Designing metamaterials with D-Wave 2000Q quantum annealer

< arXiv: 1902.06573 >

Koki Kitai , Jiang Guo, Shenghong Ju, Shu Tanaka,

Koji Tsuda, Junichiro Shiomi, Ryo Tamura

Overview

Storyline that we are expecting from our work.

1. To solve an combinatorial optimization with QA, we usually use the representation of the objective function as Ising/QUBO.
2. It is less likely existing in real-world.
3. We propose a method to tackle any binary combinatorial optimization, by learning the QUBO representation dynamically.
   - learn QUBO in a data-driven way by factorization machine
   - optimize the constructed QUBO by QA
4. We can take the advantage of QA on wider variety of tasks.
   - Harnessing the combinatorial explosion.

Agenda

1. Algorithm of black-box optimization
2. Application on metamaterial design
3. Introduction to fmbqm

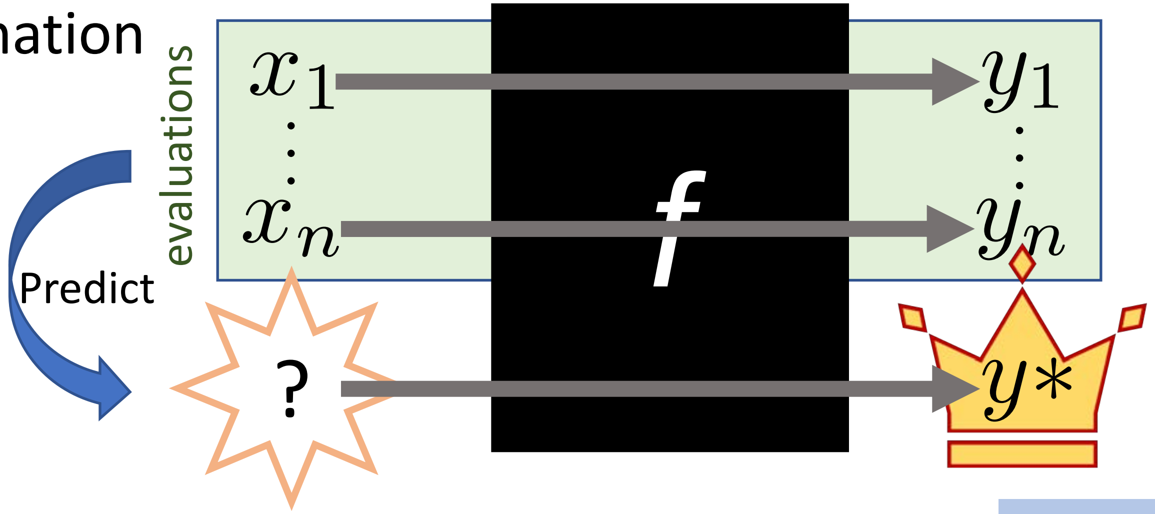
Background: **Black-box optimization**

Black-box function f receives some input x, and returns output value $f(x)$, while other information such as analytical form of it or derivative with respect to x are not available.

Evaluation of black-box function is often expensive.

- Efficiency of wind farm layout
- Stability of protein/molecular conformation
- Property of designed materials

Black-box optimization is to find x which minimizes $f(x)$ with as few evaluations as possible.

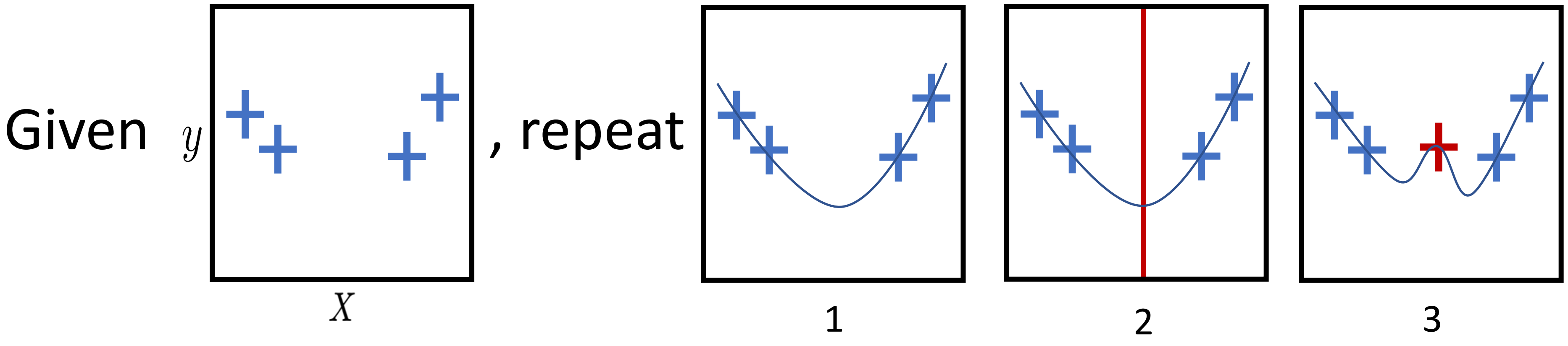


Background: **Surrogate-based method**

Surrogate-based method is an approach relying on regression model.

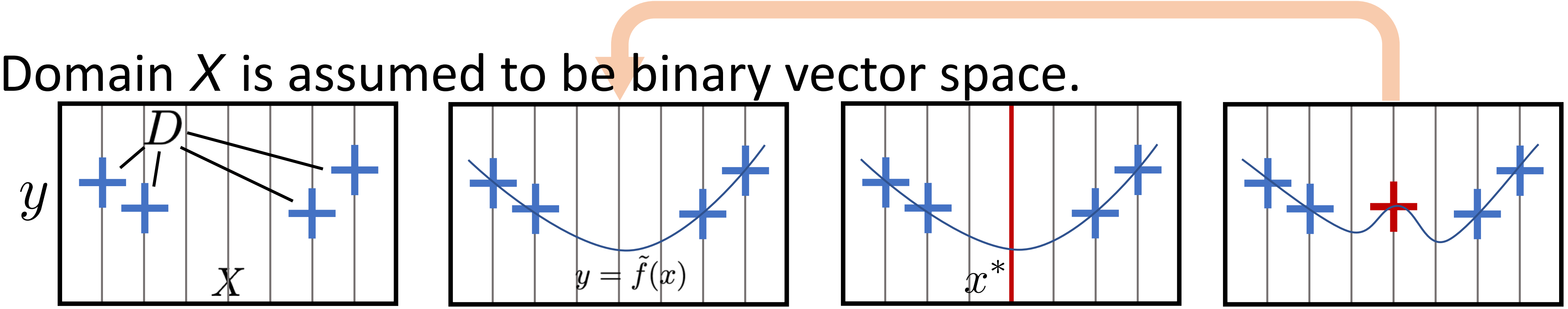
1. Train a regression model $\tilde{f}(x)$ based on dataset.
2. Find x which minimizes the trained model $\tilde{f}(x)$. ($\leftarrow$ surrogate)
    - Sometimes other index rather than the raw $\tilde{f}(x)$ is used.
3. On found x, we evaluate $f(x)$ and join it to the dataset.

