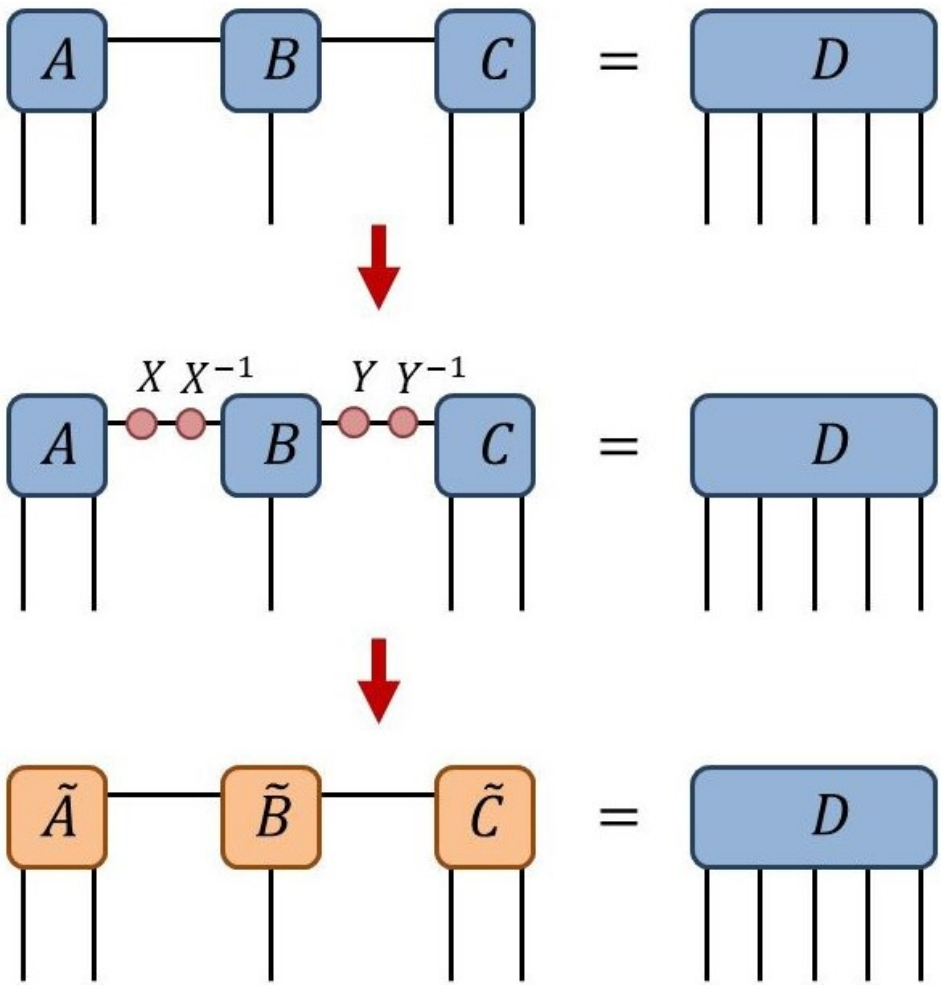


Figure: Right-orthogonal MPS

Appendix D: Gauge freedom and MPS orthogonality



Appendix E: Truncation with SVD

Any matrix $M \in \mathbb{C}^{m \times n}$ can be decomposed as

$$M = USV^{\dagger}$$

where

- $U \in \mathbb{C}^{m \times \min(m,n)}$ has orthonormal column vectors.
- $S = diag(s_1, s_2, ..., s_r, 0, ..., 0) \in \mathbb{R}_{\geq 0}^{\min(m,n) \times \min(m,n)}$, $r \leq min(m, n)$ where $s_1 \geq s_2 \geq ... \geq s_r$ are called *singular values of M*.
- $V^{\dagger} \in \mathbb{C}^{\min(m,n) \times n}$ has orthonormal row vectors.

