

# 量子コンピュータ時代の最適化セミナー

～Amplify-BBOptのご紹介及び技術解説～



# 本日の予定

- Fixstars Amplify の紹介
- ブラックボックス最適化とは
- QA-BBOの概要と活用例
- Amplify-BBOptの技術解説
  - 実装方法
  - 業務課題への適用方法
- まとめ

質問は隨時、ZoomのQ&Aへお願いします

# 会社概要

## ● 株式会社 Fixstars Amplify

- 設立 2021年10月
- オフィス 東京都港区芝浦1-1-1 BLUE FRONT SHIBAURA TOWER S 31階
- 事業 最適化のための量子コンピューティング プラットフォーム事業

## ● 株主

- 株式会社 フィックスターズ(東証プライム市場)100%
  - ✓ ソリューション(受託)事業
  - ✓ ソフトウェア高速化プロフェッショナル集団
  - ✓ 日本で初めて D-Wave Systems 社と提携 (2017年)



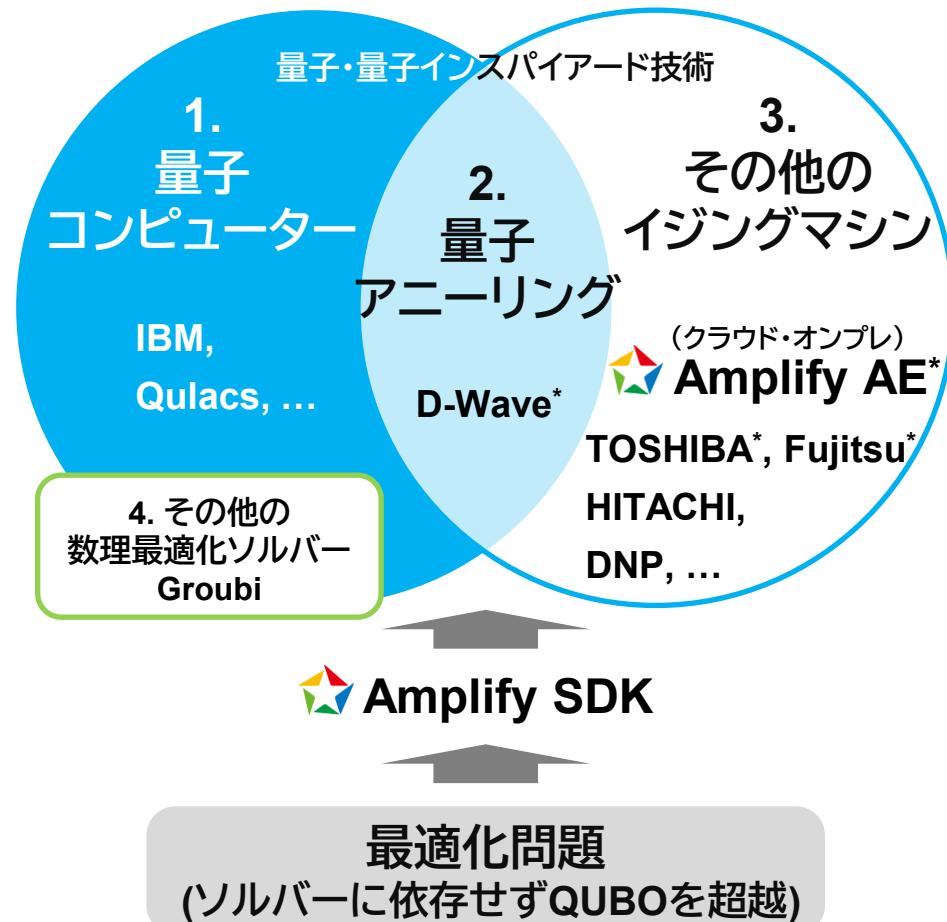
# Fixstars Amplify の役割

- 開発環境 **Amplify SDK** 簡単・最速な組合せ最適化アプリ実装 (Python)
- 実行環境 **Amplify AE** GPUクラウド上で実行されるイジングマシン (リファレンスマシン)

全結合問題: 131,072 ビット

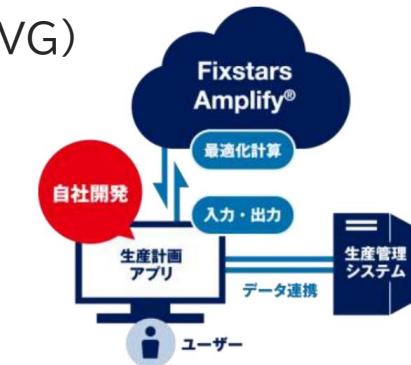
疎結合問題: 262,144 ビット

→ 短期～長期を見据えた組合せ最適化の社会実装の実現 (無料で利用スタート可能)



# 活用領域とユースケース (PoC→実稼働)

- 生産計画
  - 多品種少量生産、保全計画、設備投資、在庫
- 従業員割り当て
  - 食品、輸送、製造
- エネマネ
  - エネルギーミックス、装置の運転制御
- 経路
  - 配送、船舶、無人搬送車 (AVG)
- メディア
  - 最適広告配信
- 研究開発、設計
  - 材料設計
  - 物理シミュレーション