Back to 1



For binary combinatorial optimization

Domain X is assumed to be binary vector space.



init.	Regression	Selection	Evaluation
Define a training dataset D	Train a regression model $y = \tilde{f}(x)$	$x^* = \underset{x \in X}{\mathrm{argmin}}\ \tilde{f}(x)$	Evaluate $f(x^*)$ and add data to D

- Regression model can be trained by gradient descent. (e.g. Adam [Kingma, Ba, 2014])
- Selection part suffers from combinatorial explosion 🤯 (➔😀 if QA)

Regression by Factorization Machine

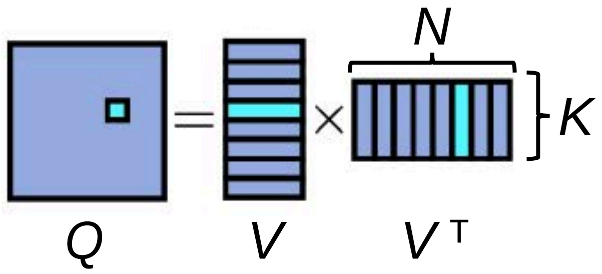
We used Factorization Machine (FM) [Rendle, 2012] as a regression model.

The function have two types of model parameters, *h* and *v*.

$$\tilde{f}(x) = \sum_{i}^{N} h_i x_i + \sum_{i<j}^{N} \sum_{f=1}^{K} v_{i,f} v_{j,f} x_i x_j$$

linear coeff.　　pairwise interaction

It can be seen that the matrix representing pairwise interaction term (QUBO's Q_{ij}) is approximated by a matrix V of rank K. The reduction of the number of parameters is intended to avoid **overfitting problem**.



Selection by quantum annealing

Because of the relationship between V and Q, FM model is easily converted to Quadratic Unconstrained Binary Optimization (QUBO) problem.

FM $$\tilde{f}(x) = \sum_{i}^{N} h_i x_i + \sum_{i<j}^{N} \sum_{f=1}^{K} v_{i,f} v_{j,f} x_i x_j$$

QA incompatible

QUBO $$H(x) = \sum_{i}^{N} h_i x_i + \sum_{i<j}^{N} q_{i,j} x_i x_j$$

😀 QA compatible

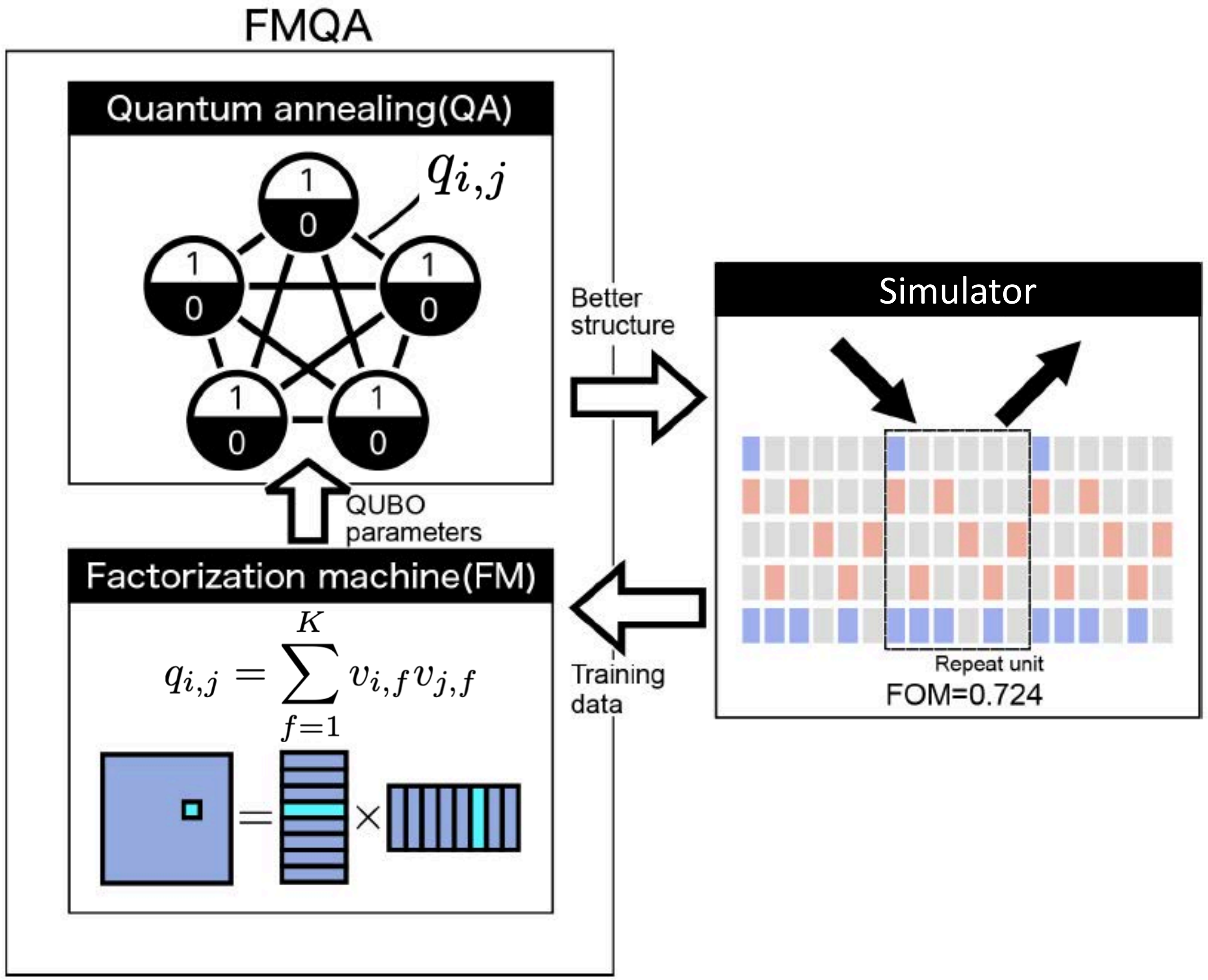
Solving the QUBO problem means, searching for x which minimizes $\tilde{f}(x)$, from the binary vector space X.

Proposed method

The surrogate-based black-box optimization method we proposed is termed as FMQA.

This framework is applicable to any binary combinatorial optimization problems.

The problems related with model/QA accuracy should be inspected carefully, though.



Use case for metamaterial design

Comparison with existing method

Bayesian Optimization (BO) is a popular surrogate-based method.

	FMQA (proposed)	BO
Regression	Factorization Machine (parametric)	Gaussian Process (non-parametric)
Selection	Quantum Annealing	Exhaustive(+Random) Search

Generally, expressive power of Gaussian Process (GP) is stronger than that of FM. But GP does not scale well.

For selection part, FMQA is superior to BO for its use of QA rather than exhaustive search.

Agenda

1. Algorithm of black-box optimization
2. Application on metamaterial design
3. Introduction to fmbqm

Automated metamaterial design

Metamaterial is material that...

