Solving Data Assimilation on Quantum Annealing Machines

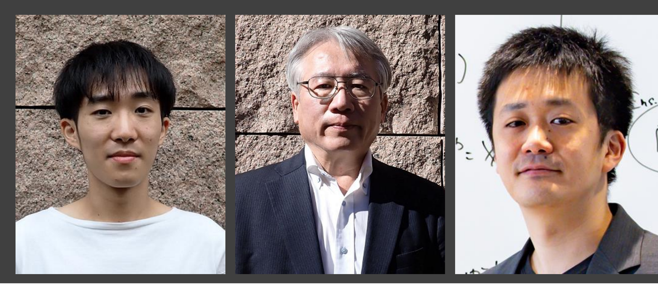
Submitted paper info.

Fumitoshi Kawasaki[1], Masanao Ohashi[2], Shunji Kotsuki[3]

[1] Graduate School, Chiba U. [2] CEReS, Chiba U., [3] IAAR, Chiba U.

(fkawasaki@chiba-u.jp)

(Kotsuki *et al.*, 2023, NPGD, doi: 10.5194/npg-2023-16)



Overview

- This study proposes solving data assimilation on quantum annealing (QA) machines.
- We have conducted experiments with Lorenz-96 and have shown that QA solves data assimilation (DA) faster than conventional approaches.

Introduction

Variational methods (*e.g.*, 4DVAR)

- Variational methods (*i.e.*, **optimization problems**) that iteratively compute the cost function to minimize it are used in NWP around the world.
- Due to the iterations, **huge computational resources are consumed** for DA in NWP.

Quantum Annealing (QA) machine

- Quantum computers have developed rapidly in recent years toward practical use.
- By employing a **QA machine**, one of the quantum computer types, **optimization problems can be solved much faster**.

➡ **Applying quantum annealing would solve the problem of computational resources in 4DVAR.**

Quantum Annealing for Data Assimilation

What's QA ?

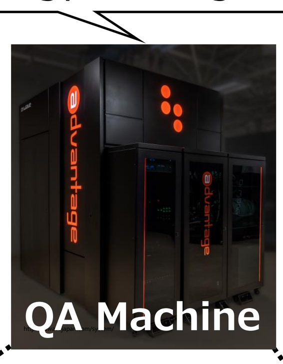
- QA is an algorithm for solving an optimization problem formulated in **QUBO** $\mathcal{H}(\mathbf{b})$.

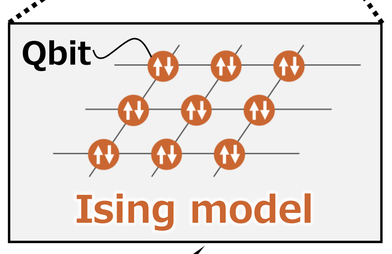
$$\mathcal{H}(\mathbf{b}) = \sum_{i<j} a_{ij} q_i q_j + \sum_i u_i q_i \quad (q_i \in \{0, +1\})$$

$$= \mathbf{b}^{\mathrm{T}}\mathbf{A}\mathbf{b} + \mathbf{u}^{\mathrm{T}}\mathbf{b} + C \quad (\mathbf{b}^{\mathrm{T}} = [q_0, q_1, \cdots, q_n])$$

- QUBO is compatible with the Ising model, which can be handled by QA machines.

Minimizing the energy of Ising model

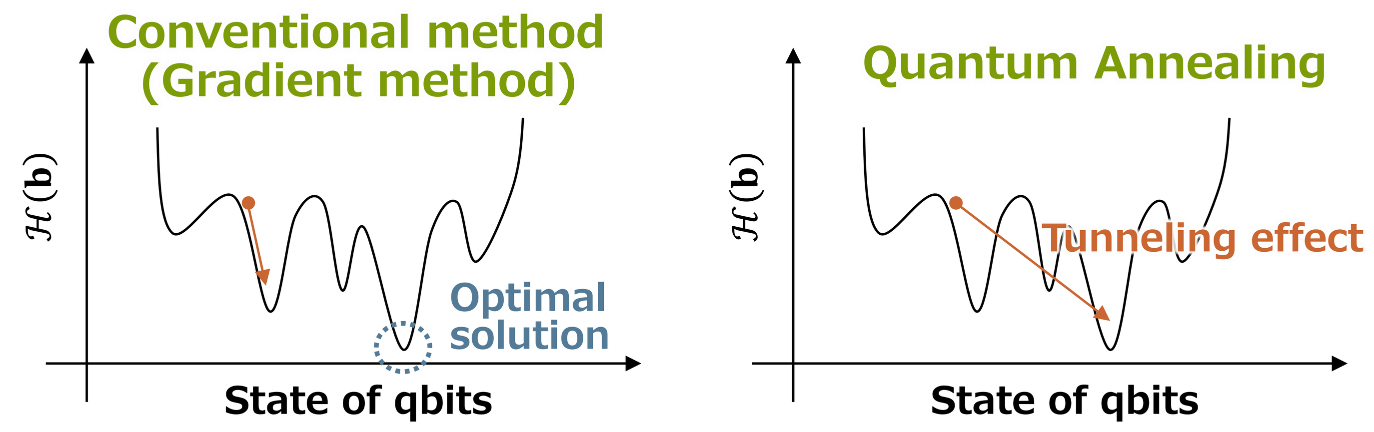




Representing magnetic materials

Characteristics of QA

- QA can solve an optimization problem much faster by employing the quantum effect (*e.g.*, tunneling effect).