Amplify インタビュー 検索

お客様事例

組合せ最適化問題に取り組む、国内外で700以上の企業・学校が、アプリケーションの開発・実行基盤としてFixstars Amplifyを利用しています。

 Kawasaki Powering your potential

 早稲田大学

 慶應義塾

 0テレ NIPPON TV

 豊田中央研究所

 TATEYAMA

 AVAL DATA

 豊田通商

 ペルメゾンロジスコ

 スマイルボードネット

 会津大学

 KIOXIA



# 活用領域とユースケース (PoC→実稼働)

Amplify インタビュー

検索

- 生産計画

- 多品種小量生産 / 保守計画 / 在庫追跡 / 在庫

- 従業員割り振り

- 食品

- エネマネジメント

- エネルギーミックス、装置の運転制御

- 経路

**1,000** を超える企業、研究所、大学

事例

700以上の企業・学校が、アプリケーションの開発・実行基盤としてFixstars Amplifyを利用しています。

- 併用用法、販売

- 材料設計
- 物理シミュレーション



# QA活用のブラックボックス 最適化の活用事例

# ブラックボックス最適化とは？

- ブラックボックス最適化 (BBO)

- 直接の定式化が困難な目的関数

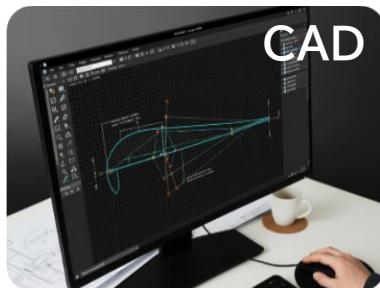
- ✓ 低損失な流体デバイス形状？
    - ✓ 高性能な材料/構造トポロジー？

- 最適化の実施

- ✓ 実験やシミュレーションの試行錯誤により、定式化不可な目的関数を最小化

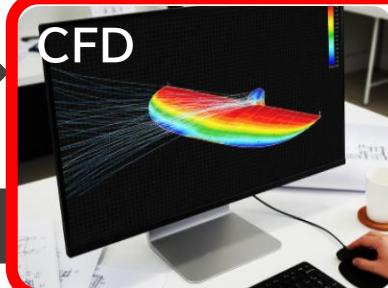
従来手法

- ✓ 実験計画法
- ✓ ベイズ最適化
- ✓ 遺伝的アルゴリズム
- ✓ ...



CAD

評価結果



CFD

修正モデル

## ブラックボックス最適化

入力: 設計した翼形状

↓

流体シミュレーション + 後処理

↓

出力: 抵抗

# QA-BBO: 活用例 (Amplify サンプルプログラム)

## 材料最適化

FMQA

X

## 物理モデル



## 化学プラント 運転条件最適化

FMQA

## 化学シミュレーション

## 翼形状最適化

FMQA

7

## 流体シミュレーション

## 信号制御最適化

FMQA

1

## マルチ・エージェント・シミュレーション

## 機器設計最適化

FMQA

1

## 搅拌シミュレーション

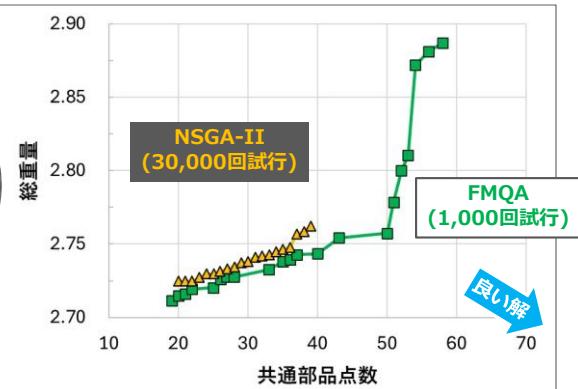
# QA-BBO: ユーザー事例

## 【マツダ様】複数車種の同時設計ブラックボックス最適化

衝突性能を含めた品質特性の条件を満たした上で、部品の軽量化と共に部品数の最大化を行う多目的最適化問題

- 実数変数200以上
- 衝突性能や製造制約、構造制約などの50以上の制約条件、そのうち40程度がブラックボックスな制約
- 従来手法では、良解の発見に1~3万回程度の試行が必要

FMQAにより、1,000回程度(3%)の試行で最適化に成功!  
より広範囲なパレートフロント、より良い最適解を取得!



日経BP [日経クロステックとは?](#) [お申込み](#) [ログイン](#)

キーワードで記事を検索 [検索](#)

日経 **XTECH**

ニュースレター カバストアーリー ランキング 専門メディア マイページ

新着 IT 電機 自動車 建築 土木 #テクノロジー未来投資指標 #トラブル 全館DXインサイト AIエフェクト VR/ARプロトタイプ

日経クロステックトップ > 設計・生産革新 > 製造業に革命をもたらす量子技術 > マツダ、車体設計に量子計算 軽量化や部品共通化を加速

製造業に革命をもたらす量子技術 第2回 [フォローする](#)

### マツダ、車体設計に量子計算 軽量化や部品共通化を加速

佐藤 雅哉 日経クロステック記者  
2025/11/06

自動車 3 min read [有料会員限定記事](#)

シンプル表示 [印刷](#) [保存](#)