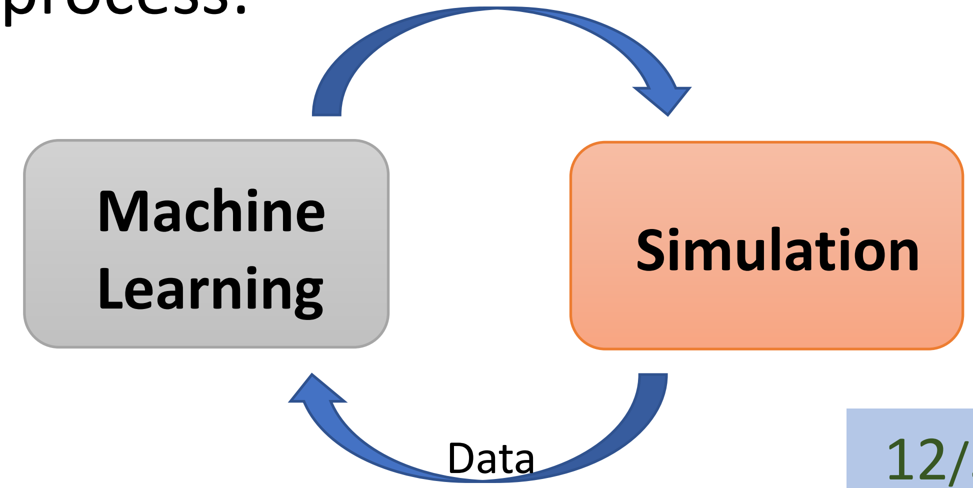
- is composed of some basic materials
- has a special structure, to achieve an unusual property

The search space grows exponentially to the number of building-blocks.

How the structure affects the property is black-box function

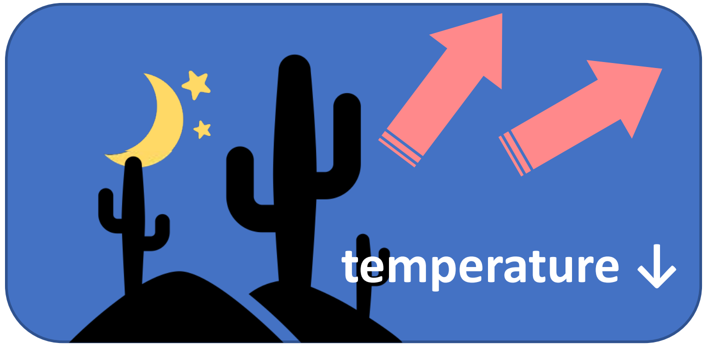
The key is to automate and accelerate the process:

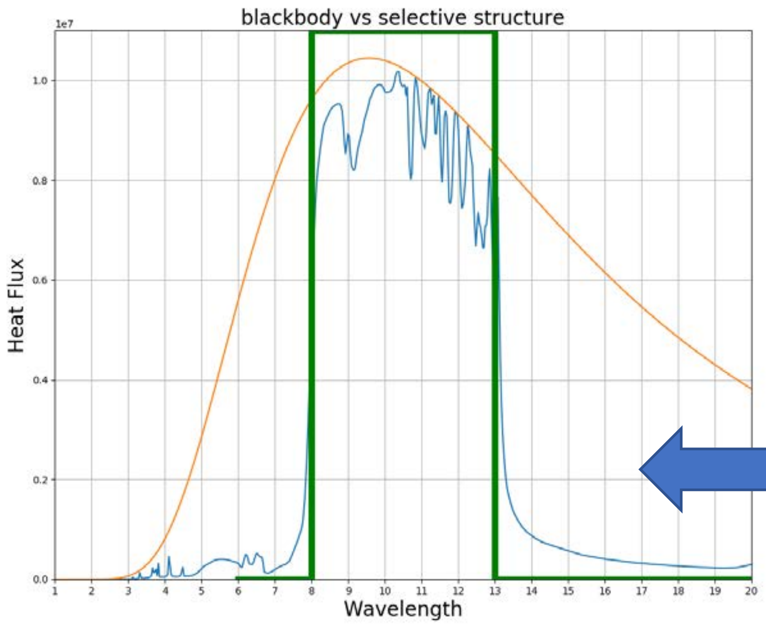
- Evaluation by computer simulation
- Learning by proposed method



Demo case - Thermal radiator

Radiative cooling is an effect that the heat escape from body as emitting light. (well known for the temperature at night in desert)



