

OpenBox 技术报告

作品类别：系统软件

一、简介

近年来，人工智能与机器学习备受人们关注，越来越多的人使用机器学习模型解决实际问题，如数据分析与预测、人脸识别、商品推荐等。在应用机器学习模型的过程中，模型超参数的选择对模型性能有着重要影响，超参数优化问题成为了机器学习的重要挑战之一。超参数优化是典型的黑盒优化问题，对于优化目标不存在具体表达式或梯度信息，且验证代价较大，其目标是在有限的验证次数内，尽快找到全局最优解。

除了超参数优化外，黑盒优化的典型场景还包括：自动化 A/B 测试、实验设计、数据库参数调优、处理器架构和芯片设计、资源配置优化、自动化化工设计等。

OpenBox 是我们针对黑盒优化设计的一套开源系统。其使用场景广泛，不仅支持传统的单目标黑盒优化（如超参数优化），还支持多目标优化、带约束条件优化、多种参数类型、迁移学习、分布式并行验证、多精度优化等。OpenBox 不仅支持本地安装与优化调用，还提供在线优化服务，用户可通过网页可视化监控并管理优化过程，也可以部署私有优化服务。OpenBox 拥有便捷的使用接口，并且能根据用户任务自动选择合适的优化算法，减轻用户使用负担。

我们为 OpenBox 撰写了详细的在线使用教程与文档，并且在多个测试任务上进行了实验，效果优于已有黑盒优化/超参数优化系统。

二、设计方案

OpenBox 是解决黑盒优化（超参数优化）问题的高效且通用的开源系统，我们的设计遵循以下理念：

- 易用：用户以最小代价使用黑盒优化服务，可通过用户友好的可视化界面监控与管理优化任务。
- 性能优异：集成最先进（state-of-the-art）的优化算法，并可自动选择最优策略。
- 资源感知管理：为用户提供基于成本（时间、并行数等）的建议。
- 高效：充分利用并行资源，并利用迁移学习、多精度优化加速搜索。
- 规模可扩展：对于输入维度、目标维度、任务数、并行验证数量等有较好的可扩展性。

- 错误容忍、系统可扩展性、数据隐私保护。

相较于现有的黑盒优化系统，OpenBox 支持更广泛的使用场景，包括多目标优化、带约束条件优化、多种参数类型、迁移学习、分布式并行验证、多精度优化等。OpenBox 与现有系统的支持场景对比如图 1：

System/Package	Multi-obj.	FIOC	Constraint	History	Distributed
Hyperopt	×	✓	×	×	✓
Spearmint	×	×	✓	×	×
SMAC3	×	✓	×	×	×
BoTorch	✓	×	✓	×	×
GPflowOpt	✓	×	✓	×	×
Vizier	×	✓	×	△	✓
HyperMapper	✓	✓	✓	×	×
HpBandSter	×	✓	×	×	✓
OPENBOX	✓	✓	✓	✓	✓

图 1 OpenBox 与其他黑盒优化系统支持场景对比。

- 多种参数类型（FIOC，即浮点型、整型、序数型、类别型）：输入参数不局限于浮点型（实数），例如超参数优化中，SVM 模型核函数用类别型表示，如果单纯用整型代替序数型或类别型参数，将对参数附加额外的序关系，不利于模型优化。
- 多目标优化（Multi-objective Optimization）：同时优化多个不同（甚至相互冲突）的目标，例如同时优化机器模型准确率和模型训练/预测时间。
- 带约束条件优化（Optimization with Constraints）：最优化目标的同时，要满足（黑盒）条件。

现有系统往往不能同时支持以上特性。OpenBox 在支持上述场景的基础上，还支持：

- 利用历史任务信息（History）指导当前优化任务，即迁移学习（Transfer Learning）。
- 提供并行优化算法、支持分布式（Distributed）验证，充分利用并行资源。
- 提供多精度（Multi-Fidelity）优化算法，从而在高验证代价场景下（如大数据集，机器模型训练时间较长），进一步加速搜索。

OpenBox 不仅支持本地使用，还支持服务部署。与算法库（Library）不同，用户无需部署与运行优化算法，只需要与服务交互，获取参数配置、进行验证并更新结果。OpenBox 提供了开源优化服务，用户既可以使用我们提供的 OpenBox