PR 次世代ウェアラブル機器に最適!革新的な高品質・低消費電力バイオセンサー

PR IT/製造/建設分野の製品・サービス実証支援情報サイト: 日経クロステックActive

**マツダのラージ薄板群、開発費用25%削減**

From 日経クロステック Special

「AI First」実験の開始 代わり切り拓く

マツダのラージ薄板群、開発費用25%削減

**ピックアップ**

Featured Picks

エヌルギー問題を左右、ガステービンの性能を決める設計技術とは

電池グループが目もくマーケでのAIエージェント活用

自律的に動くエージェンティックAI先駆事例を紹介

製造業のデータ活用も高度化、迅速な意思決定に必要なIT基盤とは

ブルワーに指揮をも、北海道鹿居村に個性的な移住者が集う理由

予知保全導入の3ステップ、生産性向上とコスト削減の実現方法

アサヒビールが導入したナレッジ管理ツール、思ふ所資料の発見も

ティーチングを10倍速、工場レイアウト設計を5倍速にする技術

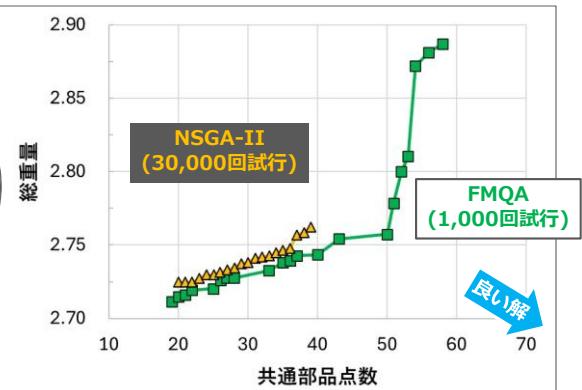
# QA-BBO: ユーザー事例

## 【マツダ様】複数車種の同時設計ブラックボックス最適化

衝突性能を含めた品質特性の条件を満たした上で、部品の軽量化と共通部品数の最大化を行う多目的最適化問題

- 実数変数200以上
- 衝突性能や製造制約、構造制約などの50以上の制約条件、そのうち40程度がブラックボックスな制約
- 従来手法では、良解の発見に1~3万回程度の試行が必要

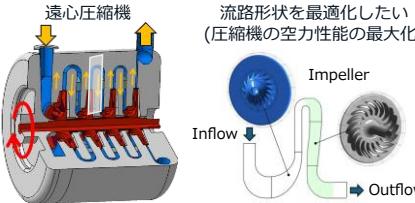
FMQAにより、1,000回程度(3%)の試行で最適化に成功!  
より広範囲なパレートフロント、より良い最適解を取得!



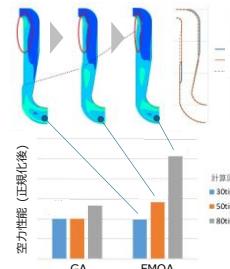
## 【川崎重工業様】ターボ機械形状のブラックボックス最適化

- 従来より遺伝的アルゴリズム(GA)を活用。大規模問題では求解に時間がかかるという課題

同じ試行回数で、従来手法より優れた解を取得



最適化が進むごとに損失発生領域が低減する形状へ



## 【Amplify ユーザー】非公開事例：適用領域と適用目的

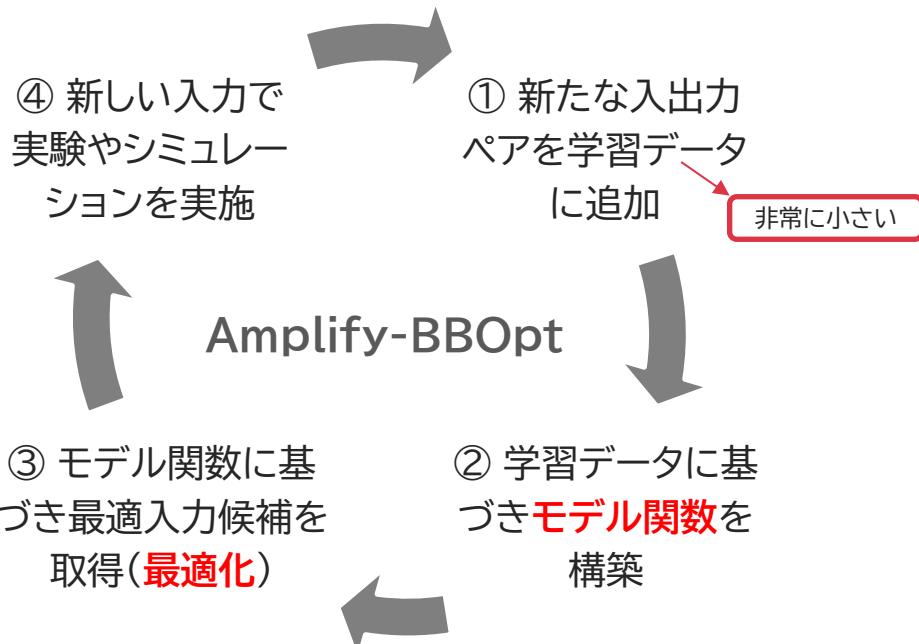
### 【適用領域】

化学、創薬、食品、自動車、電機、通信、重工、エネルギー



# BBOのフローとQA-BBO

- 逐次最適化: 最適化サイクルの実施



- QA-BBO<sup>†</sup>

FMQA [Kitai, et al., Phys. Rev. Res. \(2020\)](#) (東大津田先生)

- **モデル関数** → FM
- **最適化** → QA

Kernel-QA [Minamoto & Sakamoto, arXiv:2501.04225](#)