Appendix E: Truncation with SVD

Given

$$M = USV^{\dagger} \qquad \text{where } S = diag(s_1, s_2, ..., s_r, 0, ..., 0),\ r = \text{rank } M$$

we can define

$$M_k = US_kV^{\dagger} \quad \text{where } S_k = diag(s_1, s_2, ..., s_k, 0, ..., 0),\ k < \text{rank } M \quad .$$

It has the property that

$$||M - M_k||_F \leq ||M - A||_F \quad \forall A \mid \text{rank}(A) = k$$

Appendix F: Entanglement in MPS representation

Theorem (Schmidt decomposition)

Given a bipartition of a Hilbert space $\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B$, any state $|\psi\rangle \in \mathcal{H}$ can be written as a *Schmidt decomposition* into orthonormal vectors $|\alpha_1\rangle, \ldots, |\alpha_n\rangle \in \mathcal{H}_A$ and $|\beta_1\rangle, \ldots |\beta_m\rangle \in \mathcal{H}_B$ as

$$|\psi\rangle = \sum_i s_i |\alpha_i\rangle \otimes |\beta_i\rangle.$$

Definition (entanglement entropy)

Given a bipartition of a Hilbert space $\mathcal{H} = \mathcal{H}_A \otimes \mathcal{H}_B$, and a state $|\psi\rangle \in \mathcal{H}$, the *entanglement entropy* of one part with the other is given by

$$S(A) = S(B) = -\sum_i |s_i|^2 \log |s_i|^2$$

where s_i are the Schmidt coefficients of $|\psi\rangle$.

Appendix G: The power method

Let $H \in \mathbb{C}^{n\times n}$ be a diagonalizable matrix. Let's suppose its eigenvalues satisfy

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \ldots \geq |\lambda_n|.$$

We build a series of vectors

$$z^{(k+1)} = Hz^{(k)}, \quad k = 0, 1, 2, \ldots$$

Let's consider a basis of eigenvectors $\{v_1, \ldots, v_n\}$. The series satisfies

$$z^{(k)} = H^k z^{(0)} = H^k \sum_i^n \alpha_i v_i = \sum_i^n \lambda_i^k \alpha_i v_i = \lambda_1^k \left[\alpha_1 v_1 + \sum_{i=2}^n \left(\frac{\lambda_i}{\lambda_1}\right)^k \alpha_i v_i\right].$$

As $(\lambda_i/\lambda_1)^k \to 0$ for $i > 1$ when $k \to \infty$ then the direction of $z^{(k)}$ tends towards the direction of v_1 (if $\alpha_1 \neq 0$)

Appendix H: Estimation of the minimum gap

Goal: Find the second lowest energy of the system (first excited state's energy).

Given the ground state ψ, run again DMRG on the modified Hamiltonian:

$$H' = H + w|\psi\rangle\langle\psi| \ ,$$

DMRG will then minimize

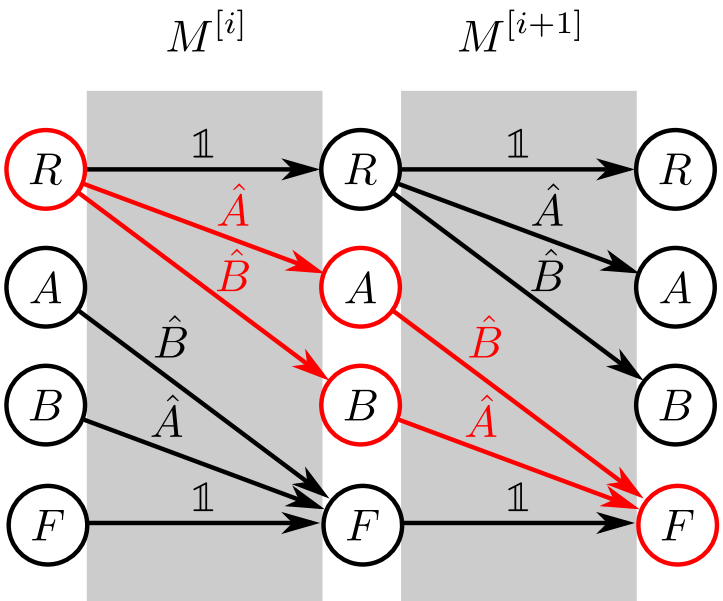
$$\langle\phi|\mathcal{H}|\phi\rangle + w\langle\phi|\psi\rangle\langle\psi|\phi\rangle \ .$$

