



2026.05.08 Quantum CAE研究会@産業技術総合研究所 臨海副都心センター

# 準ランダム列に基づく初期学習データ 生成を用いたFMQAの性能評価 — 人力航空機翼形状最適化への適用 —

T. Hayashi et al., arXiv:2605.04825 (2026).

林泰雅(慶大) 関優也(慶大/SQAI) 寺田晃太郎(Quanmatic)

武笠陽介(慶大/Quanmatic) 菊池脩太(慶大/SQAI) 田中宗(慶大/SQAI/WPI-Bio2Q)

# 本発表の概要

## 研究目的



- 機械学習とイジングマシンを組み合わせたハイブリッド手法であるブラックボックス最適化アルゴリズム, Factorization machine with quadratic optimization annealing (FMQA) の性能改善

## 研究手法



- 初期学習データにおいて, 1の値をとるビットの割合に着目し, すべてのビットに一度, 1の値をとるような準ランダム列に基づいて初期学習データを準備する拡張FMQAを提案

## 研究結果



- 提案手法は同一BB関数評価回数において最も良い目的関数値を探索することができた
- 提案手法は目標値に対して少ないサイクル数で到達することができた
- 提案手法により, FMのBlack-box関数に対する推定精度が向上することを確認することができた

# 研究背景 | イジングマシンによる最適化の手順

## イジングマシンによる解探索までの流れ



### 組合せ最適化問題とは

ある制約条件を満たしつつ、コストまたは価値を表す目的関数を最小化(あるいは最大化)するような離散変数の組合せを求める問題

$$x^* = \operatorname{argmin}_x f(x), \quad x \in F$$

$x$ : 離散変数,  $x^*$ : 最適解,  $f(x)$ : 目的関数,  $F$ : 制約条件

例:

巡回セールスマン問題 (Traveling Salesman Problem, TSP)



### QUBO定式化について

組合せ最適化問題をイジングモデルやQUBO\*で定式化し、最適解を短時間かつ高精度に探索することができる

\*QUBO : Quadratic Unconstrained Binary Optimization

$$E_{QUBO}(x) = \sum_{1 \leq i < j \leq N} Q_{i,j} x_i x_j \quad x_i \in \{0,1\}$$

$N$ : 入力  $x$  の次元数,  $Q_{i,j}$ : パラメータ

# 研究背景 | イジングマシンによる最適化の手順

## イジングマシンによる解探索までの流れ



しかし、組合せ最適化問題がいつでも、QUBOに定式化できるとは限らない

- 問題が複雑すぎて定式化困難な場合
- 問題に高次の相互作用を含む場合

 **ブラックボックス最適化問題**

### QUBO定式化について

組合せ最適化問題をイジングモデルやQUBO\*で定式化し、最適解を短時間かつ高精度に探索することができる

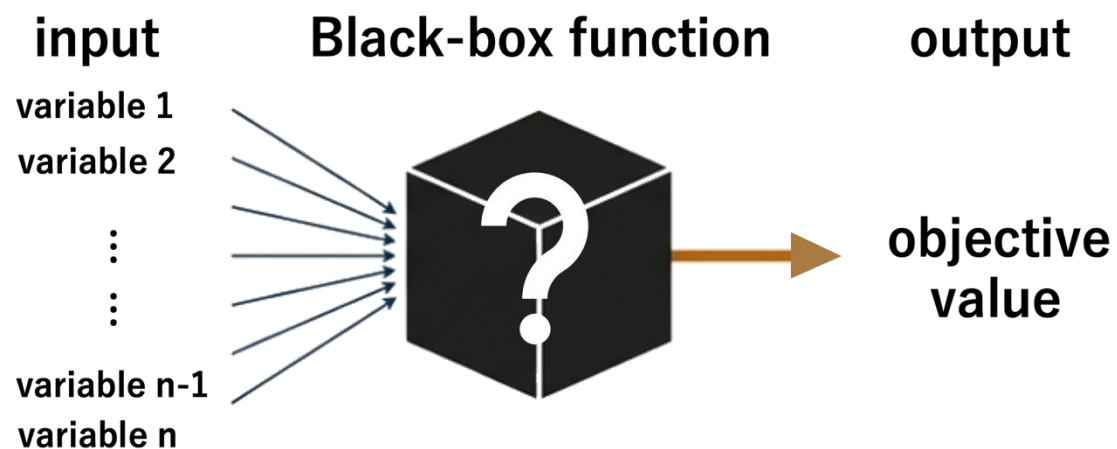
\*QUBO : Quadratic Unconstrained Binary Optimization

$$E_{QUBO}(x) = \sum_{1 \leq i < j \leq N} Q_{i,j} x_i x_j \quad x_i \in \{0,1\}$$

$N$ : 入力  $x$  の次元数,  $Q_{i,j}$ : パラメータ

# 研究背景 | ブラックボックス最適化 (Black-box optimization, BBO)

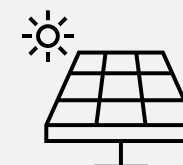
ブラックボックス最適化: 入力データとそれから得られる出力のみを用いて出力を最小化(最大化)する入力となる決定変数の組合せを探索する問題



BBOは工学的な場面で広く現れる最適化問題



形状最適化



材料設計



創薬

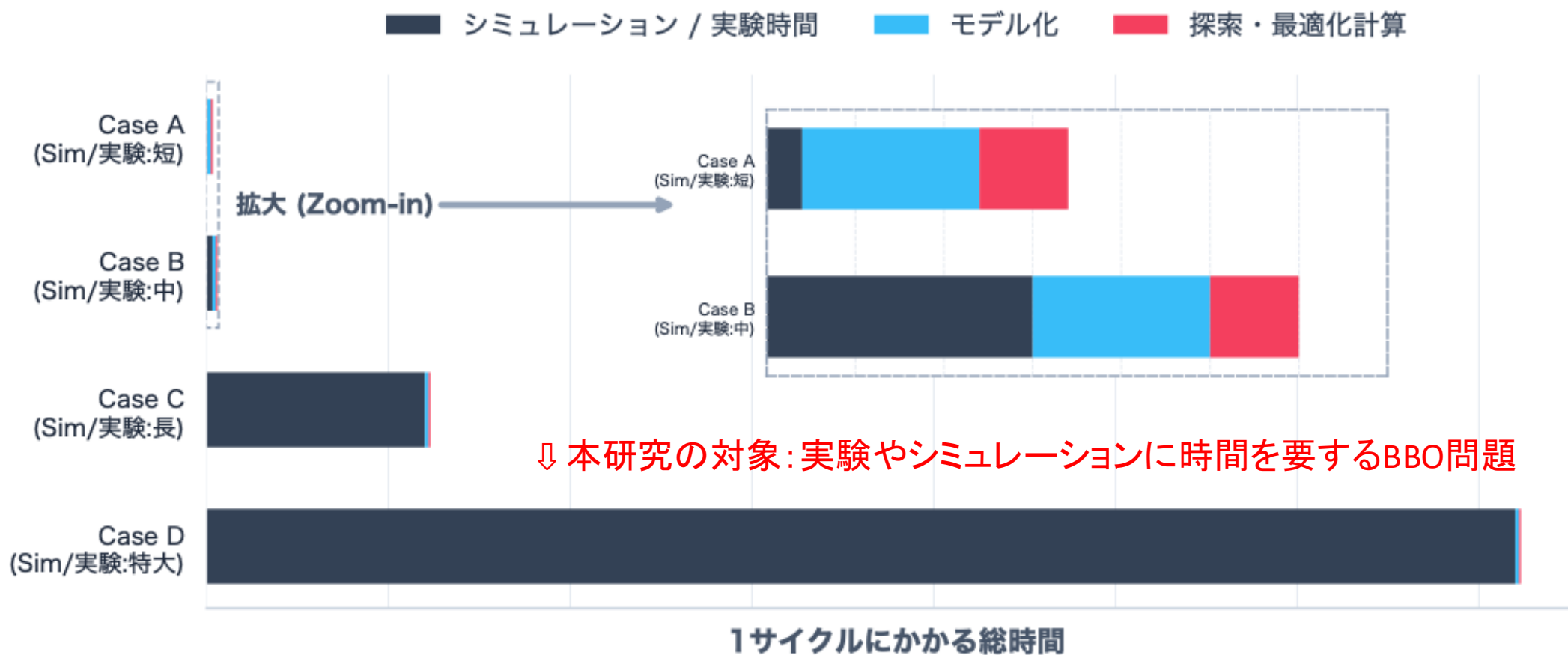
入力と出力のみで出力を最適化する

このようなBBOでは1回のBB関数を評価するために非常に多くの時間や金額がかかる

# 研究背景 | ブラックボックス最適化 (Black-box optimization, BBO)

工学的な観点におけるBBOの要望

前提: 1回のBB関数を評価するためには「時間」・「金額」のコストがかかる



# 研究背景 | ブラックボックス最適化 (Black-box optimization, BBO)

1回のBB関数の評価に時間や金額を要する問題に対して、評価回数を削減するために、様々なBBOアプローチが提案されている

## 確率的探索手法

生物の遺伝と進化のメカニズムを計算機上で模倣して、適応や学習、最適化などの機能を再現する手法

## 代理モデルベース最適化手法

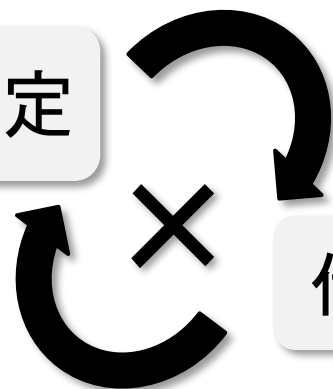
観測済みデータから目的関数を近似する代理モデルを構築し、その予測に基づいて次に評価すべき候補解を選ぶ手法

# 研究背景 | 代理モデルを用いたブラックボックス最適化

代理モデル(サロゲートモデル)を用いたBBOに対するアプローチ

観測データに基づいて, 代理モデルの構築と, 候補点探索を逐次的に繰り返す

代理モデルを用いてBB関数を推定



代理モデルから候補解を定める

# 研究背景 | 代理モデルを用いたブラックボックス最適化

## 代理モデル(サロゲートモデル)を用いたBBOに対するアプローチ

観測データに基づいて、代理モデルの構築と、候補点探索を逐次的に繰り返す

代理モデルを用いてBB関数を推定

ベイズ推論

機械学習モデル

代理モデルから候補解を定める

ベイズ推論

イジングマシン

# 研究背景 | 代理モデルを用いたブラックボックス最適化

## 代理モデルを用いたBBOに対するアプローチ

- [1] R. Baptista and M. Poloczek, arXiv:1806.08838 (2018).
- [2] K. Kitai *et al.*, *Phys. Rev. Research* **2**, 013319 (2020).
- [3] R. Tamura *et al.*, *Applied Physics Reviews* **13**, 021307 (2026).