- **モデル関数** → Kernel model
- **最適化** → QA

□ 高次元の最適化問題に強い！

↔ 次元の呪い

□ 制約条件にも強い！

# Amplify-BBOpt: QA-BBOをより簡単に！

Amplify-BBOpt 公式ドキュメント:  
[amplify.fixstars.com/ja/docs/amplify-bbopt/v1/](https://amplify.fixstars.com/ja/docs/amplify-bbopt/v1/)

- ブラックボックス最適化の本質に注力できるように、簡単実装！
- *Fixstars Amplify* の機能をフル活用:
  - Amplify SDK
    - ✓ 種々の変数やソルバーの利用
    - ✓ ソフト・ハード制約条件の考慮
  - Amplify AE
    - ✓ 高速・大規模な最適化
      - ✓ 数1,000次元+ のバイナリ決定変数BBO
      - ✓ 数100次元+ の実数・離散決定変数BBO

# Amplify-BBOpt: 簡単実装

Amplify-BBOpt 公式ドキュメント:  
[amplify.fixstars.com/ja/docs/amplify-bbopt/v1/](https://amplify.fixstars.com/ja/docs/amplify-bbopt/v1/)

The diagram illustrates the simplification of the code through several stages:

- エンコードイング (Encoding):** The first stage shows the raw C++ code for a blackbox function, which is highly complex and difficult to read.
- サロゲートモデル構築 (Surrogate Model Construction):** The second stage shows the code for building a surrogate model, which is still complex but more structured than the raw code.
- 最適化サイクル (Optimization Cycle):** The third stage shows the code for the optimization cycle, which is highly complex and contains many loops and conditionals.
- ←200行+の実装 (Implementation of over 200 lines):** The fourth stage shows the full implementation of the optimization algorithm, which is over 200 lines of code.
- 10行の実装に→ (Implementation in 10 lines):** The final stage shows the simplified Python API for the optimizer, which is only 10 lines of code.

←200行+の実装  
が  
10行の実装に→



チュートリアル応用編  
ブラックボックス最適化  
(6)

プログラミング難易度 ★ ★ ★ ★

ブラックボックス最適化により、搅拌性能に影響を与える設計パラメータに対して、混合効率を最大化されるような搅拌機の最適設計を実施します。最適化の実施および評価には、濃度分布に基づく簡単な搅拌シミュレーションを用います。

サンプルコード

関連デモ:  
[https://amplify.fixstars.com/ja/demo/fmqas5\\_mixer](https://amplify.fixstars.com/ja/demo/fmqas5_mixer)

```
from amplify_bbopt import FMTrainer, Optimizer, Dataset

# 最適化クラスのインスタンス化
optimizer = Optimizer(
    blackbox=bbofunc,
    trainer=FMTrainer(),
    client=client,
    training_data=Dataset(np.array(x), np.array(y)),
)

# 最適化を実行
optimizer.optimize(num_iterations=20)
```

# Amplify-BBOpt: 問題設定

- 搅拌機 設計パラメータのブラックボックス最適化

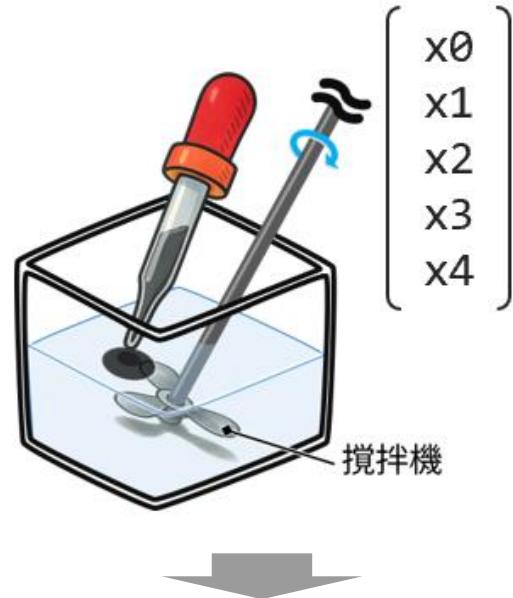
<https://amplify.fixstars.com/ja/demo/fmq5 mixing>

- 搅拌機の設計に関する 5 つの設計パラメータを最適決定

- ✓ これらのパラメータは、具体的に何を指すかは明示されていませんが、搅拌機の性能を左右する「設計上の選択すべき項目」として捉えてください。
    - ✓ 実際の設計や制御と同様に、「最適な条件はあるが、各設計パラメータがどういう効果があるかは必ずしも明確ではない」状態からスタートします。

- ある設計パラメータに対する搅拌機の性能評価値は、搅拌シミュレータで評価します

- ✓ つまりこの評価値が良くなるように設計パラメータを最適に決定する。
    - ✓ シミュレーターは最適パラメータを読み込み、シミュレーションを実施、結果を後処理すると、混ざり具合(濃度の分散=低い方が良い)が出力



【後処理結果】  
混ざり具合(濃度分布の分散)

# Amplify-BBOpt: ブラックボックス最適化の実装

```
from amplify_bbopt import Optimizer, FMTrainer

optimizer = Optimizer(
    blackbox=bbfunc,
    trainer=FMTrainer(),
    client=client,
)

# 初期学習データを生成(既存のデータを使う場合は不要)
optimizer.add_random_training_data(num_data=10)

# 最適化を実行
optimizer.optimize(num_iterations=20)
```

