arXiv:2603.01349v1 [physics.comp-ph] 2 Mar 2026

PHYSBO 更新：提升贝叶斯优化在物理学与材料研究中的可用性和可移植性

Yuichi Motoyama[a,*], Kazuyoshi Yoshimi[a], Tatsumi Aoyama[a], Kei Terayama[b], Koji Tsuda[c,d], Ryo Tamura[c,d,*]

[a]东京大学固体物理研究所，日本千叶 277-8581 [b]横滨市立大学生命医科学研究科，日本横滨 230-0045 [c]东京大学新领域创成科学研究科，日本千叶 277-8561 [d]材料基础研究中心，物质材料研究机构，日本茨城 305-0044

摘要

贝叶斯优化（BO）被广泛用于加速物理学和材料研究，在这些领域中，目标函数评估在计算或实验上的成本都很高。许多贝叶斯优化框架侧重于算法效率，但在真实研究环境中长期使用时，实用性和可移植性同样至关重要。PHYSBO 是一个贝叶斯优化库，旨在满足这些需求：它支持在用户定义的候选池上进行优化，并支持特定领域的问题设置。

本文介绍了 PHYSBO 第2版和第3版中引入的主要更新，重点在于可用性、可移植性和实际部署方面的改进，而非新的优化算法。在 PHYSBO 第2版中，软件许可证由 GPL 改为 MPL，以提高与更广泛的研究和软件生态系统的兼容性。在此修订基础上，PHYSBO 第3版引入了一系列面向实现的更新，旨在提高可用性和可移植性，而不修改核心优化算法。这些更新包括计算性能和可扩展性的改进、对多目标优化的扩展支持、面向连续变量优化引入基于范围的策略、移除紧密耦合的 Cython 模块等环境相关组件，以及与 NumPy 2 的兼容性。

这些改进减轻了用户的技术和组织负担，使 PHYSBO 能够部署于多样化的计算环境和研究工作流程中。通过在保持足够性能的同时强调可移植性和易于集成，PHYSBO 第3版被定位为面向物理学和材料科学中贝叶斯优化的可持续研究基础设施。

关键词： 贝叶斯优化、材料科学、物理模拟、多目标优化、连续优化、科研软件

程序概要/新版程序概要

程序标题：PHYSBO*CPC* 程序文件库链接：（由技术编辑补充）

*通讯作者。电子邮箱地址： y-motoyama@issp.u-tokyo.ac.jp（Motoyama）、tamura.ryo@nims.go.jp（Tamura）

开发者代码仓库链接：`https://github.com/issp-center-dev/PHYSBO`*Code Ocean* 胶囊：（由技术编辑补充）许可条款：MPL-2.0编程语言：Python补充材料：无上一版本的期刊参考文献：Y. Motoyama 等，Computer Physics Communications 278 (2022) 108405。新版本是否取代上一版本？：是

新版本的原因：
PHYSBO 第3版旨在提升其在实际研究环境中的可用性、可移植性和长期可持续性。尽管先前版本为离散候选空间提供了高效的贝叶斯优化，但实际使用表明，该软件在软件环境依赖、对连续优化和多目标优化的支持有限，以及妨碍其集成到更大型工作流程中的许可证约束等方面存在局限。新版本针对这些问题进行了改进，从而支持更广泛、更灵活的部署。

修订概要：
PHYSBO 第3版的主要更新包括提升计算性能和可扩展性、增强对多目标优化的支持，以及引入用于连续变量优化的基于范围的策略。紧密耦合的 Cython 模块等依赖环境的组件已被移除，并新增了与 NumPy 2 的兼容性。此外，软件许可证已从 GPL 更改为 MPL，以便更轻松地集成到多样化的研究工作流程中。问题性质（约 *50–250* 字）：

物理学和材料科学中的优化问题通常涉及代价高昂的目标函数评估，例如第一性原理计算、大规模模拟或实验测量。贝叶斯优化是一种强大的方法，可以减少为确定最优参数所需的评估次数。然而，在研究环境中实际应用贝叶斯优化仍面临多项挑战，包括扩展到大型搜索空间、处理多个相互竞争的目标、表示连续设计变量，以及实现软件环境兼容性。此外，研究人员通常还需要将领域特定约束和先验知识纳入优化过程，而通用优化框架并不总是支持这些需求。

解决方法（约 *50–250* 词）：
PHYSBO 提供了一个专为物理学和材料科学应用设计的贝叶斯优化框架。它支持在用户定义的候选池上进行优化，也支持基于范围的策略，从而能够直接对连续变量进行优化。PHYSBO 第3版扩展了对多目标优化的支持，采用标量化和非支配排序等成熟方法。该软件旨在兼顾计算效率与实际可用性，尽量减少依赖特定环境的组件，并确保与现代数值库兼容。这些设计选择使 PHYSBO 能够轻松集成到模拟和实验工作流中，同时为实际优化任务提供足够的性能。

其他意见，包括限制和特殊功能（约 *50–250* 词）：
PHYSBO 旨在作为科研库使用，并可导入更大型的软件框架和工作流。第3版采用 MPL 许可证，降低了使用库时可能遇到的法律障碍，并促进产学研合作。性能关键组件经过专门实现，以确保实际运行效率，同时优先考虑可移植性和安装便捷性。PHYSBO 不限制目标函数的形式，允许用户集成任意外部模拟或实验代码。

1. 引言

贝叶斯优化（BO）已成为加速材料和物理学研究的重要工具，因为目标函数评估往往耗时且成本高昂[cite]。典型应用包括第一性原理计算和大规模模拟中的参数优化，以及实验设计；在这些场景中，每次评估都需要付出大量计算或实验成本。在这类情况下，贝叶斯优化库的实际可用性与其算法性能同样重要。

PHYSBO 是一款专为物理学和材料科学应用打造的贝叶斯优化（BO）库[1]。PHYSBO 的一项显著特点是能够在预先定义的离散候选池中执行优化，使研究人员能够将化学直觉、物理约束和实验限制等领域特有知识无缝融入探索过程。虽然这一设计对物理学和材料科学尤为有利，但其底层框架并不局限于这些领域，也可广泛应用于各种科学与工程领域中的离散约束优化问题，尤其适用于评估成本高昂且先验知识至关重要的场景。这种灵活性使其区别于通用贝叶斯优化框架，后者往往难以整合这类细致入微的先验信息。此外，PHYSBO 的设计几乎无需设置参数，最大限度地减少了手动超参数调优的需求，从而确保其在各种现实问题中都能取得更好的性能。