### Bayesian Optimization

ガウス過程回帰を代理モデル  
観測データで事後分布を更新  
現在の不確かさを踏まえて、次の探索点を決定

計算量は $O(n^3 + n^2d)$

ベイズ推論

### BOCS<sup>[1]</sup>

Bayesian optimization of combinatorial structures

QUBO形式でBB関数を推定  
代理モデルの係数を事後分布からサンプル  
イジングマシン等により次の探索点を決定

計算量は $O(n^2M^2)$

ベイズ推論 × イジングマシン

### FMQA<sup>[2,3]</sup>

Factorization machine  
with quadratic optimization annealing

機械学習モデル Factorization machine  
(FM)でBB関数を推定  
イジングマシンにより次の探索点を決定

計算量は $O(KMn)$

機械学習モデル × イジングマシン

$n$ : データ数  
 $M$ : ビット数  
 $d$ : 次元数

# 研究背景 | 代理モデルを用いたブラックボックス最適化

## 代理モデルを用いたBBOに対するアプローチ

- [1] R. Baptista and M. Poloczek, arXiv:1806.08838 (2018).
- [2] K. Kitai *et al.*, *Phys. Rev. Research* 2, 013319 (2020).
- [3] R. Tamura *et al.*, *Applied Physics Reviews* 13, 021307 (2026).

### Bayesian Optimization

ガウス過程回帰を代理モデル  
観測データで事後分布を更新  
現在の不確かさを踏まえて、次の探索点を決定

計算量は $O(n^3 + n^2d)$

ベイズ推論

### BOCS<sup>[1]</sup>

Bayesian optimization of combinatorial structures

QUBO形式でBB関数を推定  
代理モデルの係数を事後分布からサンプリングマシン等により次の探索点を決定

計算量は $O(n^2M^2)$

ベイズ推論 × イジングマシン

### 本研究の対象

### FMQA<sup>[2,3]</sup>

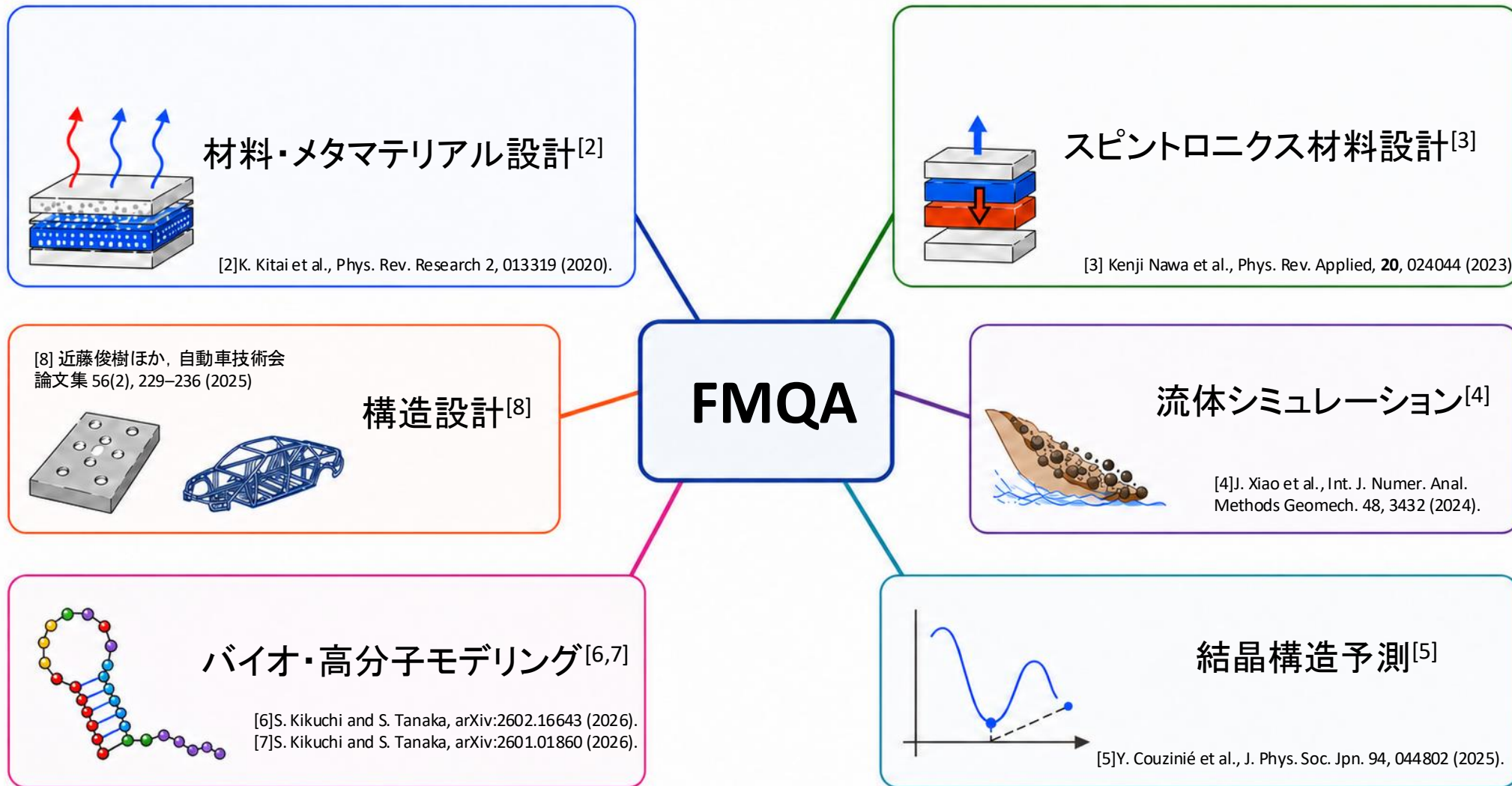
Factorization machine  
with quadratic optimization annealing  
機械学習モデル Factorization machine (FM)でBB関数を推定  
イジングマシンにより次の探索点を決定

計算量は $O(KMn)$

機械学習モデル × イジングマシン

$n$ : データ数  
 $M$ : ビット数  
 $d$ : 次元数

# 研究背景 | 機械学習とイジングマシンをハイブリッドしたBBOアルゴリズム FMQA



# 研究手法 | Factorization machine with quadratic optimization annealing (FMQA)

## 機械学習モデル Factorization machine (FM)とQUBOの等価性

イジングマシンで解探索を行う際には、QUBO形式に定式化する必要

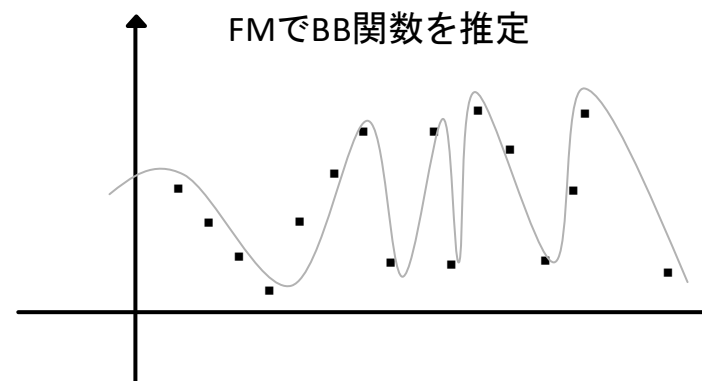
FMはQUBO形式と等価  $\Rightarrow$  FMのモデルパラメータとQUBO係数が対応

[9] S.Rendle,Steffen.2010 IEEE International conference on data mining.IEEE,(2010).