Optimizerクラスをインスタンス化

- **bbfunc**  
ブラックボックス関数を評価する仕組み
- **FMTrainer()**  
モデル関数クラスのインスタンス
- **client**  
Amplifyソルバークライアント  
最適化サイクルを実行
- **Optimizer.optimize()**

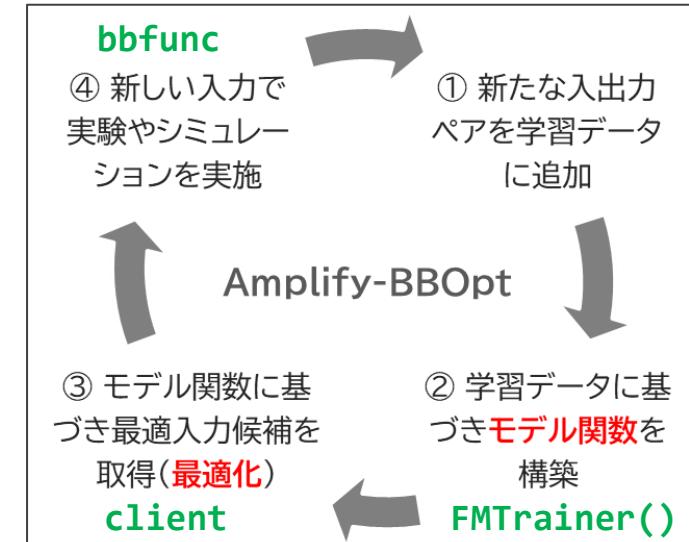
# Amplify-BBOpt: ブラックボックス最適化の実装

```
from amplify_bbopt import Optimizer, FMTrainer

optimizer = Optimizer(
    blackbox=bbfunc,
    trainer=FMTrainer(),
    client=client,
)

# 初期学習データを生成(既存のデータを使う場合は不要)
optimizer.add_random_training_data(num_data=10)

# 最適化を実行
optimizer.optimize(num_iterations=20)
```



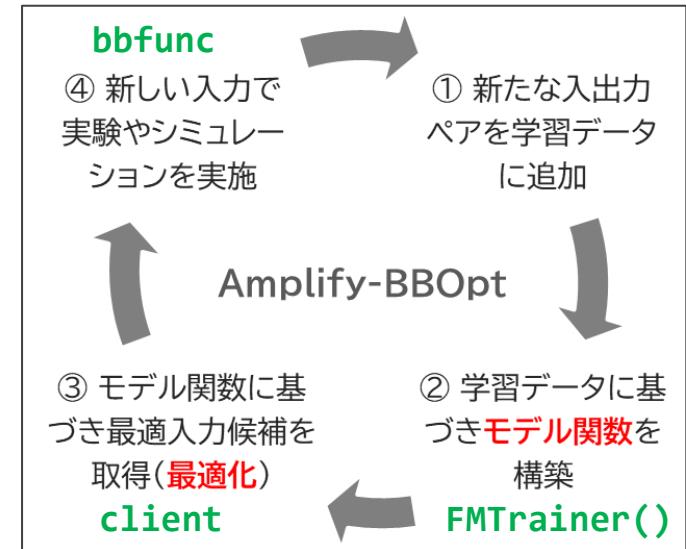
# Amplify-BBOpt: ブラックボックス最適化の実装 (既存のデータを使う場合)

```
from amplify_bbopt import Optimizer, FMTrainer

optimizer = Optimizer(
    blackbox=bbfunc,
    trainer=FMTrainer(),
    client=client,
    training_data=Dataset(x, y),  # 既存のデータを使う場合
)

# 初期学習データを生成(既存のデータを使う場合は不要)
# optimizer.add_random_training_data(num_data=10)

# 最適化を実行
optimizer.optimize(num_iterations=20)
```



# Amplify-BBOpt: ブラックボックス関数を評価する仕組みの実装 (シミュレーションの場合)

```
from amplify_bbopt import blackbox, IntegerVariable
```

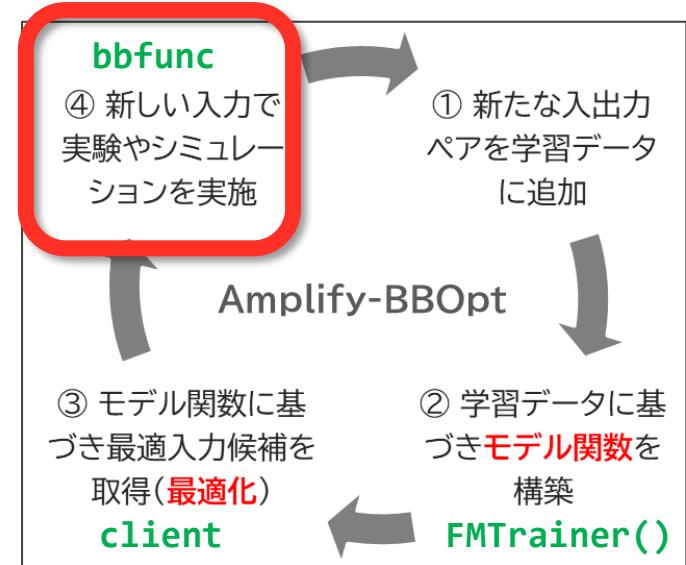
おまじない

```
@blackbox
def bbfunc(
    x0: int = IntegerVariable((2, 10)),
    x1: int = IntegerVariable((5, 20)),
    x2: int = IntegerVariable((0, 45)),
    x3: int = IntegerVariable((1, 5)),
    x4: int = IntegerVariable((1, 4)),
) -> float:
    s = MixingSimulator(x0, x1, x2, x3, x4)
    c_std = s.simulate(duration=500)
    print(f"{c_std:.3f}")
    return c_std
```