在线服务，也可以通过开源代码将服务部署在自己的服务器中，满足用户自定义与隐私性需求。

三、实现方案

（一）使用方式与优化流程

目前 OpenBox 支持全平台（Linux、macOS、Win10）使用，并为用户提供本地和服务两种使用方式：

- 本地使用：用户可通过 PyPI 安装 Python 包，调用黑盒优化（超参数优化）框架与算法。只需运行命令 `pip install openbox` 即可安装。
- 服务使用：用户可通过接口访问 OpenBox 服务，从服务端获取推荐的参数配置，在本地执行参数性能验证（如机器学习模型训练与预测）后，将结果更新至服务端。用户可通过访问服务网站页面，可视化监视与管理优化过程。

OpenBox 本地优化模块有两个关键抽象：优化器（Optimizer）和建议器（Advisor）。用户执行优化任务时，需向优化器提供目标函数（Objective Function）和参数搜索空间（Configuration Space）。之后优化器持续从建议器咨询需要验证的参数，调用目标函数进行验证，直至达到约束条件。

OpenBox 本地优化不仅支持串行优化，还支持并行优化，包括两种模式：（1）单机多进程并行优化；（2）及基于消息队列的分布式并行优化。分布式并行优化架构如图 2，主要组件为：优化器、验证器、配置队列、结果队列。优化流程为：优化器根据建议器向配置队列发送需要验证的配置，验证器从配置队列获取配置并验证，验证器将验证结果发送至结果队列，优化器从结果队列获取结果并更新历史。在这种模式下，验证器可部署在不同机器节点上，充分利用分布式机器资源加速验证。

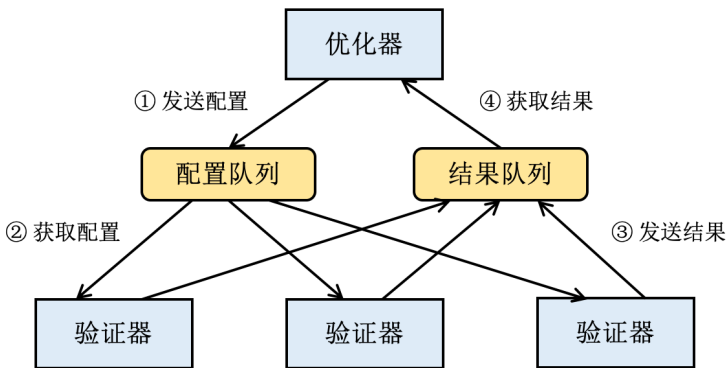


图 2 基于消息队列的分布式并行架构图。

OpenBox 本地优化的不同方式通过不同优化器实现，代码见

[openbox/optimizer]，包含上述优化器。

OpenBox 在线服务架构如图 3：

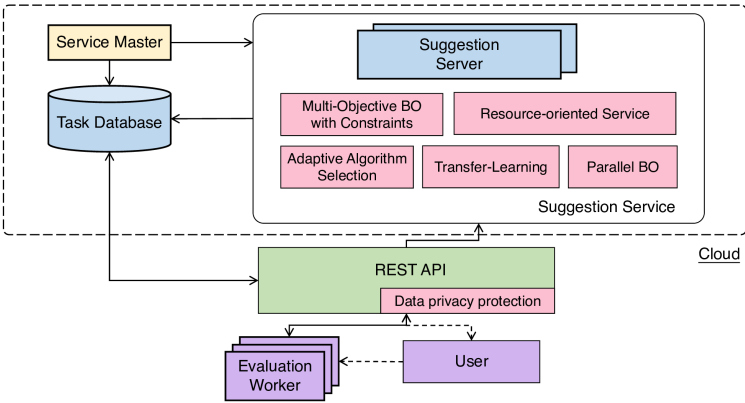


图 3 OpenBox 在线服务架构图。

OpenBox 在线服务包含 5 个主要组件：服务主机、任务数据库、建议服务器、验证工作者、REST API。运行流程为：由用户向服务主机提交优化任务；由服务主机根据负载均衡分配建议服务器，将任务信息与所述建议服务器绑定；以及由用户提供的验证工作者与建议服务器交互，持续从建议服务器拉取新配置，进行验证，并将结果更新至建议服务器，直至优化任务结束。

OpenBox 在线服务代码见[openbox/artifact]。

（二）基础算法模块

OpenBox 集成了最先进的黑盒优化算法，以支持广泛的使用场景。目前包含算法如下（新算法仍在不断开发中）：

1. 贝叶斯优化

贝叶斯优化是一种迭代式黑盒优化框架，使用代理模型对历史数据建模，然后使用采集函数评估各个配置的验证价值，对采集函数执行优化，得到下一个需要验证的配置。主要组件为代理模型、采集函数、采集函数优化器。