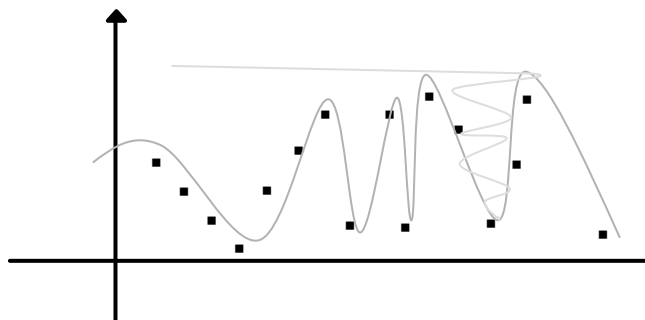
## 機械学習モデル Factorization machine (FM)<sup>[9]</sup>

$$f_{FM}(x; \theta) = \omega_0 + \sum_{i=1}^N \omega_i x_i + \sum_{1 \leq i < j \leq N} \langle v_i, v_j \rangle x_i x_j$$

モデルパラメータ:  $\theta \equiv \omega_0 \in \mathbb{R}, \omega_i \in \mathbb{R}, v_i \in \mathbb{R}^K \quad x = (x_1, \dots, x_N) \in \{0, 1\}^N$



イジングマシンによりFMの最適解を探索



FMのモデルパラメータからQUBOの係数を直接変換可能

## QUBO形式

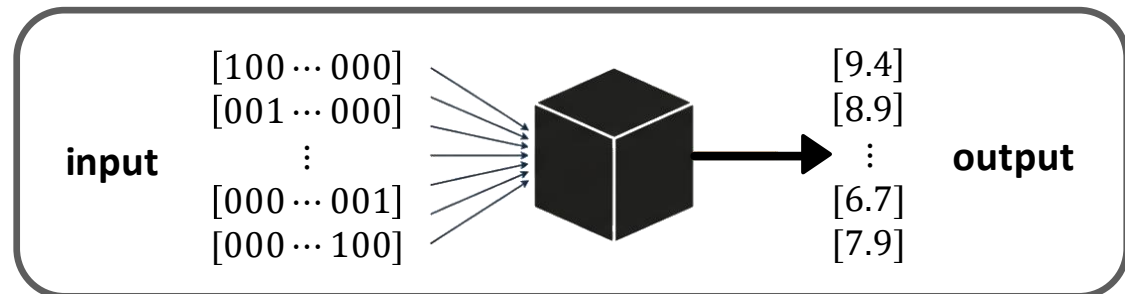
$$E_{QUBO}(x) = \sum_{1 \leq i < j \leq N} Q_{i,j} x_i x_j$$

$x = (x_1, \dots, x_N) \in \{0, 1\}^N$

$$Q_{N,N} = \begin{pmatrix} Q_{1,1} & \cdots & Q_{i,1} & \cdots & Q_{N,1} \\ \vdots & \ddots & \vdots & \cdots & \vdots \\ Q_{1,j} & \cdots & Q_{i,j} & \cdots & Q_{N,j} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ Q_{1,N} & \cdots & Q_{i,N} & \cdots & Q_{NN} \end{pmatrix}$$

# 研究手法 | Factorization machine with quadratic optimization annealing (FMQA)

## 0. 初期学習データの準備



## 1. FMの構築

$$f_{\text{FM}}(x; \theta) = \omega_0 + \sum_{i=1}^N \omega_i x_i + \sum_{1 \leq i < j \leq N} \langle v_i, v_j \rangle x_i x_j$$

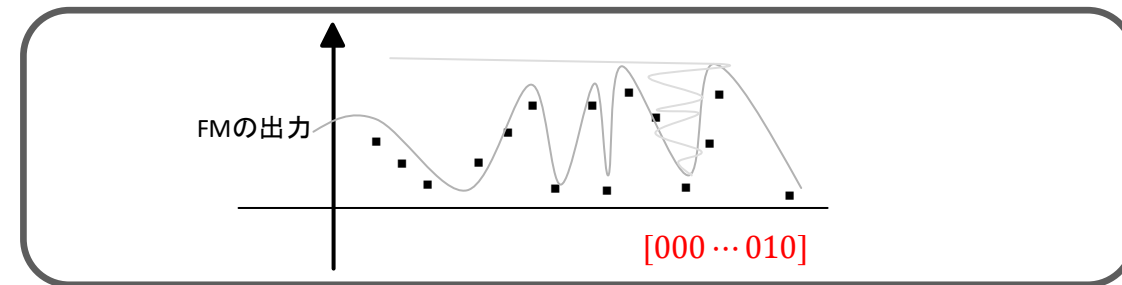
モデルパラメータ:  $\theta \equiv \omega_0 \in \mathbb{R}, \omega_i \in \mathbb{R}, v_i \in \mathbb{R}^K \quad x = (x_1, \dots, x_N) \in \{0, 1\}^N$

## 4. データセットの更新

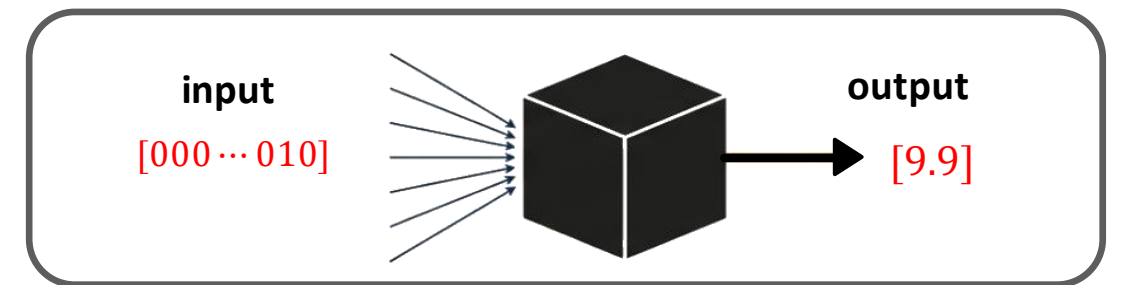
|             |        |   |             |        |
|-------------|--------|---|-------------|--------|
| input       | output |   | input       | output |
| [100...000] | [9.4]  |   | [100...000] | [9.4]  |
| [001...000] | [8.9]  |   | [001...000] | [8.9]  |
| ⋮           |        |   | ⋮           |        |
| [000...001] | [6.7]  | + | [000...001] | [9.9]  |
| [000...100] | [7.9]  | = | [000...100] | [6.7]  |
|             |        |   | [000...010] | [9.9]  |

# FMQA最適化の流れ

## 2. イジングマシンによる解探索



## 3. BB関数の出力を得る



## 整数・バイナリ変換

連続変数を有するBBOをFMQAに適用する際には、整数・バイナリ変換を用いて連続変数をバイナリ変数へ離散化  
先行研究ではFMQAにおいては、one-hot表現が他の手法に対して優位性<sup>[10]</sup>

## one-hot表現

ある変数に対応するビット列において、1つのビットのみが1の値をとり、それ以外のビットが0の値をとるバイナリ表現方法

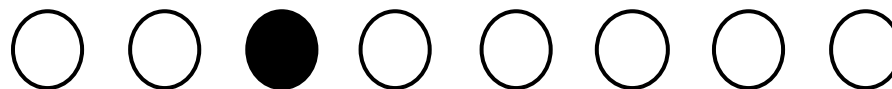
$$\sum_{m=0}^{M-1} x_m = 1$$

$$x \in \{0, 1\}$$

$m$ : bit index

$M$ : 用いたビットの総数

2を8つのビットで表現する場合、2に対応するビットのみが1の値をとる。



0 1 2 3 4 5 6 7

# 研究手法 | One-hot表現におけるFMパラメータ更新における課題

[11]K. Endo et al, Phys. Rev. Research 7, 013149 (2025).

## FMパラメータ更新方法とその課題

整数・バイナリ変換としてone-hot表現を用いた場合、  
データセット内で、1の値をとったビットに対応するパラメータのみをデータセットに基づいて更新

本研究では、FMのパラメータ更新に用いる損失関数を、FMの予測値とBB関数の出力値との平均二乗誤差として定義

$$L(\theta) = \frac{1}{D} \sum_{d=1}^D (f_{\text{FM}}(x_d; \theta) - f_{\text{BB}}(\phi(x_d)))^2$$

$D$ : データ数

$f_{\text{BB}}(\phi(x_d))$ : BB関数の出力値

$f_{\text{FM}}(x_d; \theta)$ : FMの出力値

出力差を最小化するため、モデルパラメータを更新

**×** データセット内で  $x_i = 0$  であるときは、勾配  
データセットに基づいたFMのパラメータの更新

$$\frac{L(\theta)}{\partial \theta} = \frac{2}{D} \sum_{d=1}^D (f_{\text{FM}}(x_d; \theta) - f_{\text{BB}}(\phi(x_d))) \frac{\partial f_{\text{FM}}(x_d; \theta)}{\partial \theta} = 0$$

$$\frac{\partial f_{\text{FM}}(x_d; \theta)}{\partial \theta} = \begin{cases} 1 & (\theta = \omega_0) \\ x_i & (\theta = \omega_i) \\ \left( \sum_{j=1}^N v_{jl} x_j \right) x_i - v_{il} x_i^2 & (\theta = v_{il}) \end{cases}$$

# 研究手法 | One-hot表現におけるFMパラメータ更新における課題

[11]K. Endo et al, Phys. Rev. Research 7, 013149 (2025).