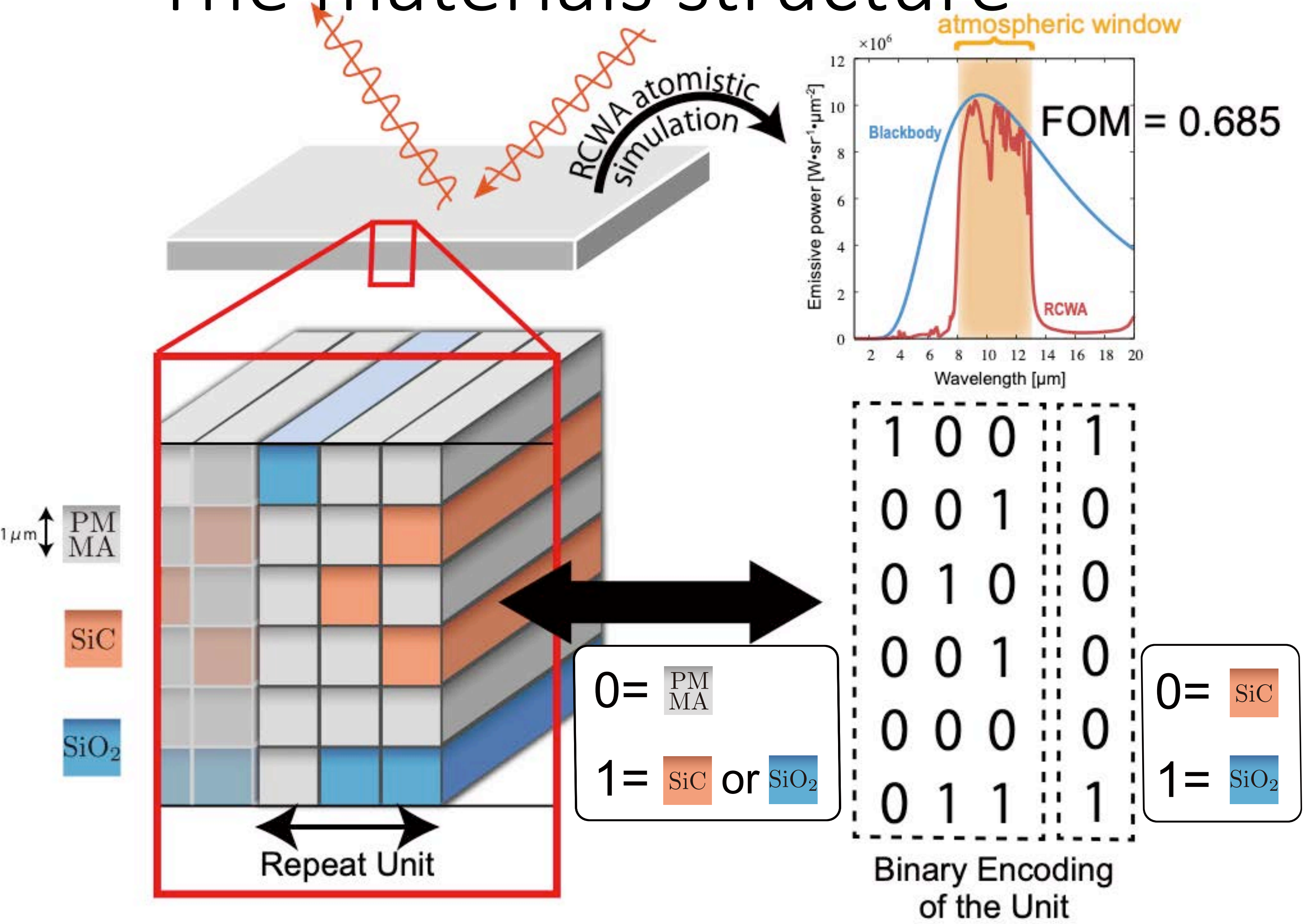


We can use the effect for powerless cooling

➔ **Thermal radiator**

Radiative cooling is most effective when the radiation spectrum concentrates on atmospheric window (8-13 μm wavelength). The spectrum can be calculated by Rigorous Coupled-Wave Analysis (RCWA) simulation.

The materials structure



Our thermal radiator is designed as a stack of fibers of SiC, SiO_2, and PMMA. The structure is nicely encoded into a binary array.

Only 1 type of Si-based fiber within a layer. (limitation by binary representation)

The concordance of the spectrum is calculated as a score called Figure of Merit (FOM), which should be maximized as close to 1.0 as possible.

Designing 4x3 structure

- FOM maximization on small size problem
  - the number of layers L=4
  - the number of columns C=3
  - 16 bits for encoding

0	1	0	0
1	0	0	0
0	1	1	0
0	1	1	1

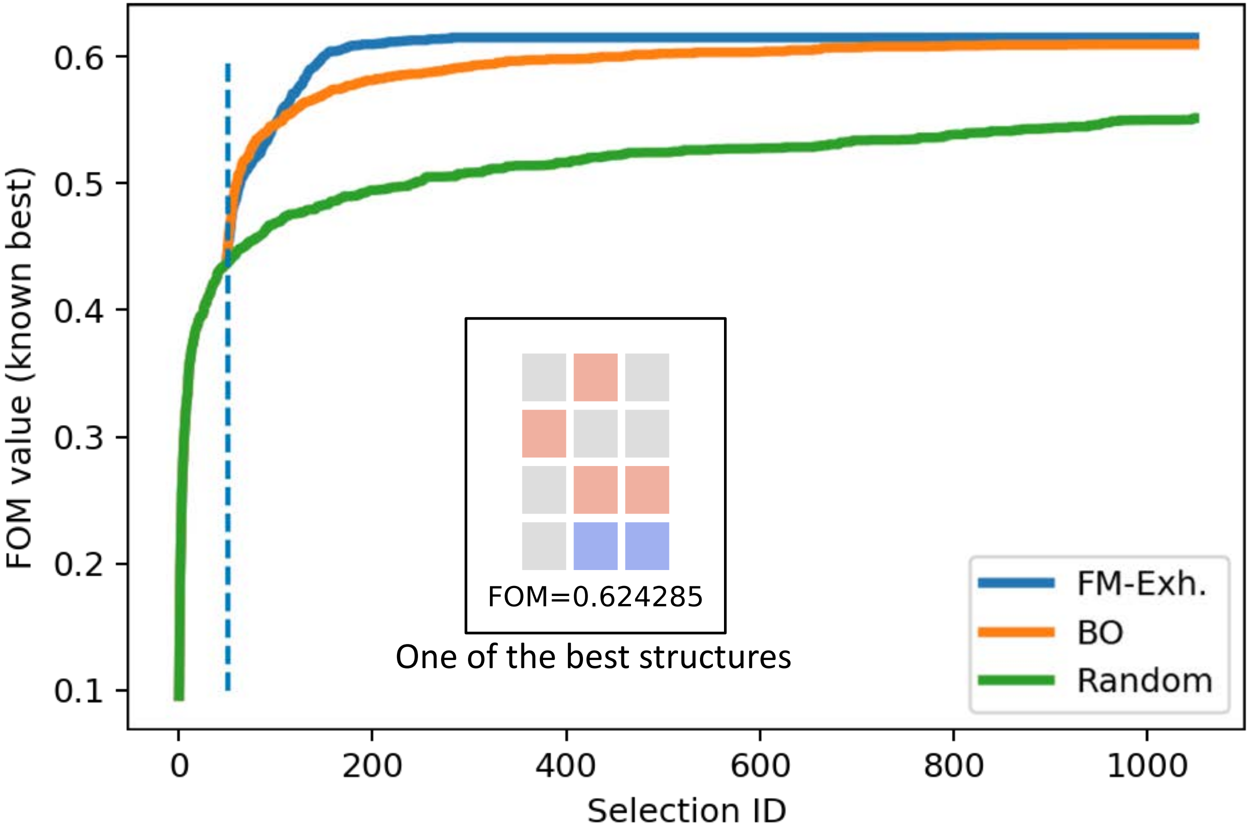
The best structure for this setting

- Compared methods

	FM-Exh.	BO	Random
Regression	FM	GP	None
Selection	Exhaustive	Exhaustive	Random

  - The main purpose is to compare FM and GP
  - Exhaustive search (over 2^{16}=65536 candidates) can be conducted.

Designing 4x3 structure - result



This graph means the best FOM obtained within the numbers of samples.

The first 50 samples were taken at random on all methods as initial dataset.

Our method reached the best the fastest.

	FM-Exh.	BO	Random
Reg.	FM	GP	None
Sel.	Exh.	Exh.	Random

Designing 6x3 structure

- On middle size problem
  - the number of layers L=6
  - the number of columns C=3
  - 24 bits for encoding