该库在多个领域中的成功应用，证明了其实用价值，涵盖基于模拟的原子构型探索[2] 、下一代磁性材料的发现[3] 以及热超材料的设计[4]。其有效性也延伸至广泛的实验领域，尤其包括电催化剂[5]、超导材料[6]，、聚合物复合材料[7，8]，、药物发现[9]，、抗菌肽[10]，、量子化学计算[11]，、激子哈密顿量估算[12]，以及鱼饲料[13]的研发。除了这些具体应用之外，PHYSBO 还集成到了 2DMAT 等通用数据驱动框架中[14]，，用于在实验和数据分析工作流中高效优化参数。近年来，随着自驱动实验室[15, 16, 17] 这一范式日益兴起，PHYSBO 已被用作 NIMO 库的核心优化引擎[18]，，充当各类自主实验系统[19, 20, 21, 22]的“算法大脑”。此外，PHYSBO 还曾应用于 AIMHack2024 黑客马拉松等社区主导的活动。在这类活动中，来自材料科学、信息科学及相关领域的研究人员围绕数据驱动优化工作流开展协作项目，彰显了 PHYSBO 在传统研究环境之外的易用性和实用价值[23]。

然而，随着 PHYSBO 在实际研究环境中的应用日益广泛，其若干局限也逐渐显现。当前框架面临着重大的技术和战略挑战，包括严重依赖 Cython 等特定环境，以及在大规模多目标优化中存在可扩展性瓶颈。PHYSBO 最初以基于候选池的优化范式为核心，但如今越来越需要一种统一工具来处理连续设计变量，从而免去切换到 GPyOpt 或 Optuna 等其他软件所带来的操作负担。此外，与不断演进的数值库之间的兼容性问题，以及限制其在工业和协作环境中使用的严格许可条件，也制约了其长期可持续发展和更广泛的应用。PHYSBO 第3版旨在应对这些挑战，主要着眼于提升可用性、可移植性和实际部署能力，而非引入全新的优化算法。此次更新包括对

计算性能和可扩展性的改进、对多目标优化的增强支持、为连续变量优化引入基于范围的策略、移除与环境紧密耦合的组件（例如 Cython 模块）、兼容 NumPy 2，以及将软件许可证从 GPL 更改为 MPL。

本文概述了 PHYSBO 第3版（3.2.0 版于 2026年2月6日发布）的主要更新，并讨论这些更新对实际研究工作流程的影响。我们并非只关注算法细节，而是着重说明这些变化如何减轻用户负担，并让贝叶斯优化能够在现实中的物理学与材料研究中得到更灵活、更可持续的应用。

2. 许可证变更：从 GPL 到 MPL

软件许可在决定科研软件如何在实际研究环境中使用、集成和再分发方面发挥着重要作用。PHYSBO 第1版的早期版本采用 GNU通用公共许可证（GPL）发布。该许可证确保了强有力的著佐权保护，但在软件与其他代码或库结合使用时，可能会带来限制。尤其是，将 PHYSBO 作为库导入更大型的软件框架或工作流程时，使用 GPL许可软件可能会引发顾虑。在某些情况下，仅仅导入一个 GPL 许可的软件包就可能带来法律或组织方面的问题，特别是在涉及产业合作伙伴或混合许可证代码库的合作项目中。这些顾虑可能阻碍原本有用的科研软件得到采用，即使它仅作为依赖项而非独立应用程序使用。

为了解决这些问题，PHYSBO 2 版采用了 Mozilla公共许可证（MPL）。MPL 提供了一种文件级著佐权机制，既能确保对 PHYSBO 本身所做修改的开放性，又允许该库在采用不同许可证的更广泛软件生态系统中导入和使用。这一变更显著提升了 PHYSBO 与其他科学软件的兼容性，并降低了使用该库可能涉及的法律障碍。许可证的变更体现了 PHYSBO 设计理念的整体转变，即更加注重将其作为研究基础设施进行实用且可持续的部署。

3. 软件可移植性：消除环境依赖

在实际研究环境中，软件可移植性和安装便捷性是影响数值优化库可用性的关键因素。在 PHYSBO 的早期版本中，性能关键组件采用 Cython 实现，以加速基于高斯过程的计算。虽然这种方法降低了计算开销，却也带来了显著的环境依赖。特别是，构建 Cython 扩展需要特定的编译器设置和库版本，这常常导致其在某些平台上安装困难，包括 Windows 系统和异构的高性能计算环境。

在 PHYSBO 第3版中，核心库与基于 Cython 的实现之间这种紧密耦合已被移除。PHYSBO 的核心功能现在默认以纯 Python 包的形式提供，无需再执行特定于平台的编译步骤。因此，安装过程大为简化，该库也能在各种研究环境中更可靠地使用。此前使用 Cython 实现的性能关键部分，现已进一步改用 NunPy 算术运算进行修订，以保持相当的

执行时间。因此，这一设计变更不会导致典型贝叶斯优化工作流程的实际性能下降。在许多现实应用场景中，整体优化成本主要来自目标函数评估，而非代理模型更新，因此纯 Python 实现足以满足实际应用需求。

PHYSBO 第3版的另一项重要更新是兼容 NumPy 2.0 及后续版本。随着 Python 科学计算生态系统不断发展，保持与现代数值库的兼容性对于长期可持续发展至关重要。通过支持 NumPy 2.0，PHYSBO 无需依赖旧版配置，即可无缝集成到现代 Python 环境中。

4. 新的多目标优化方法

4.1. 多目标优化问题及其标量化

当需要同时优化多个目标函数时，通常不可能同时改善所有目标，因此它们之间不可避免地会出现权衡。这种权衡可用帕累托前沿的概念来描述。

设 $\boldsymbol{y}_i = (y_{i,1}, \ldots, y_{i,p})$ 和 $\boldsymbol{y}_j = (y_{j,1}, \ldots, y_{j,p})$ 分别表示两个候选点 $\boldsymbol{x}_i$ 和 $\boldsymbol{x}_j$ 的目标向量，其中假设所有目标均为最大化目标。若 $\boldsymbol{y}_i$ 支配 $\boldsymbol{y}_j$，则称

$$\forall a \; y_{i,a} \geq y_{j,a} \quad \text{and} \quad \exists a \; y_{i,a} > y_{j,a}. \tag{1}$$

如果某个候选解未被数据集中的任何其他解支配，则称其为帕累托最优。所有帕累托最优解构成帕累托前沿，它表示相互竞争的目标之间的最优权衡。给定帕累托前沿 $\{\boldsymbol{y}_i^{\mathrm{P}}\}$ 和参考点 $\boldsymbol{y}^{\mathrm{ref}}$，支配区域定义为若干超矩形的并集

$$\Omega_i = \prod_{a=1}^{p} \left[y_a^{\mathrm{ref}}, y_{i,a}^{\mathrm{P}} \right]. \tag{2}$$