（1）代理模型。OpenBox 支持不同的代理模型，包括但不限于：

- 高斯过程。
- 概率随机森林。
- TPE 模型。
- LightGBM 模型。

代理模型代码见[openbox/surrogate]。

（2）采集函数。OpenBox 支持不同的采集函数，包括但不限于：

- 单目标优化：EI、PI、UCB、MES、EIPS、Uncertainty。
- 单目标带约束优化：EIC。
- 多目标优化：ParEGO、EHVI、MESMO、USEMO。
- 多目标带约束优化：EHVIC、MESMOC。

采集函数代码见[openbox/acquisition_function]

(3) 采集函数优化器。OpenBox 支持不同的采集函数，包括但不限于：

- 随机搜索。
- 交替局部搜索和随机搜索。
- 多启动 L-BFGS-B 优化。
- CMA-ES 优化。
- 蒙特卡洛优化。

采集函数代码见[openbox/acq_maximizer]

2. 遗传算法

遗传算法是一种高效的黑盒优化算法，通过种群演化、突变、杂交、淘汰方法，提升观测结果，执行优化。OpenBox 支持单目标遗传算法 EA、REA，以及多目标遗传算法 NSGAII，并支持广泛的突变、杂交、淘汰等方法。

3. 随机搜索算法

随机搜索算法是最简单的黑盒优化算法，但往往能超过人类专家的优化结果。OpenBox 亦支持随机搜索算法。

4. 算法自动选择

OpenBox 支持根据任务类型（参考指标为参数空间大小、类型，优化轮数等）自动判断适合的黑盒优化算法，减轻用户使用负担。在 OpenBox 系统中，上述不同算法对应不同的建议器（Advisor），代码见[openbox/core]。

（三）高级算法模块

OpenBox 在支持广泛的基础黑盒优化算法的基础上，还支持许多高级特性，包括同步/异步并行优化、多精度优化、迁移学习等，进一步加速优化过程，满足用户的多种需求。

1. 同步/异步并行优化

传统的贝叶斯优化每一轮只给出一组参数建议，为了加速优化过程，充分利

用并行资源，需要对贝叶斯优化进行改进以支持同时多组超参数的建议和评估。OpenBox 支持多种并行贝叶斯优化算法，包括但不限于：

- 基于局部惩罚的并行推荐。
- 基于值插补的并行推荐。
- 基于多采集函数集成的并行推荐。

OpenBox 的并行优化算法支持同步、异步两种模式，两种模式对比示意如图 4。同步模式下，建议器每轮给出多组配置建议，待所有配置验证结果返回后生成下一轮的配置建议。异步模式下，每当一个验证工作者完成验证，就从建议器获取一组新的配置，而无需等待。

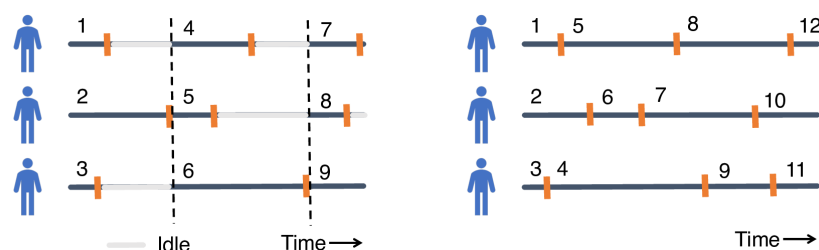


图 4 同步（左）和异步（右）并行验证对比。

OpenBox 系统在上述的本地单机并行、分布式并行和在线服务使用中，均支持同步并行和异步并行两种模式。并行建议器代码见[openbox/core]

2. 早停与多精度优化

在优化过程中，为节约验证资源，可以对配置验证过程执行早停，仅获取部分验证结果。OpenBox 支持早停方法包括：

- 基于曲线延展的性能预估早停。
- 基于排名的早停。
- 多阶段早停。

早停的引入会产生部分验证结果，因此配置验证结果被分为不同精度。OpenBox 支持利用多精度信息执行优化，支持的算法包括：

- 多阶段早停 SH 算法。
- 自适应启动 Hyperband 算法。
- 基于贝叶斯优化推荐的自适应启动 BOHB 算法。
- 基于多精度集成贝叶斯模型推荐的自适应启动 MFES-HB 算法。