## FMパラメータ更新方法とその課題

整数・バイナリ変換としてone-hot表現を用いた場合、  
データセット内で、1の値をとったビットに対応するパラメータのみをデータセットに基づいて更新

### データセットの例

|         | input |   |   |   | output |   |   |   |      |
|---------|-------|---|---|---|--------|---|---|---|------|
|         | 変数: 1 |   |   |   | 変数: 2  |   |   |   |      |
| データ1    | ●     | ○ | ○ | ○ | ○      | ● | ○ | ○ | 10.4 |
| データ2    | ○     | ○ | ● | ○ | ○      | ● | ○ | ○ | 10.9 |
| データ3    | ●     | ○ | ○ | ○ | ●      | ○ | ○ | ○ | 12.4 |
| データ4    | ○     | ○ | ○ | ● | ○      | ○ | ● | ○ | 11.3 |
| データ5    | ○     | ○ | ● | ○ | ○      | ● | ○ | ○ | 8.8  |
| データ6    | ○     | ○ | ○ | ● | ●      | ○ | ○ | ○ | 13.5 |
| データ7    | ●     | ○ | ○ | ○ | ○      | ○ | ● | ○ | 8.9  |
| 1が立った回数 | 3     | 0 | 2 | 2 | 2      | 3 | 2 | 0 |      |

1の値をとらないビットに対応するQUBOの係数がデータセットに基づいて更新されない

$$Q_{N,N} = \begin{pmatrix} Q_{1,1} & \dots & Q_{i,1} & \dots & Q_{N,1} \\ \vdots & \ddots & \vdots & \dots & \vdots \\ Q_{1,j} & Q_{2,j} & Q_{i,j} & Q_{i+3,j} & Q_{N,j} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ Q_{1,N} & \dots & Q_{i,N} & \dots & Q_{N,N} \end{pmatrix}$$

⇒ FMの推定精度の悪化を招く

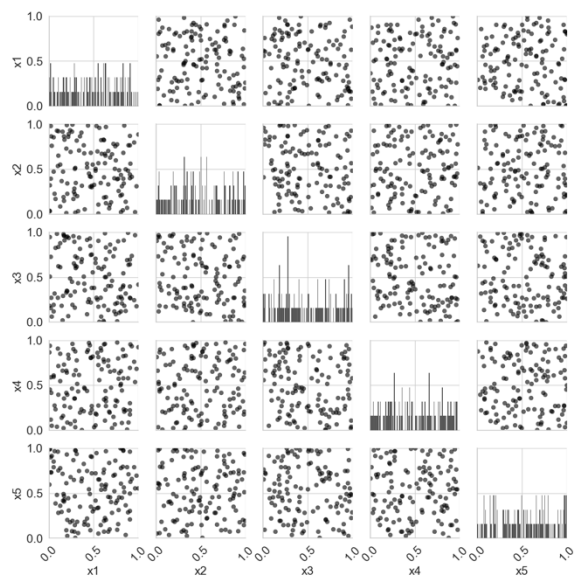
すなわち、FMの推定精度の悪化がイジングマシンにより得られる解に影響を及ぼす

# 研究手法 | One-hot表現におけるFMパラメータ更新における課題

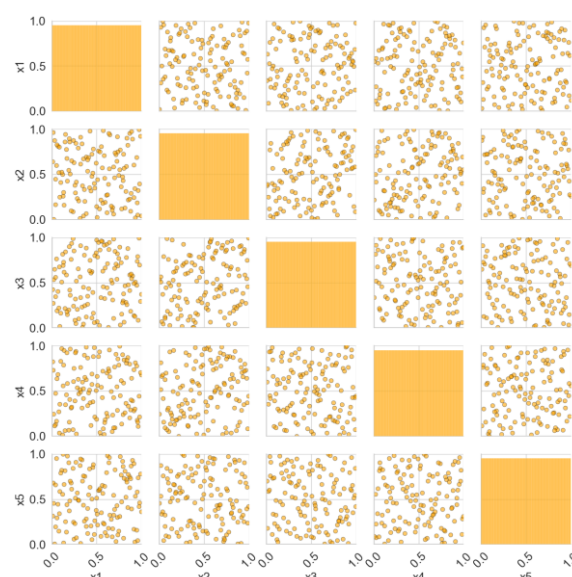
提案手法: 準ランダム列により初期学習データを準備する拡張FMQA

初期学習データセットを, すべてのビットに一度, 1の値をとることのできる (complete marginal bit coverage) 準ランダム列に基づいて生成. 本研究では, Latin hypercube sampling (LHS)<sup>[12,13]</sup> および Sobol' sequences<sup>[14,15]</sup> で生成

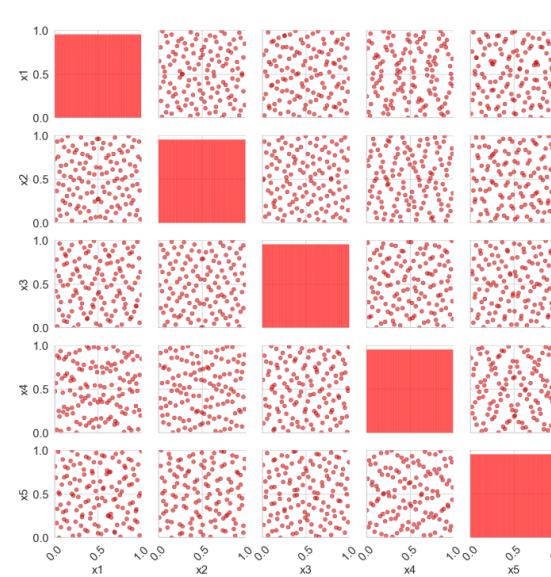
## サンプリング手法ごとのペアプロット(次元数=5)



uniform random



LHS



Sobol' sequences

- [12] M. D. McKay, R. J. Beckman, and W. J. Conover: *Technometrics* 21 (1979) 239.
- [13] D. Bingham et al., *Handbook of Design and Analysis of Experiments*, CRC Press, 2015.
- [14] I. M. Sobol', *USSR Comput. Math. Math. Phys.* 7, 86-112 (1967).
- [15] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*, SIAM, 1992.

LHSやSobol' は各グリッドに等しい数のサンプルを生成することができる

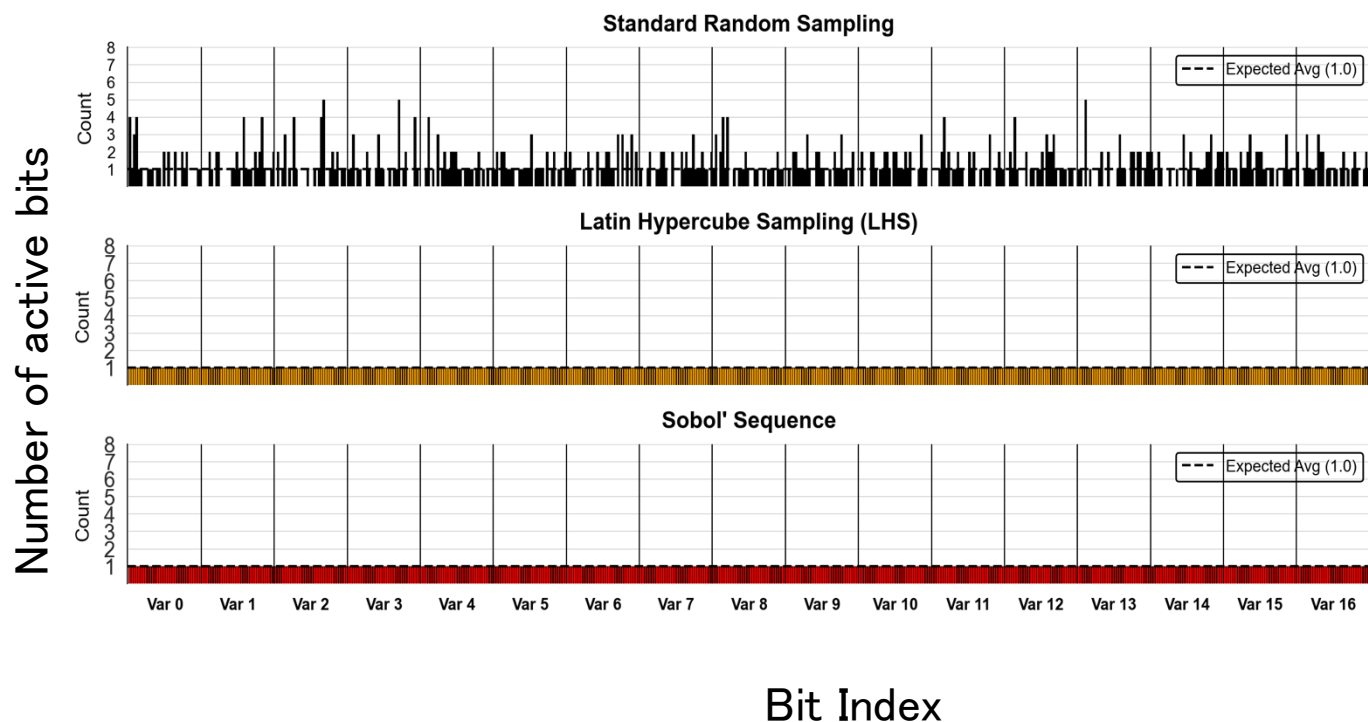
- 各変数間のサンプルを2次元平面で可視化
- 対角項は各平面をグリッドに切ったときにそのグリッドの中に幾つサンプルが存在するかを可視化

# 研究手法 | One-hot表現におけるFMパラメータ更新における課題

提案手法: 準ランダム列により初期学習データを準備する拡張FMQA

初期学習データセットを, すべてのビットに一度, 1の値をとることのできる (complete marginal bit coverage) 準ランダム列に基づいて生成. 本研究では, Latin hypercube sampling (LHS)<sup>[12,13]</sup> および Sobol' sequences<sup>[14,15]</sup> で生成

## 本研究で準備した初期学習データにおいて1の値をとるビットの分布



- [12] M. D. McKay, R. J. Beckman, and W. J. Conover: Technometrics 21 (1979) 239.
- [13] D. Bingham et al., *Handbook of Design and Analysis of Experiments*, CRC Press, 2015.
- [14] I. M. Sobol', USSR Comput. Math. Math. Phys. 7, 86-112 (1967).
- [15] H. Niederreiter, *Random Number Generation and Quasi-Monte Carlo Methods*, SIAM, 1992.