支配区域的超体积通常用作性能指标，以定量评估帕累托前沿的质量。

在 PHYSBO 的先前版本中，会为每个目标分别构建一个代理模型，并通过最大化基于超体积的采集函数来选择候选点，例如超体积改进概率（HVVI）和期望超体积改进（EHVI）。然而，超体积计算的计算成本会随目标数量呈指数增长，这在实际应用中可能成为显著瓶颈。为解决这一问题，一种替代方法是将多个目标标量化为单一目标函数 z_i，并对该变换后的目标执行贝叶斯优化。PHYSBO 第3版基于该方法实现了另一个策略类 physbo.search.discrete_unified.Policy。该类与现有的多目标策略 physbo.search.discrete_multi.Policy 具有相似的接口。主要区别在于，discrete_unified.Policy.bayes_search 方法接受一个额外参数 unify_method，用于指定计算 z_i（由 $y_{i,a}$ 得到）的标量化策略。在当前版本的 PHYSBO 中，我们实现了两种此类标量化策略：ParEGO 和非支配排序（NDS）。

4.2. ParEGO

ParEGO（帕累托高效全局优化）是一种用于多目标贝叶斯优化的标量化方法，最初由 Knowles [24] 提出。其核心思想是通过随机标量化，将一个 p 目标优化问题转化为一系列单目标问题。

令 $\boldsymbol{y}_i = (y_{i,1}, y_{i,2}, \ldots\ldots, y_{i,p})$ 表示在设计点 $\boldsymbol{x}_i$处评估得到的目标值向量。ParEGO 根据 $\boldsymbol{y}_i$ 构造标量目标函数 z_i，如下所示：

$$z_i = \rho_{\mathrm{sum}} \sum_{a=1}^{p} w_a y_{i,a} + \rho_{\mathrm{max}} \max_a \left(w_a y_{i,a}\right), \tag{3}$$

其中，$\boldsymbol{w} = (w_1, \ldots\ldots, w_p)$ 是满足 $\sum_a w_a = 1$ 的权重向量；ρ_{sum} 和 ρ_{max} 分别是控制加权和项与加权切比雪夫项相对贡献的超参数。

在 PHYSBO 中，权重向量 $\boldsymbol{w}$ 会在每个贝叶斯优化步骤中随机采样，或由离散单纯形生成。每个目标$y_{i,a}$ 都会根据当前训练数据中观察到的最小值和最大值进行归一化，使所有目标的取值尺度具有可比性。随后，使用高斯过程回归对所得标量目标z_i 建模，并应用期望改进（EI）等标准采集函数。通过在各优化步骤中反复改变权重向量 $\boldsymbol{w}$，ParEGO 隐式探索不同目标之间的权衡，并能高效近似帕累托前沿，同时避免直接优化超体积改进等多目标采集函数所带来的高计算成本。

在 PHYSBO 第3版中，ParEGO 实现为 physbo.search.unify.ParEGO，可通过统一的多目标策略使用，而无需修改底层贝叶斯优化流程。

4.3. NDS

非支配排序（NDS）是多目标优化中的一项基础技术，用于根据帕累托支配关系对候选解进行分类。与依赖对目标值进行随机标量化的 ParEGO 不同，NDS 不会引入额外的标量化权重。相反，它直接利用目标向量之间的支配关系，因此尤其适用于多目标优化结果的后分析与解读。NDS 通过递归识别帕累托前沿，为每个解 r_i 分配一个等级i 。当前数据集中所有非支配点均被赋予等级$r_i = 1$，对应于帕累托最优解。移除这些点后，剩余集合中的非支配点被赋予等级$r_i = 2$。该过程以递增的等级值$r_i = 3$、4、.....反复执行，直到所有候选解都被穷尽地划分。通过这种方式，NDS 将解集分解为一系列帕累托前沿，并提供一种衡量候选解相对质量的序数度量。NDS 得到的等级会转换为标量目标值$z_i = 1/r_i$，并在后续的贝叶斯优化过程中最大化。在这一表述中，位于更优帕累托前沿上的解会被赋予更大的目标值，从而在保留帕累托支配信息的同时，将多目标优化转化为单目标问题。

在 PHYSBO 第3版中，NDS 以 physbo.search.unify.NDS 的形式实现，用户可以控制用于标量化的最大 - 秩。对于秩较大的点i ，z_i 将为 0。

4.4. 数值示例

接下来，我们将展示使用三个测试函数 VLMOP2、KYMN 和高斯函数，对 ParEGO 和 NDS 与现有方法 HVPI 进行比较的基准测试结果。[1] [2]

4.4.1. VLMOP2 基准

我们首先考虑 VLMOP2 基准问题[26, 27], ，该问题广泛用于评估多目标优化算法。在此问题中，解 $\boldsymbol{x} = (x_1, x_2) \in \mathbb{R}^2$ 由两个目标函数评估

$$y_1(\boldsymbol{x}) = 1 - \exp\left[-\sum_{i=1}^{2}\left(x_i - \frac{1}{\sqrt{2}}\right)^2\right], \tag{4}$$

$$y_2(\boldsymbol{x}) = 1 - \exp\left[-\sum_{i=1}^{2}\left(x_i + \frac{1}{\sqrt{2}}\right)^2\right]. \tag{5}$$

这两个目标函数分别在 $\boldsymbol{x} = (\pm 1/\sqrt{2}, \pm 1/\sqrt{2})$ 处达到最小值 0，且上界为 1。由于 PHYSBO 假定采用最大化设定，因此优化针对取反后的目标函数进行，

$$\tilde{\boldsymbol{y}}(\boldsymbol{x}) = (-y_1(\boldsymbol{x}), -y_2(\boldsymbol{x})),$$

从而较大的值对应帕累托改进解。图 1 展示了两个目标函数在二维设计空间 $[-2, 2]\times[-2, 2]$ 上的情况。每个目标函数都呈现一个平滑的单谷，其最大值位于 $(\pm 1/\sqrt{2}, \pm 1/\sqrt{2})$ 附近，且两个最优点在空间上彼此分离。因此，改善一个目标函数必然会使另一个目标函数变差，从而形成平滑且连续的帕累托前沿。

输入空间在二维区域 $[-2, 2] \times [-2, 2]$ 上离散化，并使用每个维度包含 101 个点的笛卡尔网格，共得到 10,201 个候选解。贝叶斯优化先进行 10 个随机采样步骤[3]，随后使用 HVPI、基于 NDS 的统一方法（参数为 $r_{\max} = 10$）或基于 ParEGO 的标量化方法（参数为 $\rho_{\max} = 1$ 和 $\rho_{\text{sum}} = 0.05$）进行 40 个优化步骤。图 2 对比了这三种方法得到的帕累托前沿。HVPI 获得了最大的支配超体积[4]（HV $= 0.3285$），表明它对平滑且连续的帕累托前沿实现了最广泛的覆盖。不过，这一表现是以最长的计算时间为代价的。相比之下，基于 NDS 的方法在获得相近超体积（HV $= 0.3092$）的同时，将计算时间缩短了一半以上。ParEGO 的计算时间最短，但其帕累托前沿较为稀疏，未能捕捉连续的权衡曲线，因此超体积显著较小（HV $= 0.1800$）。这些结果表明，对于帕累托前沿平滑且连续的问题，NDS 能够在解的质量与计算成本之间取得平衡，成为 HVPI 的高效替代方案；而 ParEGO 可能因其基于标量化的搜索策略而覆盖范围有限。