多精度优化代码见[openbox/apps/multi-fidelity]。

3. 迁移学习

传统黑盒优化算法面临冷启动问题。而人类专家在执行优化时，往往能通过过去任务的优化经验，指导当前任务的优化。迁移学习借鉴了这种思想，利用以往优化任务历史经验，加速当前任务的优化过程。OpenBox 支持多种迁移学习算法，包括但不限于：

- 权重集成 RGPE 方法。
- 栈集成 SGPR 方法。
- 多阶段迁移学习方法。

迁移学习代码见[openbox/surrogate/tlbo]

（四）可视化模块

OpenBox 支持优化过程及结果可视化，为用户提供了结果统计与分析，包括但不限于：

- 优化任务总结。
- 优化性能曲线。
- 超参数重要性分析。
- 配置结果折线可视化。

可视化效果详见运行/测试结果章节。

（五）开源支持

OpenBox 作为一个开源黑盒优化系统，有着完善的开源支持。包括在线文档、使用样例、测试套件、测试代码等。

1. 在线文档

我们为 OpenBox 撰写了详细的在线文档，不仅介绍了我们的系统特性，还指导用户如何安装、使用基础功能、使用高级功能、部署服务。在线文档示例如图 5。

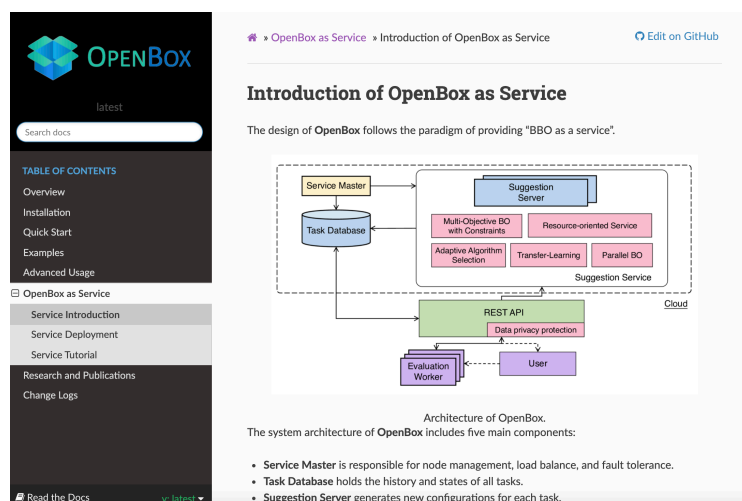


图 5 OpenBox 在线文档示例。

OpenBox 在线文档代码见[docs]。

OpenBox 在线文档已部署至在线服务器，可通过网址访问。

- OpenBox 英文文档: <https://open-box.readthedocs.io/en/latest/>
- OpenBox 中文文档: https://open-box.readthedocs.io/zh_CN/latest/

2. 使用样例

OpenBox 为用户提供了丰富的示例代码，包括如何优化单目标、多目标、带约束任务，如何执行同步并行、异步并行验证，如何调优机器学习模型 XGBoost、LightGBM 等。代码见[examples]。

3. Benchmark 测试套件

OpenBox 内置了丰富的测试套件，以供研究用途及测试使用，包含黑盒数学函数问题、机器学习模型超参数优化问题等。代码见[openbox/benchmark]。

4. 测试代码

OpenBox 包含了完善的测试代码，用户可参考测试不同组件的性能。代码见[test]。实验复现代码见[test/reproduction]。

四、运行/测试结果

(一) 使用样例及运行结果


```

import numpy as np
import matplotlib.pyplot as plt
from openbox import Optimizer, sp

# Define Search Space
space = sp.Space()
x1 = sp.Real("x1", -5, 10, default_value=0)
x2 = sp.Real("x2", 0, 15, default_value=0)
space.add_variables([x1, x2])

# Define Objective Function
def branin(config):
    x1, x2 = config['x1'], config['x2']
    y = (x2 - 5.1 / (4 * np.pi ** 2) * x1 ** 2 + 5 / np.pi * x1 - 6) ** 2 \
        + 10 * (1 - 1 / (8 * np.pi)) * np.cos(x1) + 10
    return y

# Run
if __name__ == "__main__":
    opt = Optimizer(branin, space, max_runs=50, surrogate_type='auto')
    history = opt.run()

    # visualization
    print(history)
    history.plot_convergence(true_minimum=0.397887)
    plt.show()
    print(history.get_importance())
    history.visualize_jupyter()