Reformulation to QUBO

- When QA is applied, the cost function needs to be reformulated into QUBO.
- The cost function of 4DVAR can be reformulated into QUBO as follows;

$$J(\delta\mathbf{x}_0) = \delta\mathbf{x}_0^T\mathbf{B}^{-1}\delta\mathbf{x}_0 + \left[\mathbf{d}_{1:L}^o - \mathbf{H}_{1:L}\mathbf{M}_{1:L|0}\delta\mathbf{x}_0\right]^T\mathbf{R}_{1:L}^{-1}\left[\mathbf{d}_{1:L}^o - \mathbf{H}_{1:L}\mathbf{M}_{1:L|0}\delta\mathbf{x}_0\right]$$

$$\approx \tilde{J}(\delta\mathbf{x}_0) = \delta\mathbf{x}_0^T\mathbf{B}^{-1}\delta\mathbf{x}_0 + \left[\mathbf{d}_{1:L}^o - \mathbf{H}_{1:L}\widetilde{\mathbf{M}}_{1:L|0}\delta\mathbf{x}_0\right]^T\mathbf{R}_{1:L}^{-1}\left[\mathbf{d}_{1:L}^o - \mathbf{H}_{1:L}\widetilde{\mathbf{M}}_{1:L|0}\delta\mathbf{x}_0\right]$$

Approximation of TLM

$$= \delta\mathbf{x}_0^T\left[\mathbf{B}^{-1} + \widetilde{\mathbf{M}}_{1:L|0}^T\mathbf{H}_{1:L}^T\mathbf{R}_{1:L}^{-1}\mathbf{H}_{1:L}\widetilde{\mathbf{M}}_{1:L|0}\right]\delta\mathbf{x}_0 - 2(\mathbf{d}_{1:L}^o)^T\mathbf{R}_{1:L}^{-1}\mathbf{H}_{1:L}\widetilde{\mathbf{M}}_{1:L|0}\delta\mathbf{x}_0 + C$$

Binarization of $\delta\mathbf{x}_0$

$$\approx \mathcal{H}(\mathbf{b}) = \left(\frac{1}{\alpha}\mathbf{G}\mathbf{b}\right)^T\left[\mathbf{B}^{-1} + \widetilde{\mathbf{M}}_{1:L|0}^T\mathbf{H}_{1:L}^T\mathbf{R}_{1:L}^{-1}\mathbf{H}_{1:L}\widetilde{\mathbf{M}}_{1:L|0}\right]\left(\frac{1}{\alpha}\mathbf{G}\mathbf{b}\right) - 2(\mathbf{d}_{1:L}^o)^T\mathbf{R}_{1:L}^{-1}\mathbf{H}_{1:L}\widetilde{\mathbf{M}}_{1:L|0}\left(\frac{1}{\alpha}\mathbf{G}\mathbf{b}\right) + C$$

$$= \mathbf{b}^{\mathrm{T}}\mathbf{A}\mathbf{b} + \mathbf{u}^{\mathrm{T}}\mathbf{b} + C$$

⬅ **QUBO !!**

$$\mathbf{z}_{1:L} \equiv [\mathbf{z}_1 \quad \cdots \quad \mathbf{z}_L]^T$$

$$\mathbf{Z}_{1:L} \equiv \begin{bmatrix} \mathbf{Z}_1 & & \mathbf{O} \\ & \ddots & \\ \mathbf{O} & & \mathbf{Z}_L \end{bmatrix}$$

$$M_{i|i-1}\left(M_{i-1|0}\left(\mathbf{x}_0^f + \delta\mathbf{x}_0\right) + \boldsymbol{\varepsilon}\right) = M_{i|0}\left(\mathbf{x}_0^f + \delta\mathbf{x}_0\right) + \mathbf{M}_{i|i-1}\boldsymbol{\varepsilon}$$

$$M_{i|i-1}\left(M_{i-1|0}\left(\mathbf{x}_0^f\right) + \boldsymbol{\varepsilon}\right) = M_{i|0}\left(\mathbf{x}_0^f\right) + \widetilde{\mathbf{M}}_{i|i-1}\boldsymbol{\varepsilon}$$

$$\mathbf{G} = \begin{bmatrix} \mathbf{g}^T & & \mathbf{O} \\ & \ddots & \\ \mathbf{O} & & \mathbf{g}^T \end{bmatrix}$$

$$\mathbf{g}^T = \left[-2^{Z-1}, 2^{Z-2}, \cdots, 2^1, 2^0\right]$$

Experiments & Results

- **OSSE with Lorenz-96 (40 variable)**
- DA method : 4DVAR
- DA window : 2 days
- $dt = 0.05$
- Obs. noise ~ $N(0, 1)$
- Num. of DA : 50 times
- All grids are observed

	Formula	Opt. Algo.	System
QUO-NL	$J(\delta\mathbf{x}_0)$	Quasi-Newton method	CPU
QUO-L	$\tilde{J}(\delta\mathbf{x}_0)$	Quasi-Newton method	CPU
QUBO-Sim	$\mathcal{H}(\mathbf{b})$	Simulated QA	CPU
QUBO-Phy	$\mathcal{H}(\mathbf{b})$	QA	QPU