初期学習データで, 1の値をとらないビットの期待個数

$$n_x M \left(1 - \frac{1}{M}\right)^{N_0} \approx 198$$

各変数においてあるビットが1の値をとる確率

$$\left(1 - \frac{1}{M}\right)^{N_0} \approx 0.362 \%$$

変数数:  $n_x$   
 分割数:  $M$   
 初期学習データ数:  $N_0$

# 研究手法 | Latin hypercube sampling (LHS)の生成方法について

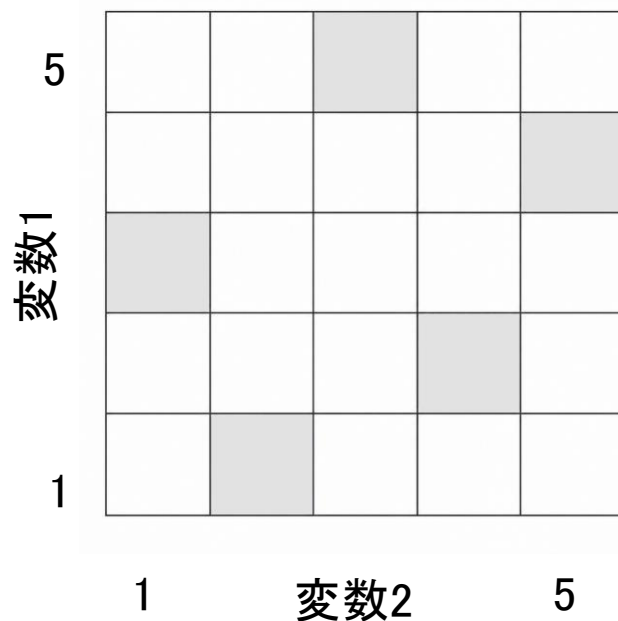
## ■ Latin Hypercube Sampling(LHS)<sup>[12,13]</sup>

- 実験計画法で、多次元空間を満遍なくサンプリングする手法
- 行と列の選び方にランダム性があるサンプリング手法
- $N_0$ (初期学習データ数) =  $M$ (分割数)という条件で、すべてのビットに1の値を取ることが可能

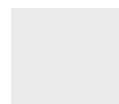
[12] M. D. McKay, R. J. Beckman, and W. J. Conover: Technometrics 21 (1979) 239.

[13] D. Bingham et al., *Handbook of Design and Analysis of Experiments*, CRC Press, 2015.

## ■ 例(2変数=2次元空間の場合)



このように  
行と列が一致しないように



をランダムに選ぶ



これが、縦、横軸が1を  
立てるbitに対応させる。

| 変数1          | 変数2          |
|--------------|--------------|
| 3(0 0 1 0 0) | 2(0 1 0 0 0) |
| 1(1 0 0 0 0) | 4(0 0 0 1 0) |
| 5(0 0 0 0 1) | 1(1 0 0 0 0) |
| 2(0 1 0 0 0) | 5(0 0 0 0 1) |
| 4(0 0 0 1 0) | 3(0 0 1 0 0) |

すべてのone-hotベクトルに0,1の両方が  
出現するようにサンプリングができる

# 研究手法 | Sobol' 列の生成方法について

[14] I. M. Sobol', USSR Comput. Math. Math. Phys. 7, 86-112 (1967).  
 [15] H. Niederreiter, Random Number Generation and Quasi-Monte Carlo Methods, SIAM, 1992.

## ■ Sobol' 列<sup>[14,15]</sup>とは

- 非負の整数  $k$  を  $b$  のべき乗で展開  $\Rightarrow b = 2$  として各項の係数を高次の項から順番に並べると、 $k$  の2進数表記となる
- その係数ベクトル  $\mathbf{a}(k)$  に、特別な方法で求めたGenerator Matrix (生成行列)  $V$  を作用させることで生成
- LHSとは異なり、決定論的生成手法
- $N_0$  (初期学習データ数) =  $M$  (分割数) =  $2^p$  という条件ですべてのビットに1の値をとることが可能

## ■ 生成方法について

1. 非負の整数  $k$  を  $b$  のべき乗で展開

$$k = \sum_{j=0}^{r-1} a_j(k) b^j = a_0(k) \times 2^0 + a_1(k) \times 2^1 + \dots + a_{r-1}(k) \times 2^{r-1}$$

右辺の係数ベクトルを  $\mathbf{a}(k)$  と置く  $\Rightarrow \mathbf{a}(k) = \{a_0(k), a_1(k), \dots, a_{r-1}(k)\}$

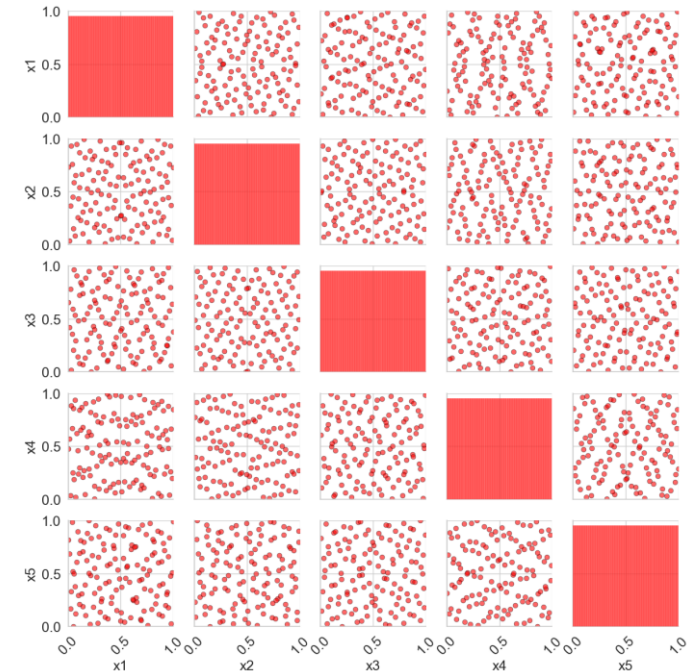
2.  $r \times r$  次Generator Matrix  $V^{(i)}$ ,  $i = 1, 2, \dots, d$  を  $d$  個用意し、 $\mathbf{a}(k)$  に作用させる。  
ただし、最後に  $\text{mod}2$  の操作を入れる

$$V^{(i)} \cdot \mathbf{a}(k) = \begin{pmatrix} v_{1,1} & \dots & v_{1,r} \\ \vdots & \ddots & \vdots \\ v_{r,1} & \dots & v_{r,r} \end{pmatrix} \cdot \begin{bmatrix} a_0(k) \\ \vdots \\ a_r(k) \end{bmatrix} \text{mod}2 = \begin{bmatrix} y_0^{(i)}(k) \\ \vdots \\ y_r^{(i)}(k) \end{bmatrix} = y^{(i)}(k), \quad i = 1, \dots, d$$

3. このようにして、生成された  $d$  個のベクトル  $y^{(i)}(k)$ ,  $i = 1, \dots, d$  を  $[0,1)$  の区間の小数点に変換する関数  $\psi_2(k)$  に代入

$$\psi_2(y^{(i)}(k)) = \sum_{j=1}^r \frac{y_j^{(i)}(k)}{2^j}, \quad i = 1, \dots, d$$

以上、 $k = 0, 1, 2, \dots, 2^r$  まで、繰り返すと、 $2^r$  個の  $d$  次元ベクトルとなる Sobol' 列が生成される