```

图 6 OpenBox 样例程序代码。

图 6 展示了一个最基础的 OpenBox 使用样例代码，用户只需定义参数搜索空间和目标函数，即可调用 OpenBox 优化器 Optimizer 执行优化。Optimizer 设置 surrogate_type='auto'，执行算法自动选择。

优化过程中，OpenBox 打印运行中间结果，并通过进度条展示进度。优化结束后，OpenBox 展示最优参数，如图 7。

Parameters	Optimal Value
x1	-3.138277
x2	12.254526
Optimal Objective Value	0.398096578033325
Num Configs	50

图 7 OpenBox 运行结果示意。

OpenBox 还支持绘制优化过程曲线，如图 8。

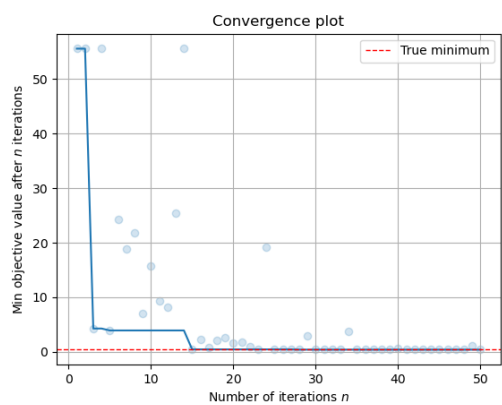


图 8 OpenBox 优化过程曲线示意。

OpenBox 可视化模块还可以对此次优化的参数、性能绘制 HiPlot 折线图，如图 9。

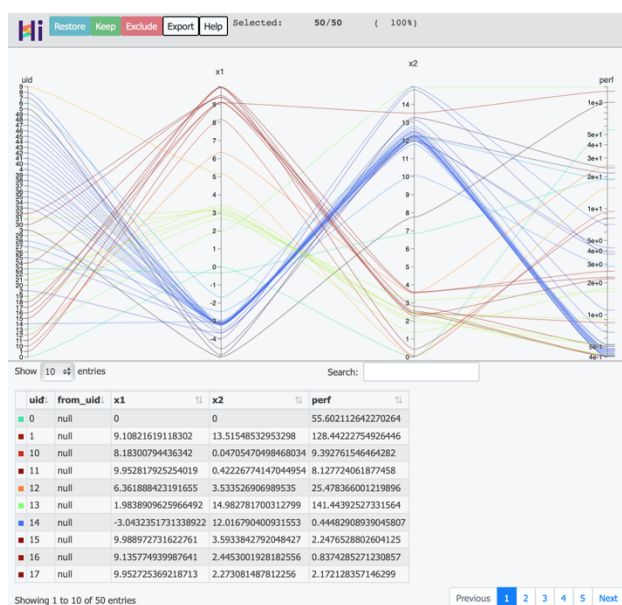


图 9 OpenBox 可视化 HiPlot 折线图示意。

OpenBox 支持对参数重要性进行分析，如图 10。

Parameters	Importance
learning_rate	0.293457
min_child_samples	0.101243
n_estimators	0.076895
num_leaves	0.069107
colsample_bytree	0.051856
subsample	0.010067
max_depth	0.000000

图 10 OpenBox 参数重要性分析示意。

OpenBox 在线服务支持用户通过邮箱注册账号，并登录网页控制台，在线监控并管理优化任务。用户可通过按钮快速启动、停止任务。在线管理界面如图 11：

Task Name	Configuration	Create Time	Status	Max_run	Operation
task_test	'hyperparameters': [{'name': 'x1', 'type': '...	2021-04-11 21	running	9/50	<div> <div>⏸</div> <div>⏹</div> <div>▶</div> </div>