1	0	0	1
0	0	1	0
0	1	0	0
0	0	1	0
0	0	0	0
0	1	1	1

The best structure
for this setting

- Compared methods

	FMQA	~~BO~~	Random
Regression	FM	~~GP~~	None
Selection	QA	~~Exhaustive~~	Random

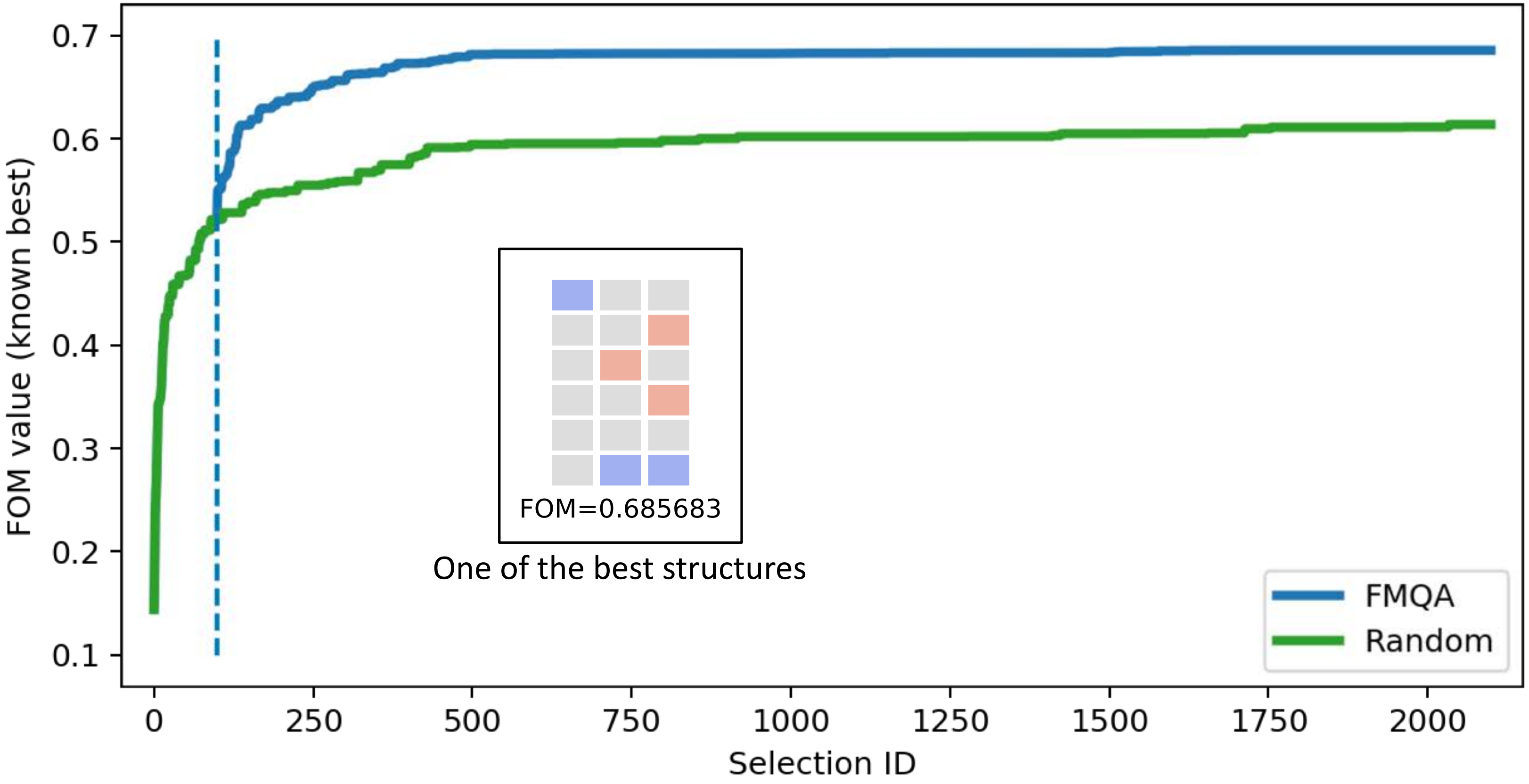
Almost default
settings for QA

D-Wave 2000Q_2_1
num_reads = 50
anneal_time = 20us

  - The main purpose is to check if our method scales by QA.
  - Exhaustive search was not conducted due to the large search space.

Designing 6x3 structure - result

The first 100 samples were taken at random as initial dataset.
FMQA worked fine as is in 4x3 structure.