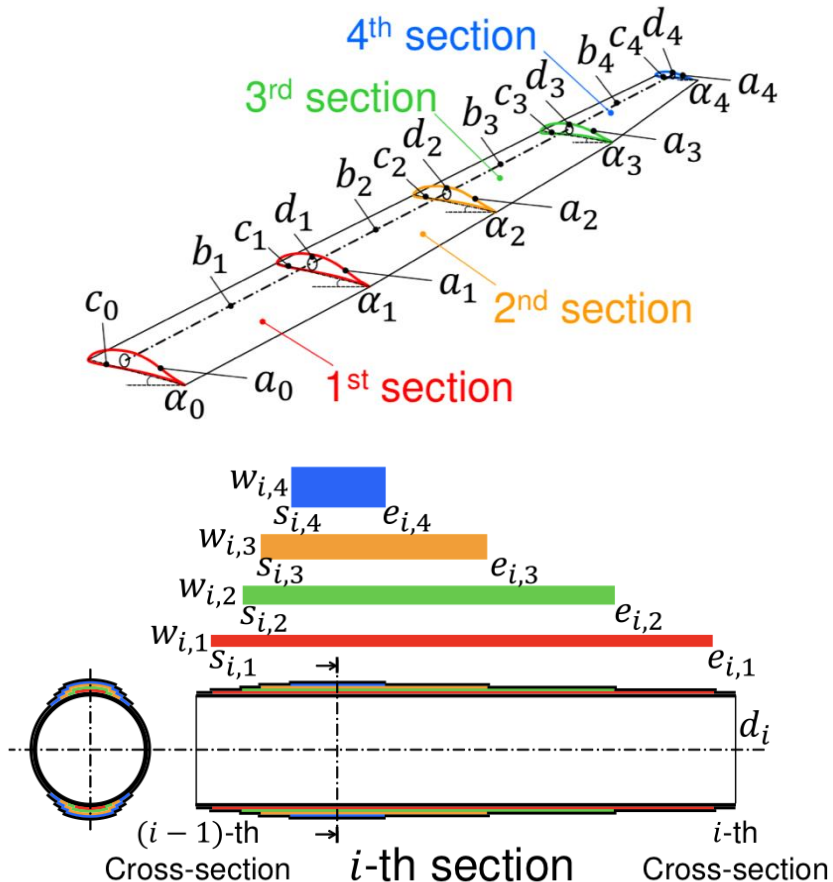


# 研究手法 | 最適化対象のBB関数

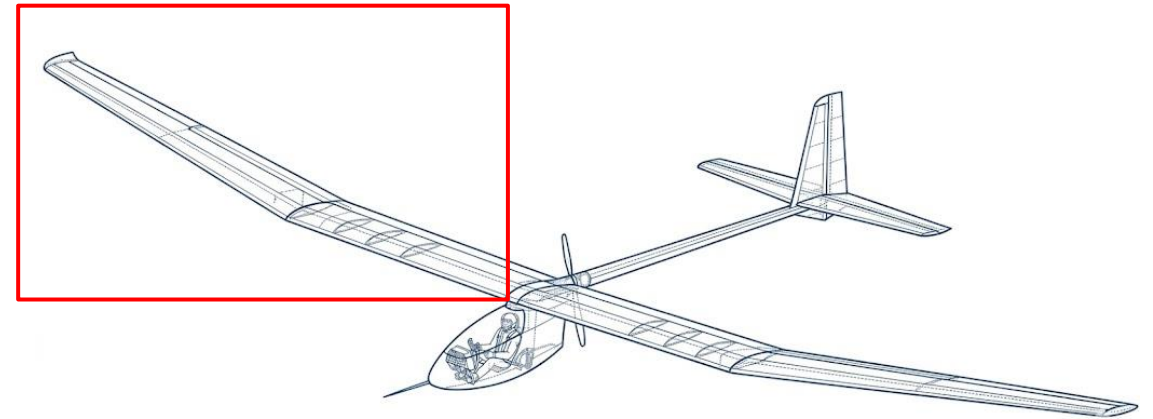
[16]N. Namura, in EMO 2025, LNCS 15512 (Springer, 2025).

## BB関数: Human-Powered Aircraft (HPA)<sup>[16]</sup>

人力航空機の翼(右翼)の最適化ベンチマーク  
単一目的最適化ベンチマークは3種類収録



最適化対象

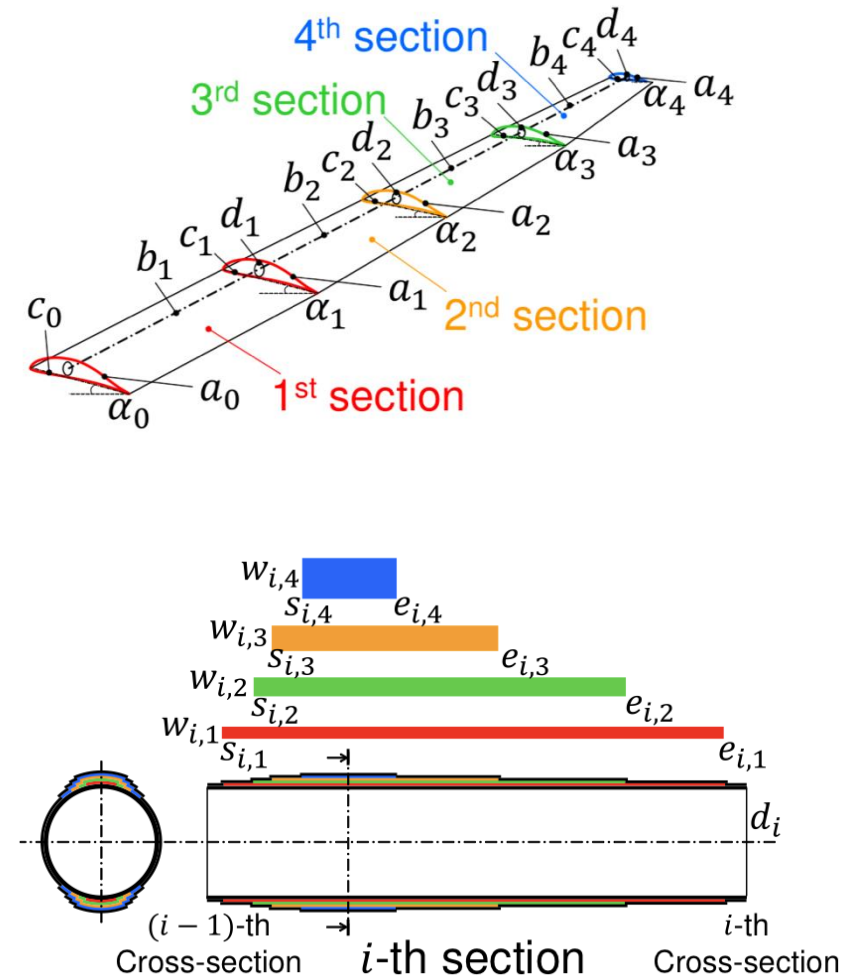


主翼の形状は翼を横方向に $n$ 個のセグメントに分割し、それぞれの断面における設計変数を指定することで定義

## BB関数: Human-Powered Aircraft (HPA)<sup>[16]</sup>

### 各設計変数の定義

- 翼の平面形状・断面形状に関する変数**
  - $a_i$ : 翼の断面形状 (DAEシリーズ)
  - $b_i$ : セグメント長 (各区画の横方向の長さ)
  - $c_i$ : 翼の前縁から後縁までの長さ (各区間の縦方向の長さ)
- 翼のねじり・取り付け角に関する変数**
  - $\omega_0$ : 翼根の迎え角 (翼の根元の部分における, 気流に対する取り付け角度)
  - $w_i$ : ねじり下げ角 (各セグメントで翼がどれくらい捻れているかを示す角度)
- 翼の反り上がりに関する変数**
  - $\varepsilon_0$ : 翼根の上反角 (翼を正面を見た時にV字型にどれくらい反り上がっているかを定める角度)
- 内部構造 (骨格) に関する変数**
  - $d_i$ : CFRPパイプの骨組み (主桁) となるカーボンパイプの太さ
  - $s_{i,j}$ : 積層材の開始位置
  - $e_{i,j}$ : 積層材の終了位置
  - $w_{i,j}$ : 積層材の幅



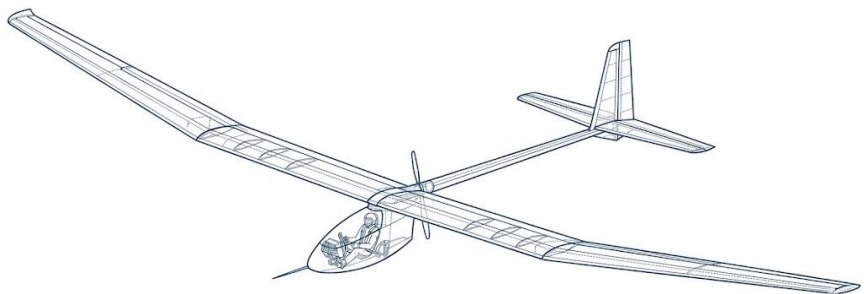
# 研究手法 | 最適化対象のBB関数

[16]N. Namura, in EMO 2025, LNCS 15512 (Springer, 2025).

## BB関数: Human-Powered Aircraft (HPA)<sup>[16]</sup>

人力航空機の翼(右翼)の最適化ベンチマーク

単一目的最適化ベンチマークは3種類収録

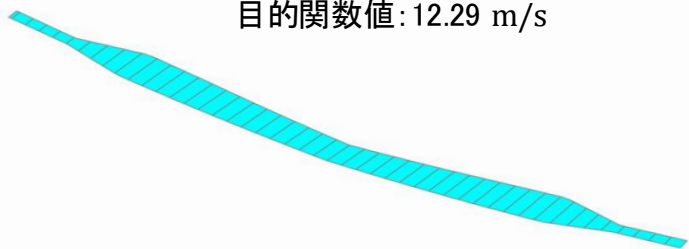


### ■ 今回対象にした問題: HPA103-1

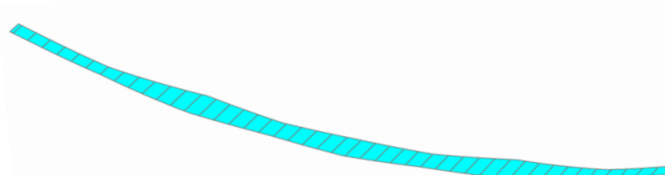
- 単一目的関数, 制約式なし  
(内部的にペナルティ関数として組み込まれている)
- 設計変数は $x_1, \dots, x_{17}$ の17個
- 入力の上下限值は  $[0.0, 1.0]$  ⇨ 32分割の離散値へ
- 整数・バイナリ変換としてone-hot表現
- 目的関数: 巡航速度 (m/s)
- 最適解/関数曲面不明 (中程度の多峰性をもつ)