[1]本文中使用的所有脚本均可从ISSP数据存储库获取 [25]。[2]这些基准测试是在东京大学固体物理研究所的超级计算机系统C（kugui）上进行的（AMD EPYC 7763，2.45 GHz，64×2核）。[3]需要注意的是，三种方法使用了相同的随机样本。[4]支配区域的参考点为$\mathbf{y}^{\text{ref}} = (-1, -1)$。

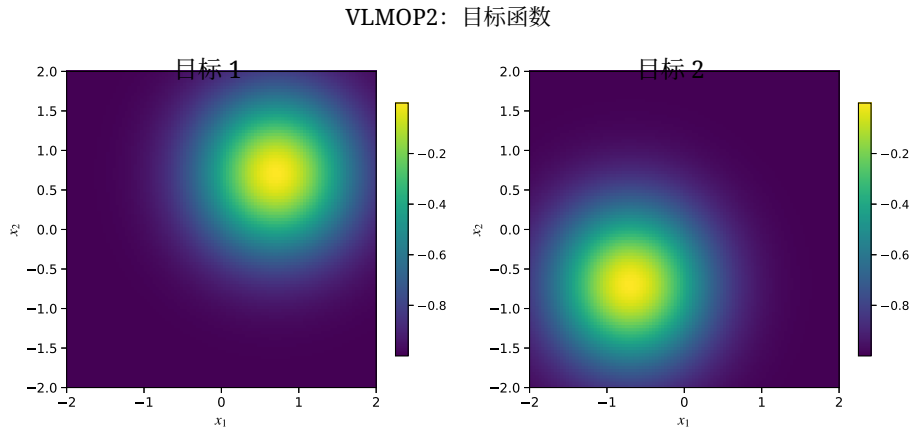


图 1：VLMOP2 基准的目标函数景观。左： $-y_1(\boldsymbol{x})$。右： $-y_2(\boldsymbol{x})$。两个目标在设计空间中的不同位置达到最大值，因此二者之间存在权衡。

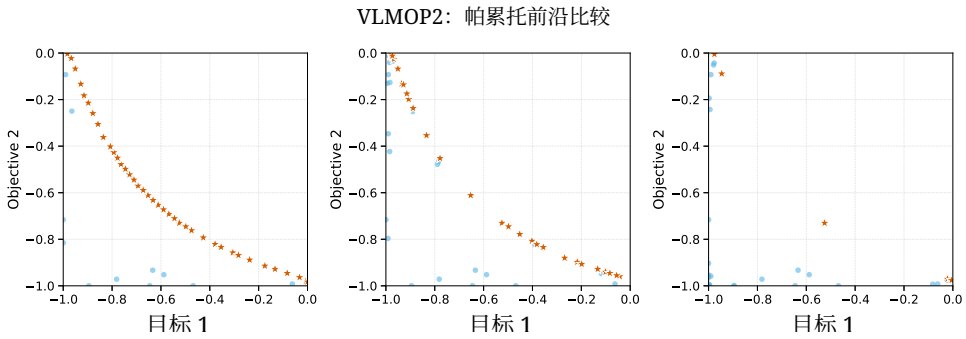


图 2：VLMOP2 基准的帕累托前沿比较。蓝色点表示已评估解，橙色标记表示非支配解。HVPI 对帕累托前沿的覆盖最为完整；NDS 以显著较低的计算成本提供了有竞争力的近似结果；ParEGO 则生成了一组稀疏的权衡解。

4.4.2. Kita–Yabumoto–Mori–Nishikawa 基准

接下来，我们考察 Kita–Yabumoto–Mori–Nishikawa（KYMN）基准问题[28]。这是一个约束双目标优化问题，最初提出该问题是为了在非线性可行性条件下测试多目标算法。在此问题中，一个解 $\boldsymbol{x} = (x_1, x_2) \in \mathbb{R}^2$ 由两个目标函数进行评估

$$y_1(x_1, x_2) = -x_1^2 + x_2^2, \tag{6}$$

$$y_2(x_1, x_2) = 0.5x_1 + x_2 + 1, \tag{7}$$

KitaYabumotoMoriNishikawa：目标函数

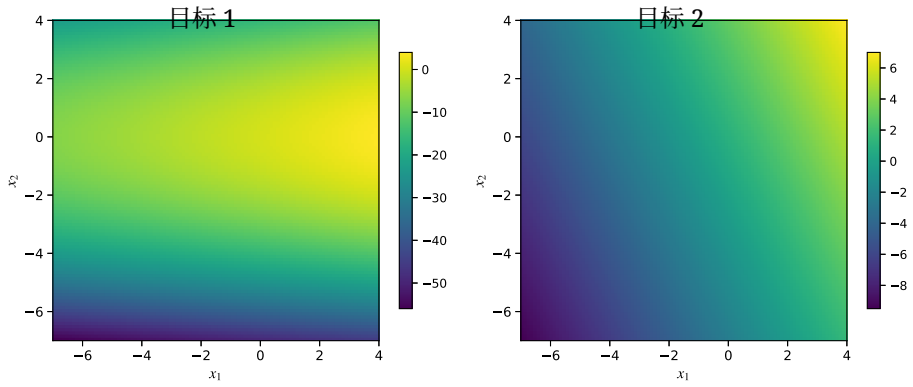


图 3：Kita–Yabumoto–Mori–Nishikawa（KYMN）基准的目标函数景观。左：$y_1(x_1,x_2)$。右：$y_2(x_1,x_2)$。可行域受到非线性不等式约束的隐式限制，从而形成高度受约束的优化景观。

受不等式约束

$$g_1(x_1, x_2) = 6.5 - \frac{x_1}{6} - x_2 \geq 0, \tag{8}$$

$$g_2(x_1, x_2) = 7.5 - \frac{x_1}{2} - x_2 \geq 0, \tag{9}$$

$$g_3(x_1, x_2) = 30 - 5x_1 - x_2 \geq 0. \tag{10}$$

所有目标均在这些约束所定义的可行域内最大化。图 3 展示了设计空间上的两个目标函数。与 VLMOP2 基准不同，两个目标函数在设计空间中沿不同方向几乎单调变化，且可行域受到约束的强烈限制。因此，可行帕累托最优解集在目标空间中占据一个狭窄而不规则的区域。

候选集通过在二维网格上离散化设计空间 $[-7, 4] \times [-7, 4]$ 生成，同时排除违反约束的点。贝叶斯优化采用与 VLMOP2 实验相同的方案，即先进行 10 个随机采样步骤，再基于 HVPI、基于 NDS 的统一方法或基于 ParEGO 的标量化进行 40 个优化步骤。图 4 比较了这三种方法得到的帕累托前沿。与 VLMOP2 平滑的帕累托前沿不同，KYMN 基准呈现出碎片化且高度非凸的帕累托前沿。在这种受约束的设置中，NDS 和 ParEGO 在支配超体积方面均优于 HVPI[5]。ParEGO 达到最大的超体积（HV = 975.9），NDS 紧随其后（HV = 970.8），而 HVPI 的数值较小（HV = 932.1）。从计算角度来看，NDS 和 ParEGO 的运行时间都不到 HVPI 的一半。这些结果表明，对于帕累托前沿不规则且非凸的受约束问题，NDS 和 ParEGO 等基于统一的方法，比直接采用基于超体积的优化能够实现更高效、更有效的探索。