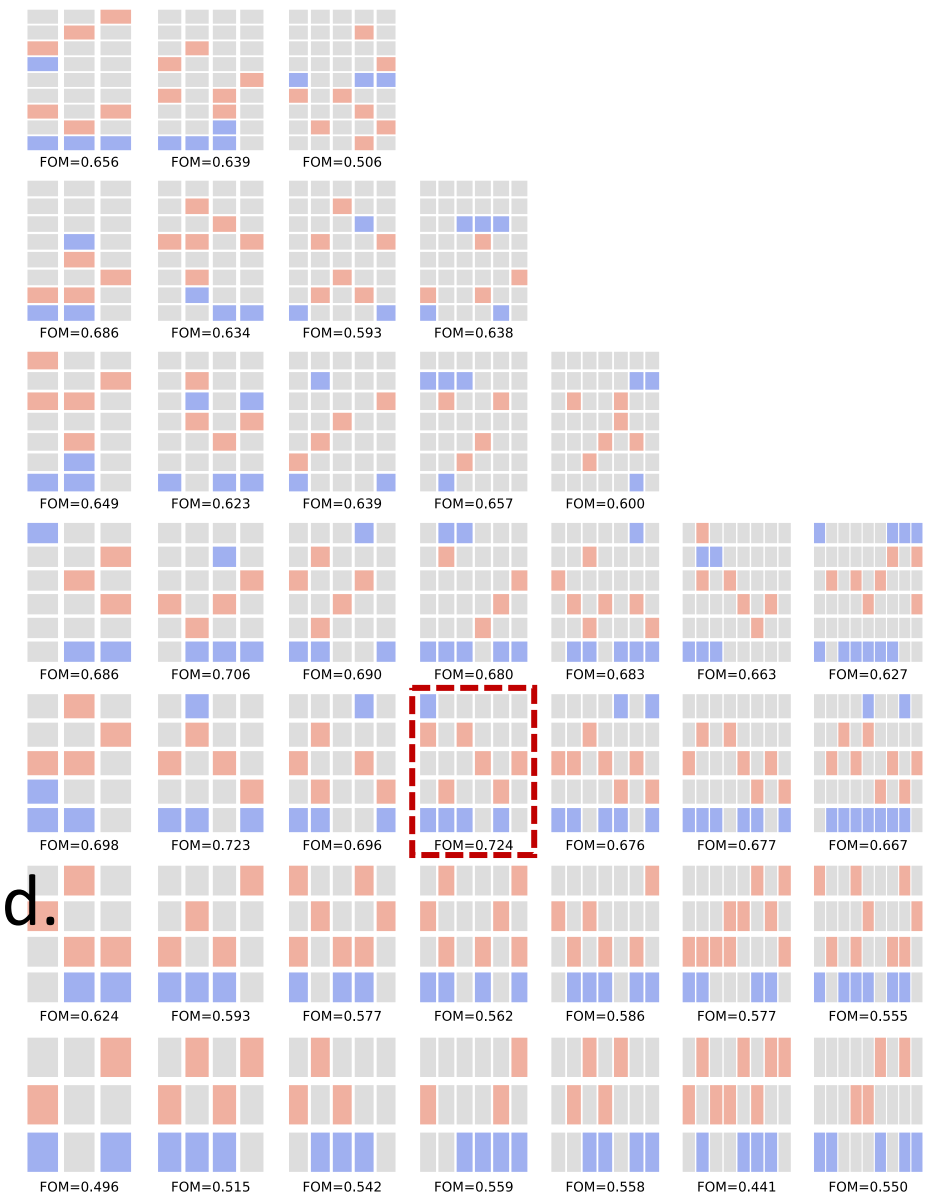


	FMQA	Random
Reg.	FM	None
Sel.	QA	Random

Designing larger structure

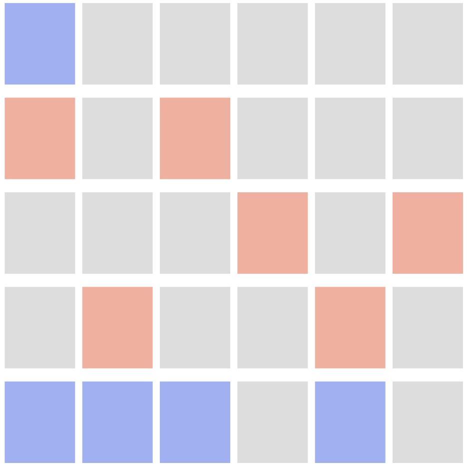
- On varied size problem
  - the number of layers L=3,4,5,6,7,8,9
  - the number of columns C=3,4,5,6,7,8,9
  - up to 60 bits for encoding
  - 2000 times of selection on all settings
    - not enough for large problems

- Better structure than in literatures is found.
  - FOM = 0.724

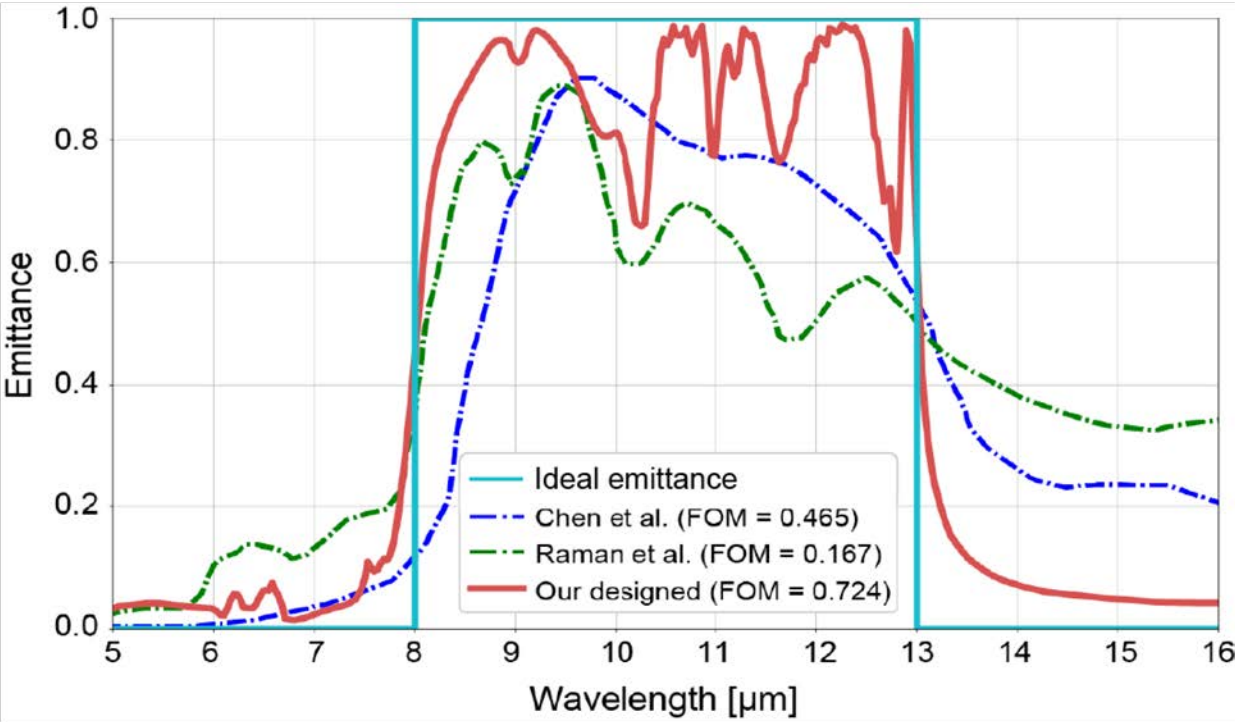


Designing larger structure - result

- The best structure found showed the best concordance with the window function.



FOM=0.724



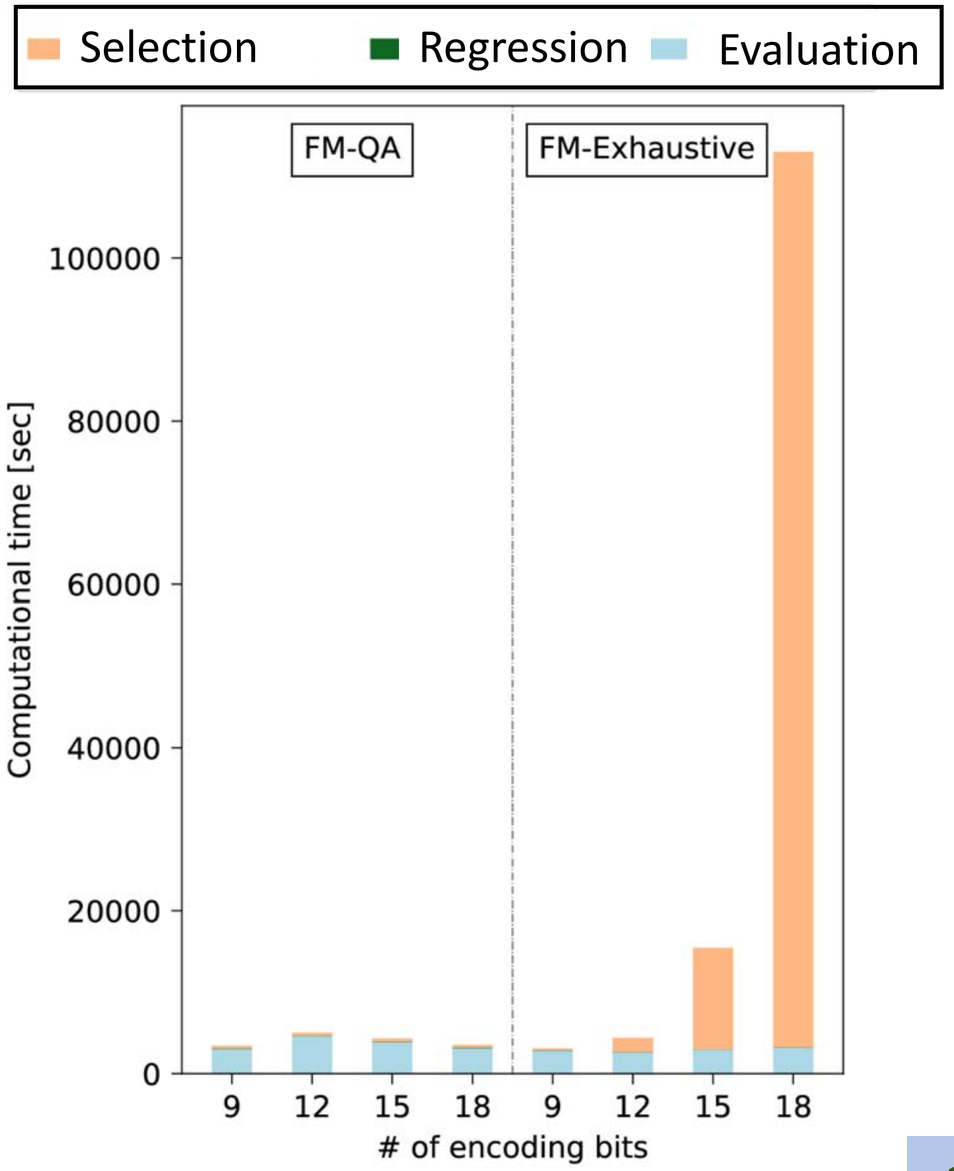
Profiling of running time

A profile of running time for various problem sizes.