決定変数の種類

Amplify-BBOpt が提案した最適入力候補が引数として `bbfunc` に渡される

最適入力候補に対する評価結果が Amplify-BBOpt に返却される



# Amplify-BBOpt: ブラックボックス関数を評価する仕組みの実装 (実験の場合)

```
from amplify_bbopt import blackbox, IntegerVariable
```

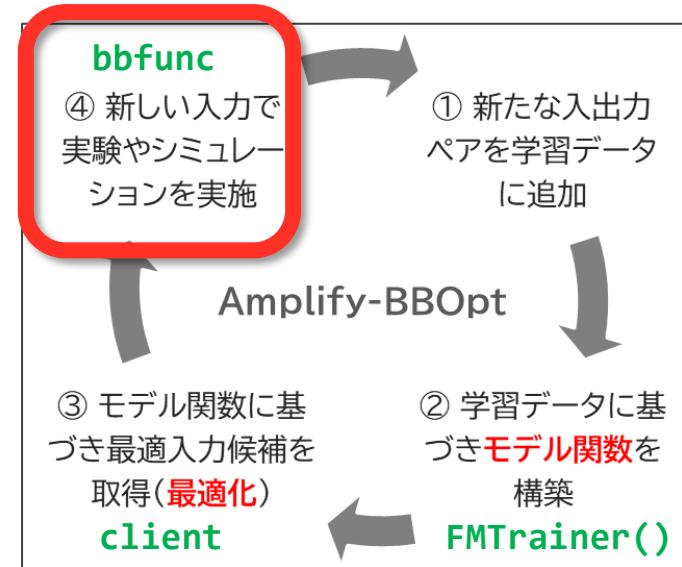
おまじない

```
@blackbox
def bbfunc(
    x0: int = IntegerVariable((2, 10)),
    x1: int = IntegerVariable((5, 20)),
    x2: int = IntegerVariable((0, 45)),
    x3: int = IntegerVariable((1, 5)),
    x4: int = IntegerVariable((1, 4)),
) -> float:
    c_std = input(
        f"\n{x0=}, {x1=}, {x2=}, {x3=}, {x4=} の条件で"
        "実験を行い、得られた濃度の分散を入力してください"
    )
    print(f"\n{c_std=:.3f}")
    return float(c_std)
```

決定変数の種類

Amplify-BBOpt が提案した最適入力候補が引数として `bbfunc` に渡される

最適入力候補に対する評価結果が Amplify-BBOpt に返却される



# Amplify-BBOpt: ソルバークライアントの設定

```
from amplify import AmplifyAEClient

# Amplify Annealing Engine を使う場合
client = AmplifyAEClient()
client.parameters.time_limit_ms = 1000 # ms
```

Amplify AE 以外にも様々なソルバーを選択可能！

## 標準マシン

標準マシンとは、Fixstars Amplifyからご利用申し込みが可能なマシンです。詳しくは各マシンのご利用方法をご覧ください。



### 標準マシン

D-Wave Systems  
D-Wave The Leap™  
Quantum Cloud Service

量子アニーリングマシンおよび量子アニーリングマシンと古典アルゴリズムで動作するハイブリッドソルバーを提供するクラウドサービスです。

製品サイト [↗](#)  
Bring Your Own License対応

ご利用方法



### 標準マシン

東芝デジタルソリューションズ  
SQBM+

東芝が開発した技術「シミュレーテッド分岐アルゴリズム」を用いた組合せ最適化ソルバー「シミュレーテッド分岐マシン」を核とした量子インスペイアード最適化ソリューションです。

製品サイト [↗](#)  
Bring Your Own License対応

ご利用方法



### 標準マシン

富士通  
デジタルアニーラ

量子現象に着想を得たコンピューティング技術です。現在の汎用コンピュータでは解くことが難しい「組合せ最適化問題」を高速で解くことができます。

製品サイト [↗](#)

ご利用方法

## BYOLマシン

BYOLマシンとは、ご自身でのマシンを使える環境をご用意いただく必要があるマシンです。詳しくはドキュメントをご覧ください。

### 日立製作所

CMOSアニーリングマシン

製品サイト [↗](#)

### Gurobi

Gurobi Optimizer

製品サイト [↗](#)

### IBM

IBM Quantum

製品サイト [↗](#)