[5]支配区域的参考点为$\boldsymbol{y}^{\mathrm{ref}} = (-56.0, -9.5)$。

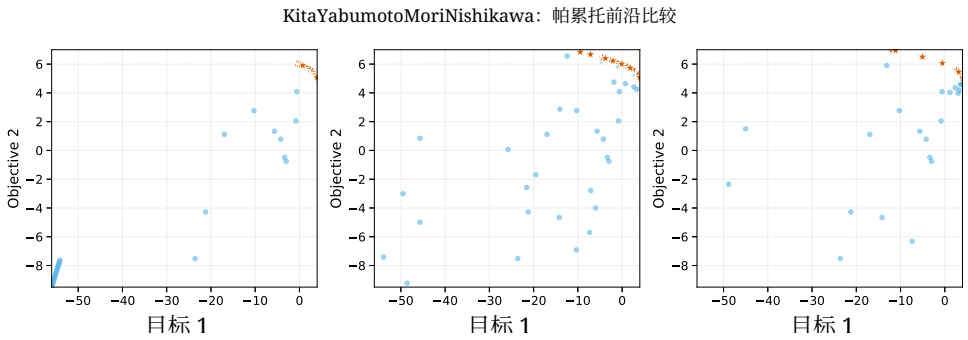


图 4：Kita–Yabumoto–Mori–Nishikawa 基准的帕累托前沿比较。蓝色点表示已评估解，橙色标记表示非支配解。ParEGO 将采样集中在狭窄的可行帕累托最优区域，而 NDS 则产生分布更广的近帕累托解。在这一约束场景中，HVPI 表现出更高的计算成本和更低的覆盖率。

VLMOP2 和 KYMN 基准共同凸显了目标函数景观的结构以及约束的存在如何显著影响多目标优化策略的相对性能。

4.4.3. 计算时间

最后，我们考虑一组高斯目标函数，以考察计算成本如何随目标数量变化。优化问题定义为

$$y_a(\boldsymbol{x}) = \exp\left[-\frac{\|\boldsymbol{x} - \boldsymbol{\mu}_a\|_2^2}{2\sigma_a^2}\right], \tag{11}$$

其中，$\mu_a = [\cos\ \theta_a, \sin\theta_a]^{\mathrm{T}}$ 表示中心，σ_a 表示第a个目标函数的宽度。在此基准测试中，中心和宽度随机生成为

$$\theta_a \sim U(0, 2\pi), \tag{12}$$
$$\sigma_a \sim U(0.5, 1.0), \tag{13}$$

其中，$U(l,h)$表示半开区间 $[l, h)$上的均匀分布。输入空间 $[-2, 2] \times [-2, 2]$ 被离散为 101×101 个网格点。

与之前的基准测试一样，我们首先执行 10 个随机采样步骤进行初始化，然后测量 40 个贝叶斯优化步骤所需的计算时间。为降低计算环境和随机种子的影响，我们进行了 10 次独立试验，并在每次试验中重新生成中心和宽度。为公平比较这三种方法，在每次试验中，所有方法使用相同的中心 $\{\mu_a\}$、宽度 $\{\sigma_a\}$以及在初始随机采样中选取的点 $\{\boldsymbol{x}_i\}$ 。

图5展示了10次独立试验中计算时间的均值和标准差。随着目标函数数量的增加，NDS和ParEGO的计算成本几乎保持不变，而HVPI的计算成本则呈指数增长。这种指数增长主要源于支配区域超体积的计算，而这是HVPI和EHVI方法所必需的。

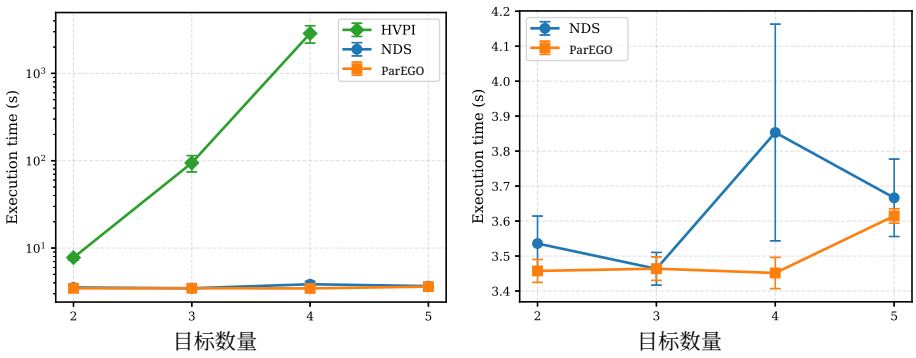


图 5：每种方法进行 40 次贝叶斯优化步骤时，计算时间与目标函数数量的关系p。图中展示了 10 次独立运行的均值和标准差。

5. 范围策略

5.1. 动机与设计理念

在物理学和材料科学中遇到的许多优化问题中，决策变量自然地表示为连续参数，而不是有限的预定义候选池中的元素。尽管PHYSBO的早期版本侧重于在离散候选集上进行优化，但当搜索空间是高维的或本质上连续时，这种方法会给用户带来显著负担。

为解决这一局限，PHYSBO 引入了范围策略，为探索由上下界定义的连续参数空间提供了统一框架。初始化范围策略时，需要明确提供这些边界，例如：

策略 = physbo.search.range.Policy(min_X=min_X, max_X=max_X)

其中，min_X 和 max_X 的维度相同，共同定义了连续参数空间中的一个超矩形区域。范围策略无需用户预先离散化搜索空间，而是允许直接根据参数范围来表述优化问题。这一设计显著降低了问题设置的复杂度，也更贴合模拟和实验参数的常见表述方式。

在贝叶斯优化中，下一个候选点通过最大化采集函数来确定，而采集函数的最大化本身构成一个辅助最大化问题，必须在每次迭代时求解。对于离散搜索空间，只需计算候选池中所有剩余点的采集函数值，并选出最大值对应的点，即可直接解决该问题。相比之下，对于连续搜索空间，则需要额外的数值优化求解器来最大化采集函数。

PHYSBO 中的 range.Policy.bayes_search 函数旨在接受用户指定的采集函数最大化优化程序，从而可以灵活选择优化策略。当前版本的 PHYSBO 提供了若干优化器，详见以下小节。例如，随机采样优化器的初始化方式如下：

优化器 =physbo.search.optimize.random.Optimizer(min_X, max_X, nsamples)

并作为传入 bayes_search 的

res =策略.贝叶斯_搜索(优化器=优化器)