- Evaluation - RCWA
- Regression - FM
- Selection - QA or Exhaustive

    finding the next structure to try

With the exhaustive search, time for selection was dominant, while in our method it was reduced to constant.



Summary & Conclusion

- We proposed a new method for black-box optimization to tackle any binary combinatorial optimization.
- FMQA is competitive with BO on small size problems, and even works fine on larger problems.
- We have shown an example of application.
  - automated materials discovery

- Now the bottleneck part is the evaluation part.

Contributors

Waseda Univ

Shu Tanaka

Tsuda Lab, UTokyo

Koji Tsuda

Ryo Tamura

Dept of Mech Eng, UTokyo

Junichiro Shiomi

Shenghong Ju

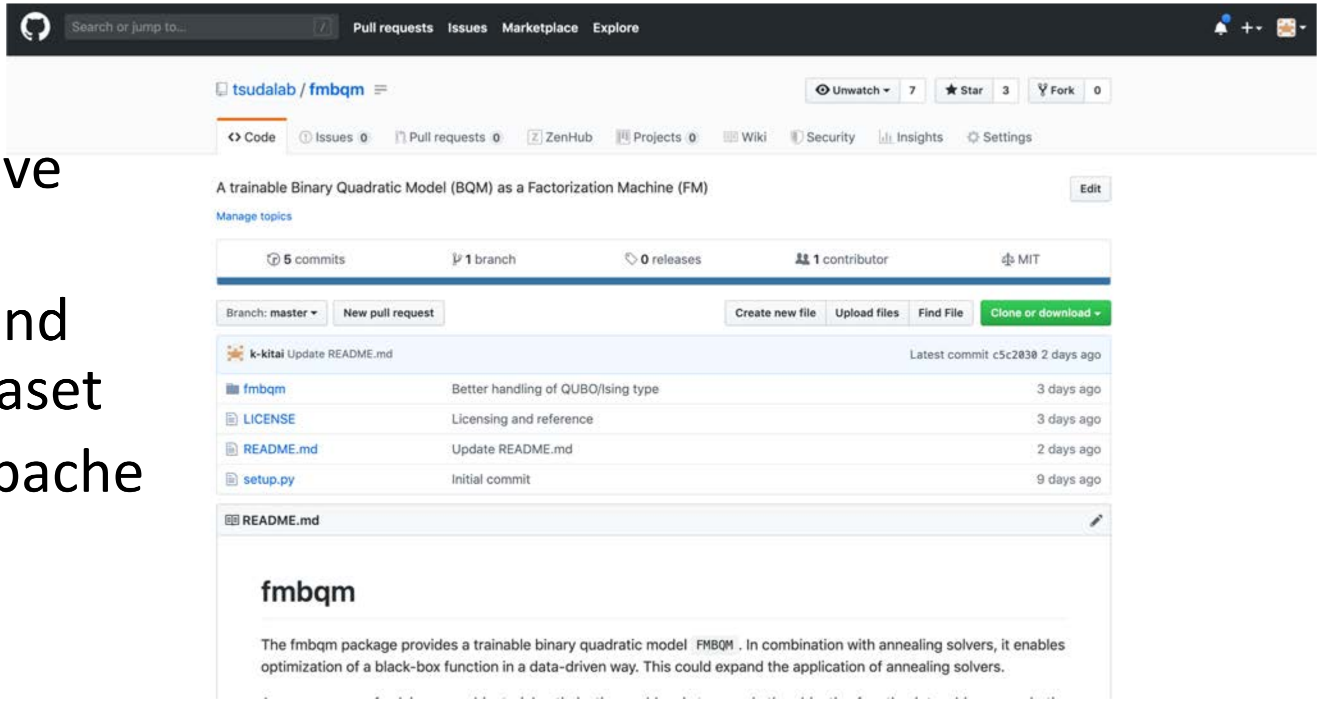
Jiang Guo

Agenda

1. Algorithm of black-box optimization
2. Application on metamaterial design
3. Introduction to fmbqm

fmbqm – An extension of BQM

- fmbqm
  - Based on BQM class from D-Wave Ocean SDK
  - FM model is contained inside, and the parameter is trained on dataset
  - FM part is implemented with Apache MXNet



- Pre-release
  - https://github.com/tsudalab/fmbqm

Demo

- The target function
  - Binary encoding of Integer
  - The first bit represents sign

```
[0,0,0,1] => 1
[0,0,1,0] => 2
[0,1,0,0] => 4
[1,0,0,1] => -1
[1,0,1,0] => -2
[1,1,0,0] => -4
```

  - Scaling to range [-1,1]
  - Strong correlation between sign bit and magnitude bits

```
def bin2int(x, scaling=True):
    '''
    Evaluation function for a binary array
    to a signed integer
    '''
    val, n = 0, len(x)
    for i in range(1,n):
        val = (val << 1) + x[i]
    if x[0] == 1:
        val = -val
    return val * (2**(1-n) if scaling else 1)
```

Demo

- Generate initial dataset
- Train the model based on it

16 bits length
5 data points

Easy to train

```
import numpy as np
from fmbqm import FMBQM

xs = np.random.randint(2, size=(5,16))
ys = np.array([bin2int(x) for x in xs])

model = FMBQM.from_data(xs, ys)
```