图 11 OpenBox 在线服务网页控制台界面示意。

（二）性能实验结果

我们在广泛的黑盒优化问题上测试了 OpenBox 系统的性能。

图 12 为 4 个单目标黑盒数学问题上各系统性能，OpenBox 最优。

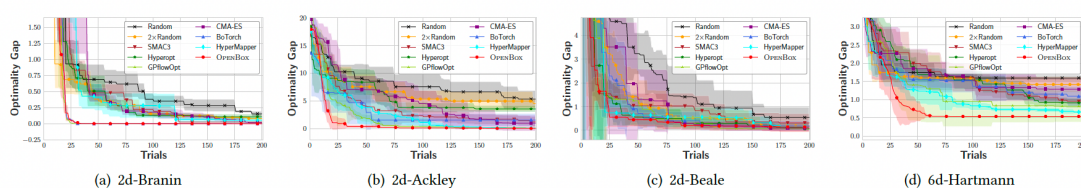


图 12 单目标数学函数优化实验结果。

图 13 为 4 个单目标带约束黑盒数学问题上各系统性能，OpenBox 最优。

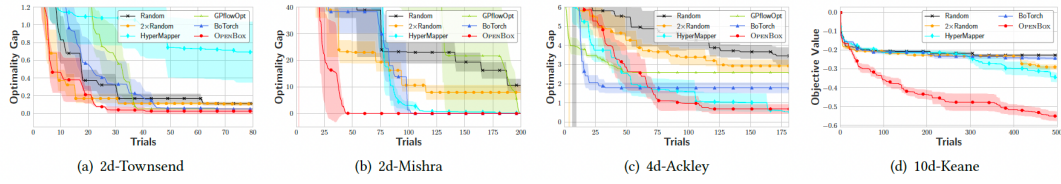


图 13 单目标带约束数学函数优化实验结果。

图 14 为 4 个多目标（不带、带约束）黑盒数学问题上各系统性能，OpenBox 最优。

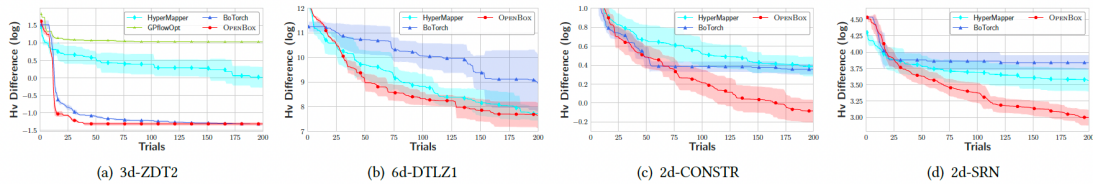


图 14 多目标不带约束（a, b）、带约束（c, d）数学函数优化实验结果。

图 15 为在 25 个数据集上，优化机器学习 LightGBM、SVM 模型超参数问题上各系统性能排名，OpenBox 平均排名第 1。

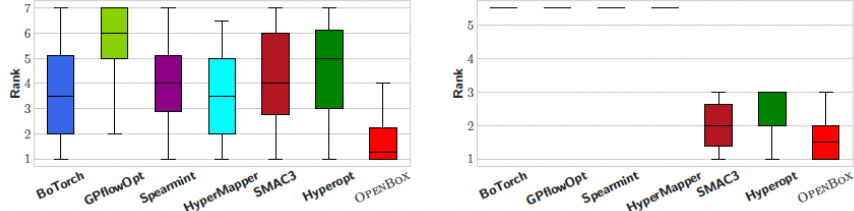


图 15 机器学习超参数优化问题实验结果。

图 16 展示了不同输入维度（4、8、16、32）下，各系统优化 Ackley 数学函数的性能。随着输入维度的增长，OpenBox 依然保持着最优性能。

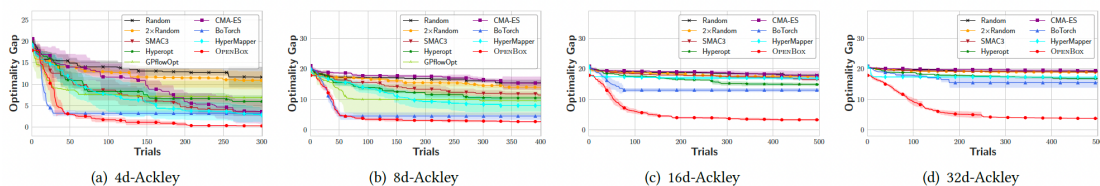
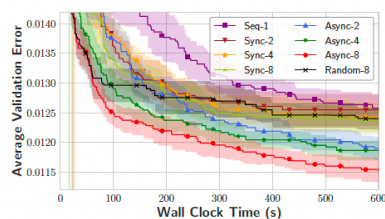


图 16 不同输入维度 Ackley 函数优化实验结果。

图 17 展示了 OpenBox 同步、异步并行策略不同 Worker 数量的性能。当增大并行 Worker 数量时，OpenBox 能获得较好的加速比。



- (3) 使用便捷，为用户提供本地 Python 包、在线服务等多种使用方式。
- (4) 相较已有系统，经实验验证，OpenBox 系统性能最优。
- (5) 有着完善的开源支持，包含在线文档、使用样例、测试套件、测试代码等。