除了单目标问题外，范围求解器的多目标问题版本也已实现为 physbo.search.range_multi.Policy 和 physbo.search.range_unified.Policy。

范围策略将搜索空间的定义与优化算法的选择解耦，从而提升了 PHYSBO 的可用性。用户只需定义一次参数范围，即可灵活应用不同的搜索策略，而无需重新构建工作流程。

5.2. 随机采样

在 PHYSBO 中，随机搜索是采集函数优化的默认策略：在预先定义的参数范围内均匀随机地采样固定数量的候选点，并选择采集函数值最大的点。默认情况下，用于采集函数优化的随机样本数设为 nsamples = 1000，以兼顾稳健性和计算成本。用户可通过指定随机优化器 physbo.search.optimizer.random.Optimizer，并将其传入 bayes_search 函数的 optimizer 关键字参数，来调整 nsamples 的值。这种可配置性使用户能够根据问题规模和可用计算资源，灵活权衡探索精度与计算开销。

作为范围策略下优化的初始化步骤，默认的探索策略是在指定的参数范围内进行随机采样。用户通过向量 min_X 和 max_X 为每个参数指定上下界，以此定义连续搜索空间。这些向量中的每个元素对应搜索空间的一个维度，并表示相关参数允许的最小值和最大值。

当目标函数的先验信息较少时，随机采样能够高效且无偏地进行初始探索，同时以最少的用户干预广泛覆盖搜索空间。这种选择符合实验设计中的常见做法：先进行初始随机采样或填充空间采样以收集信息，再应用更复杂的优化方法。

5.3. 另一个优化求解器库的封装：*ODAT-SE*

ODAT-SE（科学与工程开放数据分析工具） [29, 14] 是一个用于通过多种算法求解优化问题的 Python 框架：

- exchange：副本交换蒙特卡罗，适用于崎岖或多峰的景观 [30]。
- pamc：群体退火蒙特卡洛方法，能够高效探索复杂的参数空间 [31]。
- minsearch：Nelder–Mead 单纯形法，适用于光滑目标函数 [32,33]。
- mapper：基于网格的搜索，提供系统覆盖和高可重复性。
- 贝叶斯：基于代理建模和采集函数的贝叶斯优化。

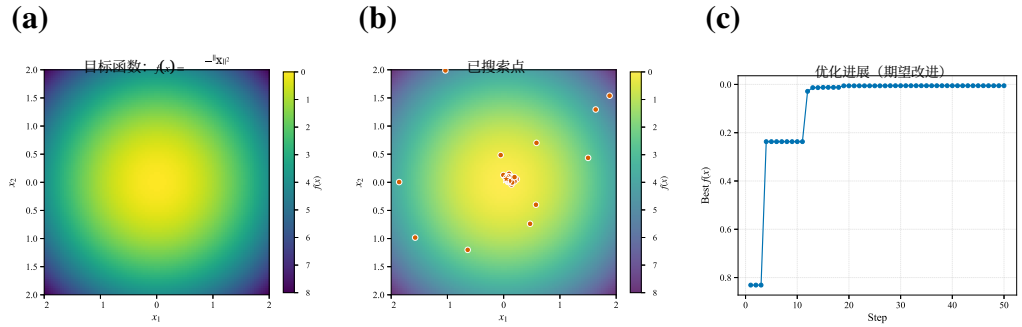


图 6：使用 PHYSBO 的范围策略在连续参数空间中进行贝叶斯优化。(a) $= -\|$ 上的目标函数景观 $f(\boldsymbol{x}) \|^2$。 (b) 优化过程中评估点的位置，显示评估点自适应地集中在最优点附近。(c) 使用期望改进（EI）采集函数，以迄今最佳目标值表示的优化进展。

PHYSBO 提供了一个将 ODAT-SE 封装在其中的优化器类，即 physbo.search.optimize.odatse.Optimizer。在使用 ODAT-SE 的实际工作流程中，优化算法通过变量 algorithm_name 进行明确指定，而算法的超参数（例如蒙特卡洛采样次数）则使用与 ODAT-SE 输入文件的 [algorithm] 部分等价的字典表示进行描述。PHYSBO 提供了一个工具函数 physbo.search.optimize.odatse.default_alg_dict，可为选定的算法生成包含默认超参数的模板字典。例如，可以获得表示副本交换蒙特卡洛方法默认设置的字典：

```
odatse_alg_dict = physbo.search.optimize.odatse.default_alg_dict(
min_X=min_X, max_X=max_X, algorithm_name="exchange" )
```

用户也可以根据需要修改参数。此设计只需修改 algorithm_name 的值，就能在不同优化策略之间无缝切换，同时保持搜索空间定义不变。准备好参数字典后，初始化优化器如下：

```
优化器 =physbo.search.optimize.odatse.Optimizer(odatse_alg_dict)
```

随后，按照与随机采样相同的方式，将其传递给范围策略。

5.4. 数值示例

我们将按照随附示例代码中的实现，演示如何使用 PHYSBO 的范围策略在连续参数空间中进行贝叶斯优化。作为一个简单而又具有说明性的示例，我们考虑对一个光滑的二维目标函数进行优化。

$$f(\boldsymbol{x}) = -\|\boldsymbol{x}\|^2, \qquad \boldsymbol{x} = (x_1, x_2) \in \mathbb{R}^2, \tag{14}$$

该函数在 $\boldsymbol{x} = (0, 0)$ 处有唯一的最大值，代表了物理学和材料科学中常见的典型单势阱能量景观。定义域$[-2, 2] \times [-2, 2]$ 上的目标函数景观如图 6 (a) 所示。函数值从原点向外平滑递减，为连续贝叶斯优化提供了一个表现良好的测试案例。

连续搜索空间由每个参数的下界和上界指定，并据此初始化范围策略。首先在指定的参数范围内进行随机采样，以开展初始探索。随后，使用期望改进（EI）采集函数进行贝叶斯优化，并通过在连续参数空间中随机采样来优化该函数。

图 6 (c) 展示了迄今找到的最佳目标值所反映的优化进展。在随机采样阶段经历初期的快速改进后，贝叶斯优化阶段高效地细化搜索，并迅速收敛至全局最优解 $f(\boldsymbol{x}) = 0$。这一表现说明，采集函数驱动搜索无需离散化，即可在连续空间中有效运行。图 6 (b) 展示了评估点在参数空间中的位置。初始随机样本分布较为广泛，而后续评估则集中在最优点附近。这表明贝叶斯优化如何自适应地将采样集中在连续搜索空间中较有希望的区域。总之，这些结果凸显了 PHYSBO 借助范围策略，直接在连续参数空间中执行高效贝叶斯优化的能力，无需显式离散化，并能在高维场景中实现可扩展的优化。

6. 结论

本文介绍了 PHYSBO 第3版的主要更新，特别强调了其在可用性、可移植性以及物理学和材料研究实际应用中的部署方面的改进。PHYSBO 第3版并未着重引入全新的优化算法，而是对该库进行了系统性改进，以更好地支持研究工作流程。