# Amplify-BBOpt: 利用可能な手法とその特徴

## ● FMQA

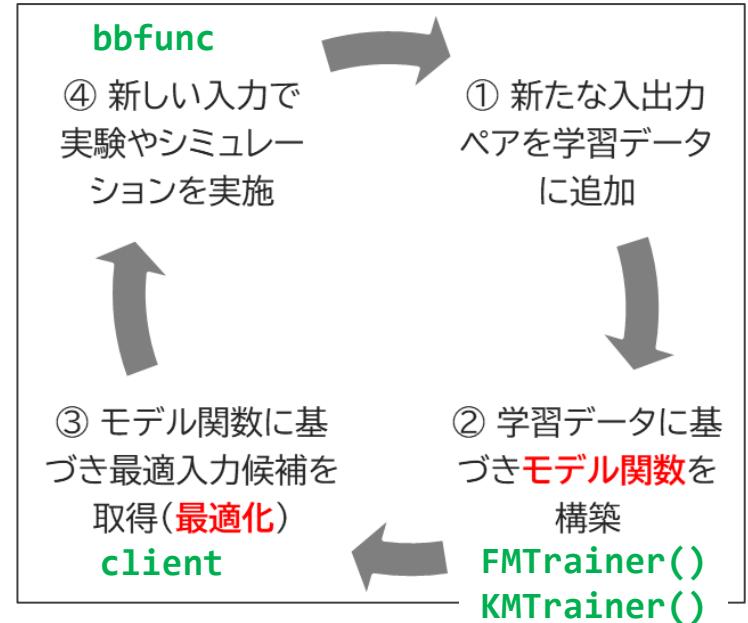
Kitai, et al., *Phys. Rev. Res.* (2020) (東大津田先生)

- モデル関数クラス: `FMTrainer()`
- 長所: モデル構築の自由度高
- 短所: 構築が遅い、学習パラメータ多

## ● Kernel-QA

Minamoto & Sakamoto, arXiv:2501.04225 (2025)

- モデル関数クラス: `KMTrainer()`
- 長所: モデル構築が速い、学習パラメータ少
- 短所: 実装が困難 (Amplify-BBOptで解消)



# サンプルプログラム: 搅拌機 設計パラメータの最適化

- Amplify SDK と PyTorchによる FMQA 生実装チュートリアル  
(そのまま実行可能)
  - [https://amplify.fixstars.com/ja/demo/fmqa 5 mixing](https://amplify.fixstars.com/ja/demo/fmqa_5_mixing)
- Amplify-BBOpt による Kernel-QA 実装チュートリアル  
(そのまま実行可能)
  - [https://colab.research.google.com/drive/1a8XP0-wsDbrWa8Lp3PGDhonbuAX\\_hIrx](https://colab.research.google.com/drive/1a8XP0-wsDbrWa8Lp3PGDhonbuAX_hIrx)
- Amplify-BBOpt ドキュメント
  - <https://amplify.fixstars.com/ja/docs/amplify-bbopt/v1/>

# ブラックボックス最適化 Tips

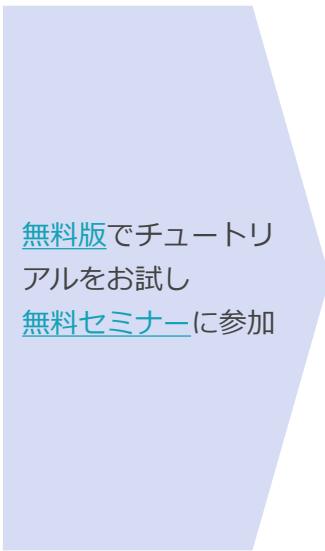
詳しくは、プライベートトレーニングや伴走型支援において解説・実演いたします。

- データスケーリング
  - 最適化(最小化問題)により適したモデル関数を構築するためのデータ作成
- 目的関数スケーリング
  - 多目的最適化における目的関数間のバランス
    - ✓ 自動調整
    - ✓ 手動調整
- データ保存
  - 目的関数の評価(シミュレーション・実験)に長時間かかる場合のワークフロー
- モデル関数構築のコツ
  - FMQAにおける学習パラメータの調整の方針及びモデル性能確認

# 研究・開発者向けおすすめの進め方

二次・非線形を上手に使いこなせるように、**弊社と一緒に**取り組みを進めていきましょう！

Step: 1~2ヶ月程度



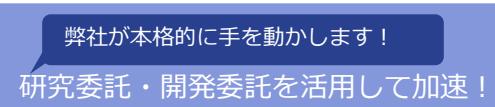
Step2: 3ヶ月程度



Step3 : 6ヶ月～



Step4: 2~6ヶ月程度



Step5: 4~6ヶ月程度



Step6



# プライベートトレーニング

<https://amplify.fixstars.com/ja/seminar/private-training>



全4回のレクチャーとお客様に実施いただく「課題」を含む約2ヵ月のコースです。コースの前半では、量子アニーリング・イジングマシン専用の開発／実行環境である Fixstars Amplify を用いて Python 言語による組合せ最適化アプリケーション開発方法を学びます。コースの後半では、お客様が解きたい課題を持ち込んでいただき、弊社のエンジニアと一緒に解きます。量子アニーリング・イジングマシンを使って実課題の解決に取り組みたい方に最適なコースです。



## ● 第1回・第2回クラス（各3時間）

前半は、以下を段階的に学び、持ち込み課題や実課題を解くための準備を行います

- 量子アニーリング・イジングマシンの概要

- 組合せ最適化問題の定式化
- 学習環境Fixstars Amplifyの使い方

- プログラミングハンズオン

- 目的関数の定式化：数の分割問題、画像のノイズ除去
- 制約条件の定式化：会議室割り当て問題、数独
- 高度な組合せ最適化問題の定式化：巡回セールスマン問題
- 応用事例を用いたハンズオン：生産計画（定式化ベース）、材料探索や設計最適化（ブラックボックス最適化ベース）等