### 最適化結果の例

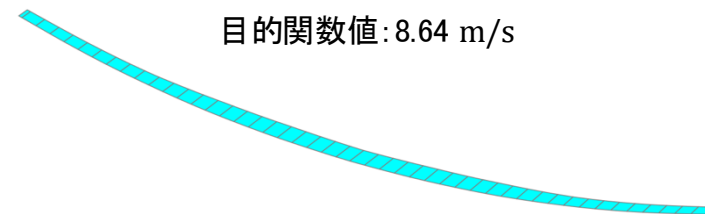
目的関数値: 12.29 m/s



目的関数値: 11.53 m/s



目的関数値: 8.64 m/s



# 研究手法 | 本研究で用いた各種設定

## Factorization machine (FM)の設定・イジングソルバの設定

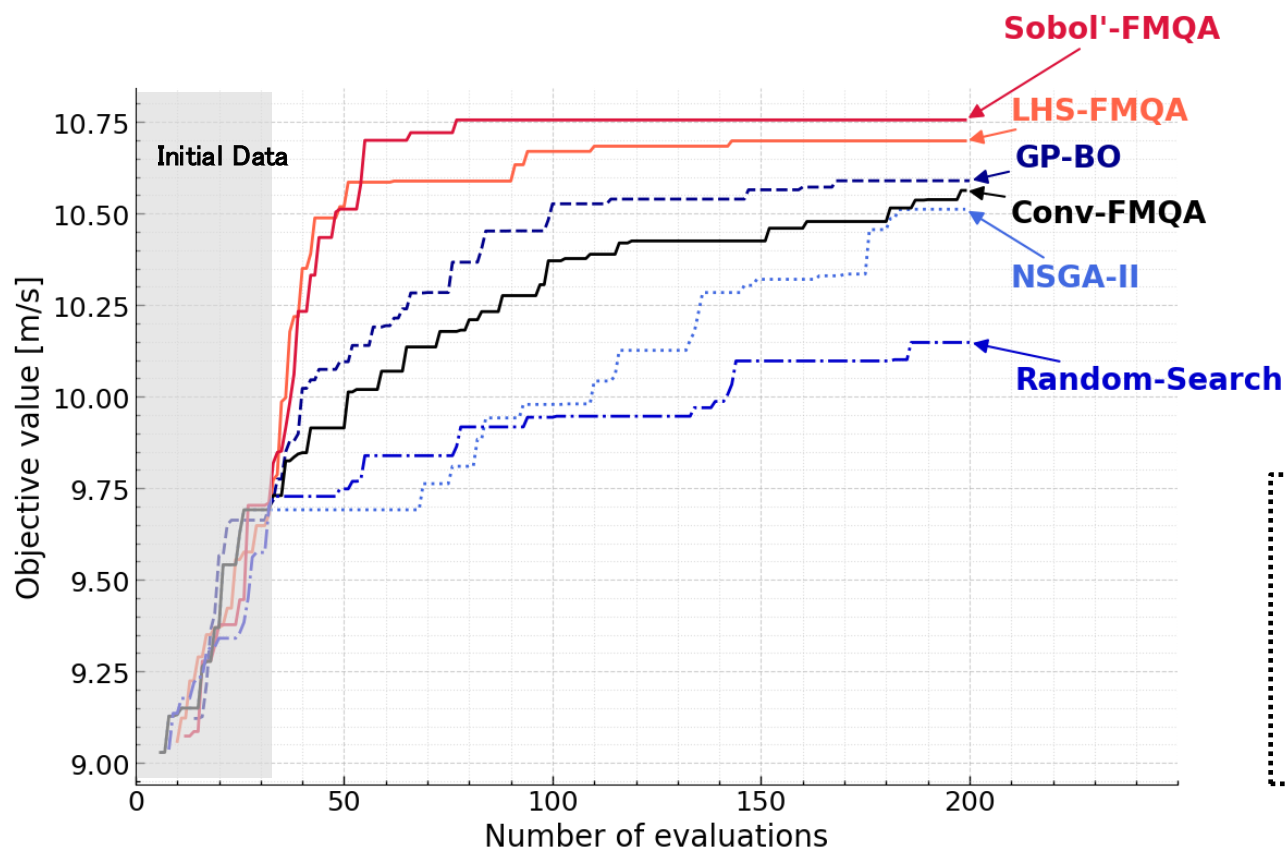
| Factorization machine (FM)の設定 |                                 |
|-------------------------------|---------------------------------|
| FMのランク $K$                    | 5                               |
| エポック数                         | 500                             |
| 学習率                           | 0.5<br>(50 epochごとに0.9倍)        |
| 最適化アルゴリズム                     | AdamW                           |
| ミニバッチサイズ                      | 8                               |
| one-hotエンコーディング制約             | データセットの目的関数値の<br>最大値 $\times 8$ |

| イジングソルバの設定             |                                     |
|------------------------|-------------------------------------|
| 初期学習データのペア数 $D_{init}$ | 32                                  |
| イテレーション数 $N_{iter}$    | 168                                 |
| 生成データ数 $D_{new}$       | 1                                   |
| データ追加セット数 $D_{add}$    | 1                                   |
| イジングマシン                | Fixstars Amplify AE <sup>[17]</sup> |
| timeout                | 2000 ms                             |
| 合計データ数                 | 200                                 |

### ■ 評価指標: 目的関数値の平均値 (10試行)

- 初期学習データをuniform randomで生成するFMQA (Conv-FMQA)
- 古典最適化アルゴリズムとの比較(Optunaライブラリ<sup>[18]</sup>)
  - Gaussian Process Bayesian Optimization (GP-BO)
  - Random Search (RS)
  - Non-dominated Sorting Genetic Algorithm-II (NSGA-II)

# 研究結果 | HPA103-1の最適化結果



## ■ 結果

- 提案手法はConv-FMQAに対して優位
- 提案手法はGP-BOに対しても優位
- 提案手法は初期サイクルから速い巡航速度を探索
- HPA103-2においても, 提案手法の有効性を確認
- 目的関数値の推移がLHSとSobol'では同様の振る舞い

## ■ 解析条件

- BB関数評価回数200回
- FMQAサイクル32回までは初期学習データに相当
- 10試行の目的関数値の平均値
- 初期学習データはone-hot表現を満たすように生成

# 研究結果 | HPA103-1の最適化結果

## ■ 目標値に対するサイクル削減率

- 目標値 (巡航速度 10.5 m/s<sup>[19]</sup>) に対して算出

| 最適化手法         | 目標値までの<br>必要な評価回数 | 評価回数削減率<br>(LHS-FMQA) | 評価回数削減率<br>(Sobol'-FMQA) |
|---------------|-------------------|-----------------------|--------------------------|
| LHS-FMQA      | 49                |                       | 2.0 %                    |
| Sobol'-FMQA   | 48                | -2.0 %                |                          |
| GP-BO         | 100               | 51.1 %                | 52.0 %                   |
| Conv-FMQA     | 181               | 72.9 %                | 73.5 %                   |
| NSGA- II      | 183               | 73.2 %                | 73.8 %                   |
| Random Search | 未到達(*848)         | (*94.2%*)             | 94.3 %                   |

[19] 森, 「鳥人間コンテスト2023を振り返って Ver. 森(QX-23設計)」, 鳥人間コンテスト振り返り, Q-birdman, 2023年10月22日, <https://q-birdman.lix.jp/2023/10/22/%E9%B3%A5%E4%BA%BA%E9%96%93%E3%82%B3%E3%83%B3%E3%83%86%E3%82%B9%E3%83%882023%E3%82%92%E6%8C%AF%E3%82%8A%E8%BF%94%E3%81%A3%E3%81%A6-ver-%E6%A3%AE%E%BC%88qx-23%E8%A8%AD%E8%A8%88%E%BC%89> (参照日: 2026年1月23日).

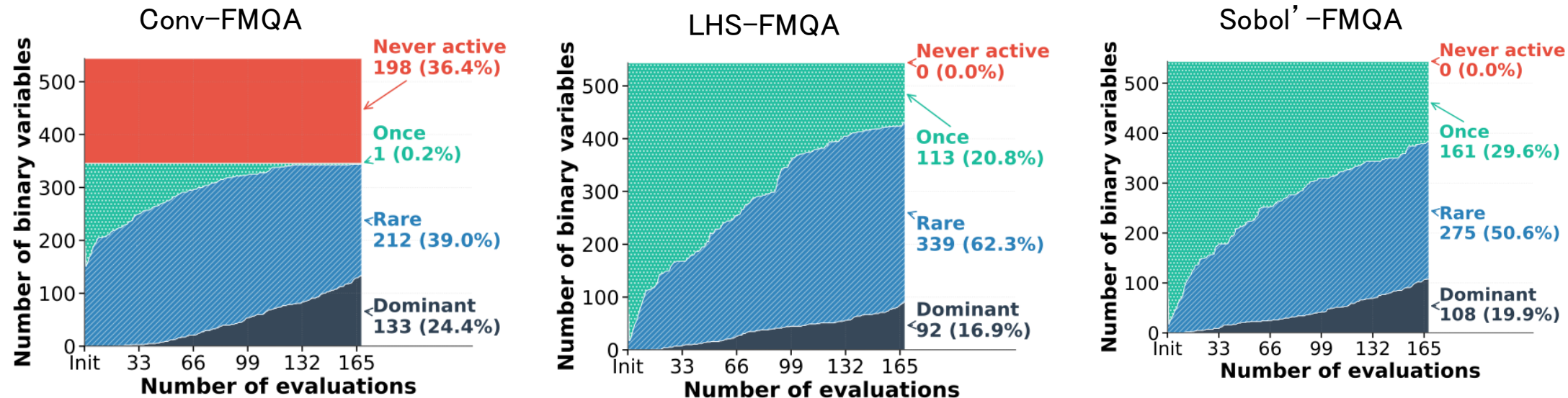
## ■ 結果

- 提案手法は目標値に対して最も少ないサイクル数で到達
- Conv-FMQAに対して最大73.5 %のサイクルを削減
- BBOにおけるサイクル数削減の観点からも提案手法の優位性