本文所述的更新解决了贝叶斯优化在实际应用中长期面临的若干挑战。计算性能和可扩展性的提升，使其能够在大规模优化问题中稳定使用。对多目标优化的支持得到增强，用户因而能够处理涉及相互竞争的目标的现实研究场景。引入基于范围的策略，将 PHYSBO 扩展至连续变量优化，提升了灵活性。此外，移除依赖特定环境的组件、兼容 NumPy 2 等现代数值库，以及转向限制更少的软件许可证，共同提升了可移植性、可持续性和集成便利性。

总的来说，这些变化使 PHYSBO 从一种专用优化工具转变为更通用的贝叶斯优化研究基础设施。通过降低技术和组织方面的门槛，PHYSBO 第3版使研究人员能够更轻松地将贝叶斯优化融入计算研究和实验研究，从而促进快速迭代、可重复性与协作。值得注意的是，移除 Cython 依赖显著提升了 Windows 用户的使用便利性。鉴于许多实验仪器运行于 Windows 环境，此次更新有助于推动贝叶斯优化从纯粹的网络空间模拟迈向现实世界实验研究。这些改进确保 PHYSBO 更易于融入现代人工智能与机器人研究，使其能够支持自动化和数据驱动科学发现的未来发展。

7. 致谢

我们谨向 Naoki Kawashima 和 Tsuyoshi Ueno 致以谢意，感谢他们参与了非常有益的讨论。. PHYSBO是在“材料科学软件易用性提升项目”的支持下开发的 -

[34] 由东京大学固体物理研究所于2020和2025财年实施。本研究部分获得了日本学术振兴会科研费（JSPS KAKENHI）资助，课题编号为25H01403和25K22013。此外，文部科学省（MEXT）通过数据创造与利用型材料研究开发项目（JPJ010337，资助编号：JPMXP1122683430和JPMXP1121467561）提供了额外支持。

参考文献

[1] Y. Motoyama, R. Tamura, K. Yoshimi, K. Terayama, T. Ueno, K. Tsuda, 贝叶斯优化软件包：Physbo, Computer Physics Communications 278 (2022) 108405. doi:10.1016/j.cpc.2022.108405. URL http://doi.org/10.1016/j.cpc.2022.108405[2] A. Kusaba, T. Kuboyama, K. Kawka, P. Kempisty, Y. Kangawa, PyAPX：用于原子构型模式探索的Python工具包（2025年）. arXiv:arXiv:2511.17972.[3] K. Nakamura, D. Furuya, Y. Miura, T. Yamazaki, A. L. Foggiatto, Y. Iwasaki, M. Kotsugi, 通过多目标贝叶斯优化高效发现具有大磁各向异性的热力学稳定材料, 《先进材料科学与技术：方法》 5 (1)（2025年4月）. doi:10.1080/27660400.2025.2485025. URL http://doi.org/10.1080/27660400.2025.2485025[4]W. Xi, Y.-J. Lee, S. Yu, Z. Chen, J. Shiomi, S.-K. Kim, R. Hu, 面向可见光-红外兼容伪装的热超材料超高效材料信息学逆向设计, 《自然·通讯》 14 (1)（2023年8月）. doi:10.1038/s41467-023-40350-6. URL http://doi.org/10.1038/s41467-023-40350-6[5] K. Sakaushi, W. Hoisang, R. Tamura, 通过人机协作加速发现含按需元素的有前景析氧电催化剂, 《ACS中心科学》 9 (12) (2023) 2216–2224. doi:10.1021/acscentsci.3c01009. URL http://doi.org/10.1021/acscentsci.3c01009[6] S. Yamazaki, T. Horide, Y. Yoshida, 借助迁移数据增强的贝叶斯优化加速$YBa_2Cu_3O_7$薄膜的工艺开发, 《ACS应用电子材料》 7 (14) （ 2025 年） 6582–6591. doi:10.1021/acsaelm.5c00884. URL http://doi.org/10.1021/acsaelm.5c00884B. Xu, T. A. Sultana, K. Kitai, J. Guo, T. Seki, R. Tamura, K. Tsuda, J. Shiomi, 面向“5G及未来”通信技术的聚合物复合材料实验在环交互优化, 《材料视界》 12 (10)（2025年）3332–3340. doi:10.1039/d4mh01606h. URL http://doi.org/10.1039/D4MH01606HY. Amamoto, C. Koganemaru, K. Kojio, A. Takahara, S. Yamamoto, K. Okazawa, Y. Tsuji, T. Aritake, K. Terayama, 一种用于设计并理解坚韧、可降解聚酰胺的机器学习方法, npj《计算材料》 11 (1)（2025年）198. doi:10.1038/s41524-025-01696-1. URL https://doi.org/10.1038/s41524-025-01696-1

[9] T. Yoshizawa, S. Ishida, T. Sato, M. Ohta, T. Honma, K. Terayama, 一种避免多目标分子设计中奖励操纵的数据驱动生成策略, 《自然·通讯》 16 (1)（2025年） 2409. doi:10.1038/s41467-025-57582-3. URL https://doi.org/10.1038/s41467-025-57582-3[10] Y. Murakami, S. Ishida, Y. Demizu, K. Terayama, 利用多目标贝叶斯优化设计含非蛋白源氨基酸的抗菌肽, 《数字发现》 2 (5) (2023) 1347–1353. doi:10.1039/D3DD00090G. URL https://doi.org/10.1039/D3DD00090G[11] K. Terayama, Y. Osaki, T. Fujita, R. Tamura, M. Naito, K. Tsuda, T. Matsui, M. Sumita, 由黑箱优化与数据驱动预测介导的、符合Koopmans 定理的有机分子长程校正（KTLC）密度泛函, 《化学理论与计算杂志》 19 (19) (2023) 6770–6781. doi:10.1021/acs.jctc.3c00764. URL https://doi.org/10.1021/acs.jctc.3c00764[12] W. Shimooka, H. Kitoh-Nishioka, S. Itoh, A. Kimura, 通过贝叶斯优化改进光合反应中心的激子哈密顿量, 《ACS欧米伽》 10 (24)（2025年）26090–26097. doi:10.1021/acsomega.5c03026. URL http://doi.org/10.1021/acsomega.5c03026[13] K. Terayama, S. Soma, H. Furuita, T. Shimizu, T. Iwasaki, K. Sakata, K.-i. Akagi, T. Asakura, T. Sanda, T. Yamaguchi, 等, 通过整合组学分析与两阶段贝叶斯优化实现可持续水产养殖的数据驱动饲料优化, bioRxiv（2025年）2025–01doi:10.1101/2025.01.27.635187. URL https://doi.org/10.1101/2025.01.27.635187[14] Y. Motoyama, K. Yoshimi, I. Mochizuki, H. Iwamoto, H. Ichinose, T. Hoshi, 数据分析软件框架2DMAT及其在二维材料结构实验测量中的应用, Computer Physics Communications 280 (2022) 108465. doi:10.1016/j.cpc.2022.108465. URL https://doi.org/10.1016/j.cpc.2022.108465[15] B. P. MacLeod, F. G. L. Parlane, T. D. Morrissey, F. Häse, L. M. Roch, K. E. Dettelbach, R. Moreira, L. P. E. Yunker, M. B. Rooney, J. R. Deeth, V. Lai, G. J. Ng, H. Situ, R. H. Zhang, M. S. Elliott, T. H. Haley, D. J. Dvorak, A. Aspuru-Guzik, J. E. Hein, C. P. Berlinguette, 用于加速发现薄膜材料的自驱动实验室, 《科学进展》 6 (20) (2020) eaaz8867. doi:10.1126/sciadv.aaz8867.[16] G. Tom, S. P. Schmid, S. G. Baird, Y. Cao, K. Darvish, H. Hao, S. Lo, S. Pablo-García, E. M. Rajaonson, M. Skreta, N. Yoshikawa, S. Corapi, G. D. Akkoc, F. Strieth-Kalthoff, M. Seifrid, A. Aspuru-Guzik, 面向化学与材料科学的自驱动实验室, 《化学评论》 124 (16) (2024) 9633–9732. doi:10.1021/acs.chemrev.4c00055. URL http://doi.org/10.1021/acs.chemrev.4c00055[17] N. Yoshikawa, Y. Asano, D. N. Futaba, K. Harada, T. Hitosugi, G. N. Kanda, S. Matsuda, Y. Nagata, K. Nagato, M. Naito, T. Natsume, K. Nishio, K. Ono, H. Ozaki, W. Shin, J. Shiomi, K. Shizume, K. Takahashi, S. Takeda, I. Takeuchi, R. Tamura, K. Tsuda, Y. Ushiku, 日本的自驱动实验室, 《数字发现》 4 (6)（2025年）1384–1403.