- 第3回・第4回のワークショップに向けて、持ち込み課題の内容や進め方に関するディスカッション

ン

## ● 課題

第3回、第4回の持ち込み課題のワークショップに向けて、事前準備を進めていただきます

## ● 第3回・第4回クラス（各1.5時間）

後半は、お客様の持ち込み課題を使ったワークショップを行います

- 持ち込み課題の内容に関するディスカッション

- 利用するデータの確認

- 定式化や実装の方向性の確認や、実際の実装で苦労している点などに関してディスカッション

# Fixstars Amplify: クラウド利用料

個人単位のプラン ～主に研究者・開発者向け～					組織単位のビジネスプラン ～社内システムの利用向け～		
(金額は税抜)	ベーシック	スタンダード	プレミアム	Sプレミアム	ビジネススタンダード	ビジネスプレミアム	ビジネスSプレミアム
月額利用料	無料	10万円 (1名) 30万円 (最大5名)	20万円 (1名) 60万円 (最大5名)	30万円 (1名) 90万円 (最大5名)	20万円 (1アプリ) (同一組織内であれば同一アプリのユーザー数は無制限)	40万円 (1アプリ)	60万円 (1アプリ)
計算環境	スマール	ミディアム	ラージ	スーパーラージ	ミディアム	ラージ	スーパーラージ
利用GPU (マルチGPUオプションあり)	NVIDIA V100	NVIDIA V100	NVIDIA A100	NVIDIA H100	NVIDIA V100	NVIDIA A100	NVIDIA H100
1ジョブの実行時間 (実行時間延長オプションあり)	10秒	1分	10分	15分	1分	10分	15分
月間累計実行回数 (実行回数追加オプションあり)	制限の可能性あり	無制限			- (上限を設定する可能性あり)		
月間累計実行時間	制限の可能性あり	無制限			- (上限を設定する可能性あり)		
東芝 SQBM+ オプション	10秒	30万円 (1名)、90万円 (最大5名)			-		
富士通 DA オプション	10秒	50万円 (1名)、150万円 (最大5名)			-		
D-Wave オプション	3分/月	ご相談可 手厚い伴走型支援			-		
サポート	ベーシック	スタンダードサポート	プレミアムサポート	プレミアムサポート	スタンダードサポート	スタンダードサポート	スタンダードサポート
Plus オプション	-	-	月額50万/人		-	-	-

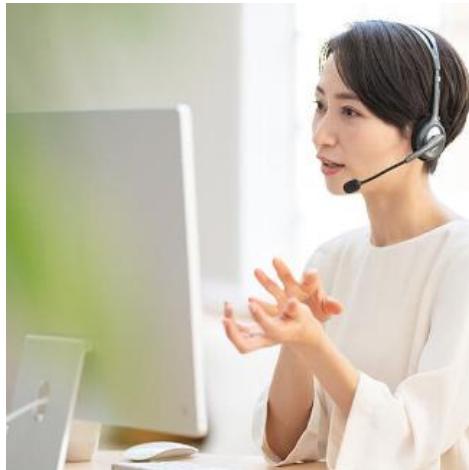
# Plusオプション(プレミアムプラン/Sプレミアムプランで追加可能なオプション)

## Plusオプション

料金：月額50万円（税込55万円）/ユーザー

- 問い合わせ回数は無制限
- ご質問には翌営業日までに回答（目安）
- 定式化・実装等のご相談
- 特別対応窓口や定例会の設置
- 特別技術支援※

※特別技術支援の内容に応じて期間等は個別にご相談



## 特別技術支援の例

### □ 開発支援

- ユーザー様の実装にお困りの部分に関して、弊社エンジニアがサンプルコードを作成して提供します
- 開発支援にかかる期間については個別相談となります

### □ コード最適化レビュー

- 弊社エンジニアがユーザー様が実装したコードを確認し、よりよい実装などがあればサンプルコードを作成して提供します

### □ 評価支援

- ユーザー様にご提供いただく問題設定で、弊社のエンジニアが様々な計算環境で実験・評価して結果をレポートします
- 複雑な問題になると限られた計算環境では十分な精度の解が得られない可能性があります。本支援では、異なる GPU (V100/A100/H100) や、GPU 数 (1機～4機)、実行時間 (～1時間) で実験・評価し、最適な計算環境の評価・検討のご支援をします
- 問題設定については、ユーザー様にプログラムやデータを送付してもらう、もしくは、問題の概要をテンプレートで回答いただく形になります
- 評価支援にかかる期間については個別相談となります

# 今後のセミナーの予定

今後も定期的に無料セミナーを開催します！

2025/12/4

## 「Amplify-BBOpt 技術解説」

イジングマシン活用のブラックボックス最適化を簡単実装可能なライブラリ Amplify-BBOpt の使い方を紹介。

2025/12/18 (受付中)

## 「経路最適化」 (Annealing Engine)

複数車両の経路距離を最小化する経路最適化にハンズオンで取り組みます。

2025/1/15 (受付準備中)

## 「AE (v1) 技術解説」

メジャーアップデートされた Amplify 独自開発イジングマシンである Annealing Engine について解説。

2025/1/22 (予定)

## 「ブラックボックス最適化 (攪拌機器の最適設計)」

機器の設計パラメータ最適化をハンズオンで実施。

2025/2/12 (予定)

## 「多目的最適化 技術解説」

定式化ベースの多目的最適化に関する定式化やスケーリングに関する Tips を紹介。

2025/2/19 (予定)

## 「最適エネルギー管理 メント」

エネルギー管理最適化をハンズオンで実施。

ご質問・ご不明点がありましたら、お問い合わせフォームでご連絡下さい

<https://amplify.fixstars.com/ja/contact>

# Q&A