\*今回設定した評価回数以上に最適化を行った時に目標値を得られた時の評価回数

# 考察 | データセット内の各ビットで1の値をとった回数について

各サイクルのデータセットにおいて、1が立っているビットの数をプロット



## ■ 結果

- Conv-FMQAは、全体の36.4%のビットが1の値をとらないままサイクルが進む
  - ⇒ Conv-FMQAにおいて、イジングマシンの解探索では、初期学習データで1の値をとったビットの中から良解を探索
  - Conv-FMQAでは、初期学習データで1の値をとった限られたビットの中から良解を探索 → 最適化性能への影響を示唆
- 提案手法は、2回以上1の値をとったビットの割合が高い
  - ⇒ 提案手法はConv-FMQAに比べ、より多くのビットの中から良解を探索
  - Conv-FMQAに比べて、イジングマシンの関数値において、より多くのビットの中から良解を探索

# 研究手法 | Rastrigin関数について

## ■ Rastrigin関数

$$f(n_1, n_2, \dots, n_N) = aN + \sum_{i=1}^N [n_i^2 - a \cos(2\pi n_i)]$$

$n_i$ : 実数変数,  $N$ : 次元数

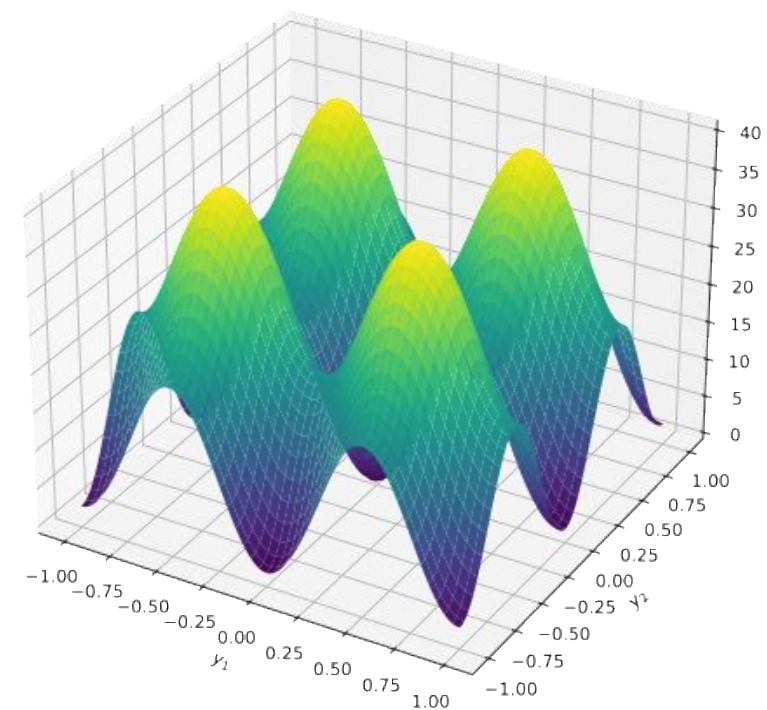
最小解:  $f(n_1, n_2, \dots, n_N) = (0, 0, \dots, 0) = 0$

特徴: 探索空間全体に周期的な波状構造が存在し, 多数の局所最小値を持つ

## ■ 解析方法:

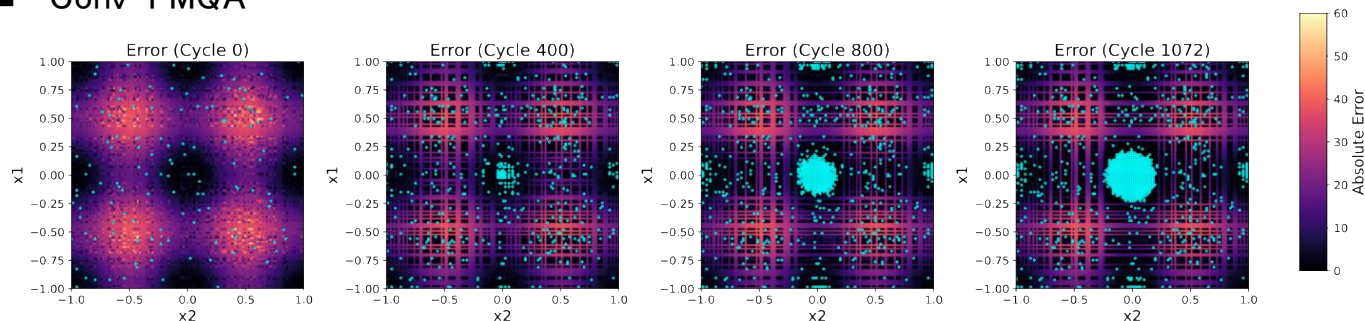
- 入力に対する $f_{BB}(\phi(x_d))$ と $f_{FM}(x_d; \theta)$ の差の絶対値
- 黄色に近い値をとるほど誤差が大きい
- $-1 \sim 1$ の範囲のRastrigin関数をBB関数
- 分割数と初期データ数は128
- FMQAサイクル数は1200
- シアン色はサンプリングしたデータ
- 400サイクルごとのFMのBB関数に対する推定精度を比較

Rastrigin関数

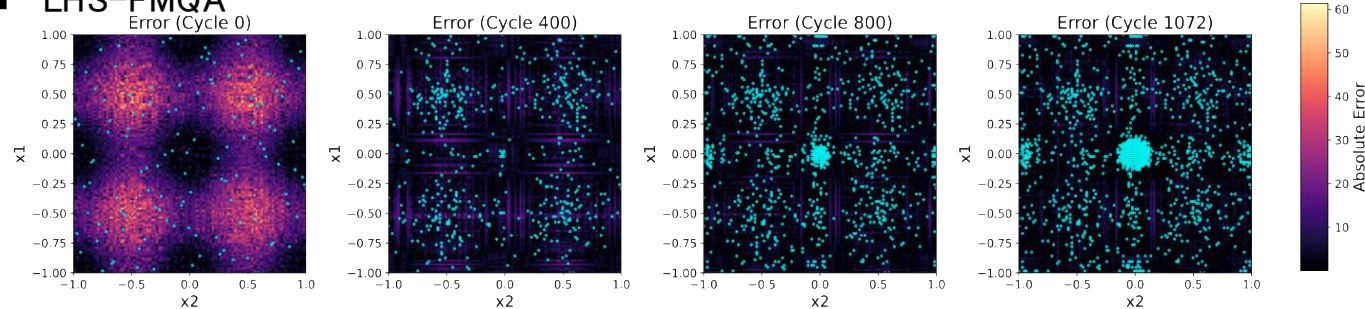


# 研究手法 | Rastrigin関数の最適化結果

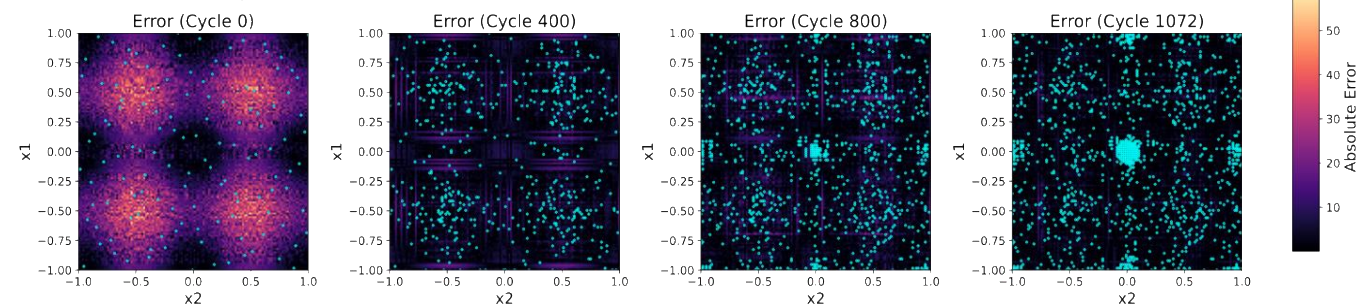
## ■ Conv-FMQA



## ■ LHS-FMQA



## ■ Sobol' -FMQA



## ■ 結果

- 提案手法はBB関数に対するFMの推定精度が高い
- Conv-FMQAは線路上に出力差が大きい部分が分布
- 誤差の大きい領域にはサンプルが存在しない
- 提案手法はConv-FMQAよりサンプルが偏りなく分布

## ■ まとめ

- 本研究は, one-hot表現における1の値を取るビットの割合に着目し, LHSおよびSobol'列に基づく初期学習データ設計をFMQAに導入
- HPA103ベンチマークでは, 提案手法が従来FMQAより高い平均巡航速度を達成し, 目標値到達までの評価回数も削減
- 提案手法がイジングマシンにおける, 選択できるビットを緩和することができるという可能性を示唆

## ■ 現状の課題

- Conv-FMQAにおいて, 未更新パラメータが残存する/しないにはBB関数への依存性があり, どのようなBB関数に対して提案手法が有効となるかが未解明
- 関数形状が既知な問題を通した提案手法の定量的な考察が不十分

## ■ 今後の展望

- 未更新パラメータの更新とBB関数の関係について調査
- 空間網羅性とmarginal bit coverage の効果の切り分けとその検証
- 準ランダム列に基づくFMQAの適用条件の検討