Demo

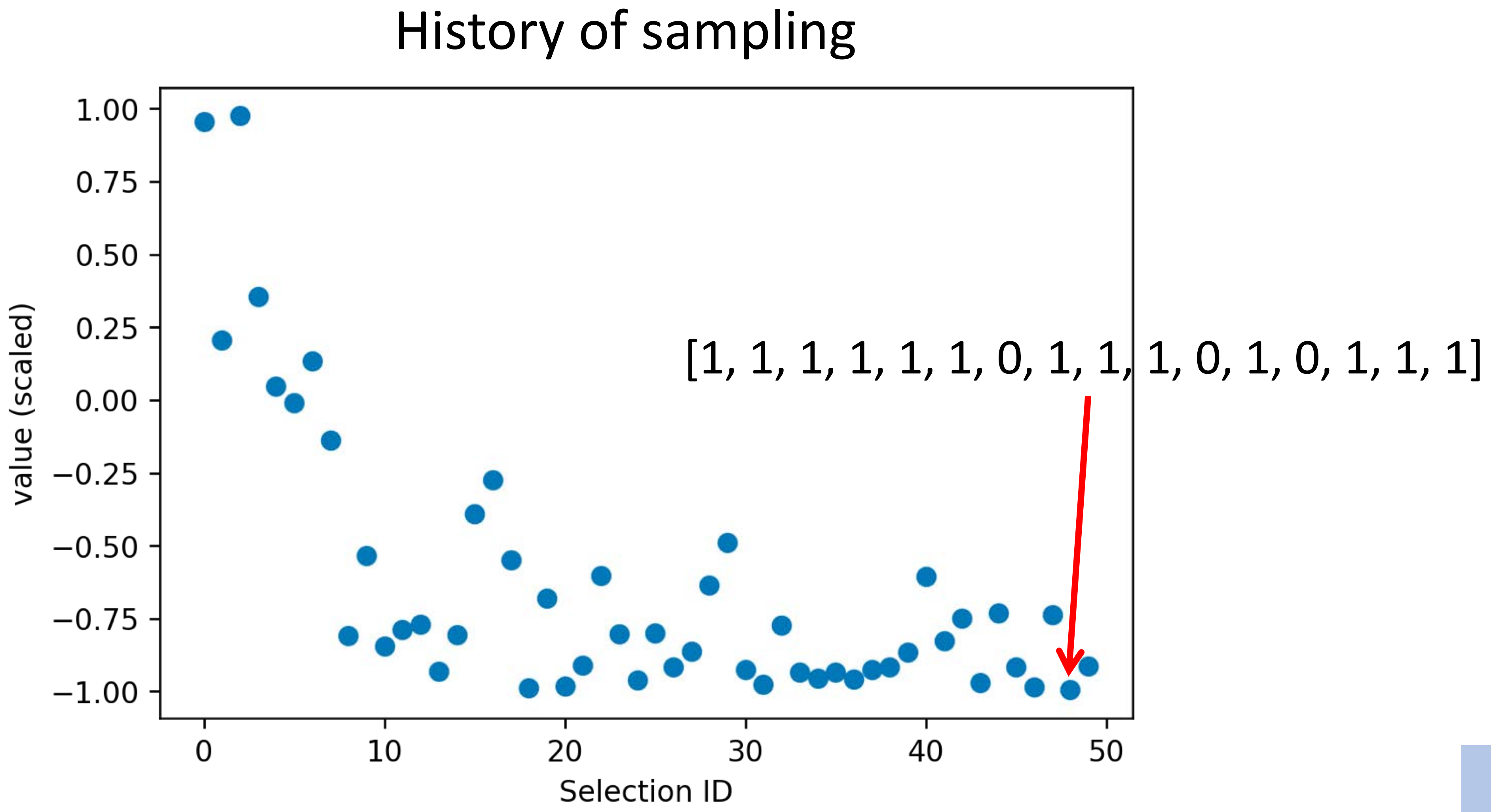
- Repeat sampling and retraining of the model several times

Sampling as easy as original BQM →

Easy to update →

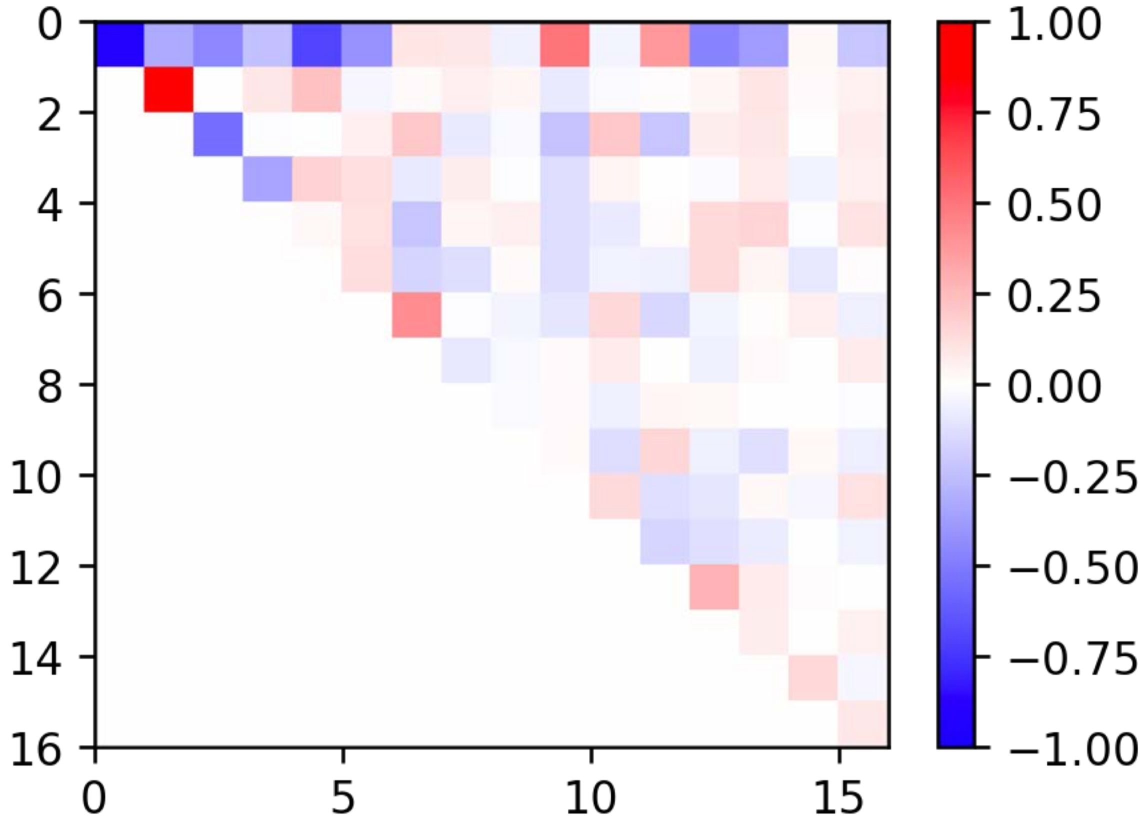
```
import dimod
sampler = dimod.SimulatedAnnealingSampler()

for _ in range(15):
    res = sampler.sample(model, num_reads=3)
    new_xs = res.record['sample']
    xs = np.r_[xs, new_xs]
    ys = np.r_[ys, [bin2int(x) for x in new_xs]]
    model.train(xs, ys)
```

Demo


History of sampling
value (scaled)
Selection ID
1.00
0.75
0.50
0.25
0.00
−0.25
−0.50
−0.75
−1.00
0
10
20
30
40
50
[1, 1, 1, 1, 1, 1, 0, 1, 1, 1, 0, 1, 0, 1, 1, 1]

Demo

- The reconstructed QUBO parameter
- Strong correlation between sign bit and magnitude bits are retrieved.
- Upper bits are strongly forced to be [1,1,1,1...].



Thank you for listening.

arXiv: 1902.06573
https://github.com/tsudalab/fmbqm