doi:10.1039/d4dd00387j. URL
http://doi.org/10.1039/D4DD00387J

[18] R. Tamura、K. Tsuda、S. Matsuda，NIMS-OS：一种用于在材料科学领域的人工智能与机器人实验之间实现闭环的自动化软件，《先进材料科学与技术：方法》3 (1)（2023年7月）。doi: 10.1080/27660400.2023.2232297。URL http://doi.org/10.1080/27660400.2023.2232297

[19] S. Matsuda、K. Nishioka、S. Nakanishi，通过对多组分电解液添加剂进行高通量组合筛选，提高锂金属二次电池的性能，《科学报告》9 (1)（2019年4月）。doi:10.1038/s41598-019-42766-x。URL http://doi.org/10.1038/s41598-019-42766-x

[20] S. Akiyama, H. Akitsu, R. Tamura, W. Matsuoka, S. Maeda, K. Tsuda, Y. Nagata，《离散反应参数的自主优化：通过铃木-宫浦反应对双官能底物进行单官能化》（2024年11月）。doi:10.26434/ chemrxiv-2024-bnj6p-v2。URL http://doi.org/10.26434/chemrxiv-2024-bnj6p-v2

[21] R. Tamura, H. Taketa, S. Murata, D. Ryuno, T. Yokota, K. Tsuda, S. Matsuda，《通过 NIMO 将旧式机器人系统无缝集成至自驱动实验室：液体处理器自动化案例研究》，《先进材料科学与技术：方法》5 (1)（2025年11月）。doi:10.1080/27660400.2025.2565144。URL http://doi.org/10.1080/27660400.2025.2565144

[22] R. Toyama, R. Tamura, S. Matsuda, Y. Iwasaki, Y. Sakuraba，《针对反常霍尔效应的成分梯度薄膜自主闭环探索》，npj《计算材料》11 (1)（2025年11月）。doi:10.1038/s41524-025-01828-7。URL http://doi.org/10.1038/s41524-025-01828-7

[23] T. Misawa, A. Koizumi, R. Tamura, K. Yoshimi，《探索生成式人工智能在数据驱动材料科学研究与教育中的应用》，《先进材料科学与技术：方法》5 (1) (2025) 2535956。doi:10.1080/27660400.2025. 2535956。URL https://doi.org/10.1080/27660400.2025.2535956[24] J. Knowles，《ParEGO：一种用于昂贵多目标优化问题的在线景观近似混合算法》，《IEEE进化计算汇刊》10 (1) (2006) 50–66。doi:10.1109/TEVC.2005.851274。[25] PHYSBO开发者，Physbo图库，https://isspns-gitlab.issp.u-tokyo.ac. jp/physbo-dev/physbo-gallery，访问日期：2026-02 (2026)。[26] C. M. Fonseca, P. J. Fleming，《多目标优化中进化算法概述》，《进化计算》3 (1) (1995) 1–16。doi:10.1162/evco.1995. 3.1.1。URL http://doi.org/10.1162/evco.1995.3.1.1

[27] D. A. van Veldhuizen, G. B. Lamont, 《多目标进化算法测试套件》，载于：1999年ACM应用计算研讨会论文集，SAC99，ACM，1999，页码 351–357。doi:10.1145/298151.298382。URL http://doi.org/10.1145/298151.298382[28] H. Kita, Y. Yabumoto, N. Mori, Y. Nishikawa, 《利用热力学遗传算法进行多目标优化》，载于：H.-M. Voigt, W. Ebeling, I. Rechenberg, H.-P. Schwefel（编），《自然界并行问题求解——PPSN IV》，Springer 柏林海德堡，柏林，海德堡，1996，页码 504–512。doi:10.1007/3-540-61723-x_1014. URL http://doi.org/10.1007/3-540-61723-X_1014[29] ODAT-SE GitHub代码库，https://github.com/issp-center-dev/ODAT-SE, 访问日期：2025-12-24。[30] K. Hukushima, K. Nemoto, 《交换蒙特卡洛方法及其在自旋玻璃模拟中的应用》，《日本物理学会杂志》65 (6) (1996) 1604–1608。doi:10.1143/JPSJ.65.1604。[31] K. Hukushima, Y. Iba, 《种群退火及其在自旋玻璃中的应用》，载于：《AIP会议论文集》第690卷，美国物理联合会，2003，页码 200–206。doi: 10.1063/1.1632130。[32] J. A. Nelder, R. Mead, 《用于函数最小化的单纯形法》，《计算机学报》7 (4) (1965) 308–313。doi:10.1093/comjnl/7.4.308。[33] M. H. Wright, 《直接搜索方法：曾遭轻视，如今备受认可》，载于：D. F. Griffiths, G. A. Watson（编），《数值分析1995》，Pitman数学研究札记丛书第344卷，Addison-Wesley，1996，页码 191–208。[34] K. Yoshimi, Y. Motoyama, T. Aoyama, M. Kawamura, N. Kawashima, 《推进材料科学软件可用性的项目》，《先进材料科学与技术：方法》5 (1) (2025) 2564055。doi:10.1080/27660400.2025.2564055。URL https://doi.org/10.1080/27660400.2025.2564055