Appendix I: Construction of MPOs

Let's say we want the MPO for an operator of the form

$$O = \sum_i \left(\hat{A}_i \hat{B}_{i+1} + \hat{B}_i \hat{A}_{i+1} \right) \quad .$$

The respective diagram and MPO matrix are



$$M = \begin{array}{c} \begin{matrix} R & A & B & F \end{matrix} \\ \begin{pmatrix} \mathbb{1} & \hat{A} & \hat{B} & 0 \\ 0 & 0 & 0 & \hat{B} \\ 0 & 0 & 0 & \hat{A} \\ 0 & 0 & 0 & \mathbb{1} \end{pmatrix} \end{array} \begin{matrix} R \\ A \\ B \\ F \end{matrix}$$

Appendix J: Tables of the MPO built

Rules for M_k, $\forall k$.

rule-number	(left, right)- input		output
1	$(1,1)$	$\rightarrow$	I
2	$(-1,-1)$	$\rightarrow$	I
3	$(-2,-1)$	$\rightarrow$	Z
4	$(1,-1)$	$\rightarrow$	$-(1-s)X + s \cdot h_k \; Z$

Additional rules for M_k, $k < \lfloor N/2 \rfloor$. The matrix dimension is $k+2 \times k+3$. $m = 2, \ldots, k$.

rule-number	(left, right)- input		output
5	$(1,2)$	$\rightarrow$	Z
6	$(1,k+2)$	$\rightarrow$	$s \cdot J_{k,k+1} \; Z$
7	$(m,m+1)$	$\rightarrow$	I
8	$(m,k+2)$	$\rightarrow$	$s \cdot J_{k-m+1,k+1} \; I$

Appendix J: Tables of the MPO built

Additional rules for $M_{\lfloor N/2 \rfloor}$. The matrix dimension is $\lfloor N/2 \rfloor + 2 \times \lfloor N/2 \rfloor + a$. $m = 2, \ldots, k$; $n = 2, \ldots, \lfloor N/2 \rfloor + a - 1$

rule-number	(left, right)- input		output
5	$(1, n)$	$\rightarrow$	$s \cdot J_{\lfloor N/2 \rfloor, N-n+2}\ Z$
6	(m, n)	$\rightarrow$	$s \cdot J_{\lfloor N/2 \rfloor - m+1, N-n+2}\ I$

Additional rules for M_k, $k > \lfloor N/2 \rfloor$. The matrix dimension is $N - k + 3 \times N - k + 2$. $m = 2, \ldots, N - k + 1$.

rule-number	(left, right)- input		output
5	$(1, m)$	$\rightarrow$	$s \cdot J_{k, N-m+2}\ Z$
6	(m, m)	$\rightarrow$	I

Appendix K: Custom scheduling time for QA

Annealing "velocity":

$$v(\tilde{t}) = \frac{g_{\tilde{t}} - g_{min} + \epsilon}{g_{max} - g_{min}} \qquad \tilde{t} = 0, \ldots, N \ .$$

Integrate to get the desired shape:

$$\tilde{s}(t) = \int p(t)\, dt \ \text{ where } p(t) \text{ is the polynomial that best fits } v(\tilde{t}) \ .$$

Shift and scale to ensure that $s(0) = 0$ and $s(1) = 1$:

$$s(t) = \frac{\tilde{s}(t) - \tilde{s}(0)}{\tilde{s}(1) - \tilde{s}(0)} \ .$$