量化 PSO-*X* 框架中模块及其交互作用的影响

Christian L. Camacho- Villalón、Ana Nikolikj、Katharina Dost、Eva Tuba、Sašo Džeroski 和 Tome Eftimo v

arXiv:2601.04100v1 [cs.NE] 7 Jan 2026

摘要——**PSO-*X* 框架整合了数十种模块，这些模块旨在利用粒子群优化求解单目标连续优化问题。模块化框架使用户能够自动生成并配置针对特定优化问题量身定制的算法，但随着框架中模块数量以及为模块交互定义的自由度增加，这一过程的复杂性也随之上升。了解模块如何影响算法在不同问题上的性能，对于提高寻找有效实现方案的效率、确定值得进一步研究的方向至关重要。尽管模块化优化框架具有实际应用价值和科学意义，但目前仍缺乏实证研究来探讨哪些模块最为重要，以及这些模块如何相互作用。本文分析了从 PSO-*X*框架实例化的 1,424 种粒子群优化算法在 CEC'05 基准测试套件中的 25 个函数上的表现，测试维度为 10 和 30。我们使用函数方差分析，量化各模块及其组合对不同问题类别中性能的影响。实际应用中，这使我们能够根据多峰性、数学变换和维度变化等问题特征，识别对 PSO-*X* 性能影响较大的模块。随后，我们进行聚类分析，以识别具有相似模块效应模式的问题类别组。结果显示，所有问题类别中模块重要性的变异性都较低，这表明粒子群优化性能主要由少数几个影响力较大的模块决定。**

索引词——**模块重要性、函数方差分析、基准测试**

I 引言

连续 CONTINUOUS 优化问题广泛存在于许多领域和定义域中。它们涵盖的范围很广，从确定能够使系统（例如模拟器）达到预期性能的参数值，到设计符合安全和性能标准的结构（例如汽车底盘）。虽然有些连续优化问题可以使用精确方法 求解（例如，二阶导数法或牛顿法的方法），许多其他问题具有复杂特征，使其不适合采用精确方法求解。具有复杂特征的连续优化问题的典型示例包括具有多峰、不可微景观的问题；维度数量庞大的问题；以及目标函数没有显式表达式的问题 [1, 2]。解决困难连续优化问题的一种有效替代方法是使用 元启发式算法，例如进化策略（ESs） [3, 4],差分进化（DE） [5], 和粒子群优化（PSO） [6, 7]。与精确方法不同，元启发式算法是无导数优化技术，通过从搜索空间中迭代采样新的候选解来近似问题的最优值。

尽管元启发式算法几乎无需调整即可投入使用，但研究表明，仔细选择实现中使用的算法组件并微调其参数值，可以显著提升性能 [8, 9, 10]。因此，大量研究致力于通过手动调整各种元启发式算法来提升性能，而系统探究为何某些算法能在特定问题类别上取得良好结果、却无法在其他类别上奏效的研究则少得多 [11, 12]。近期研究通过投入更多精力开展理论和实验研究，为算法性能提供有用的解释，从而弥补了这一不足（例如，参见 [13]）。此外，自动算法配置方面的发展，以及机器学习（ML）在分析算法性能数据方面的广泛应用，使得针对特定问题调整元启发式算法实现变得越来越高效，也使得人们能够对不同算法变体进行公平且可复现的比较，以评估它们在特定场景下的优缺点 [14, 15]。近年来，研究人员也开始深入考察如何利用自动化方法，从可复用算法组件中构建高性能实现，同时致力于应对创建全自动算法流水线所面临的挑战 [16]。

尽管该领域不断取得进展，如今也有许多工具可用于实现和研究元启发式算法，但文献中数量极其庞大的算法变体，与能够明确指导针对哪些问题、在何种条件下哪些算法效果最佳的实验和理论研究数量相对较少之间，仍存在不匹配。选择一种优化算法可能颇具挑战性，尤其是对于缺乏经验的用户，以及面对与以往遇到的问题差异很大的问题的用户而言。此外，如果所选方法未能取得预期结果，用户还将面临大量设计方案，不知如何选择以提升实现效果。

我们的贡献：本文揭示了.....的重要性

本研究获得欧盟"地平线欧洲"研究与创新计划资助，资助项目为玛丽·斯克沃多夫斯卡-居里 COFUND 博士后计划，资助协议编号为 101081355-SMASH；同时还获得斯洛文尼亚共和国和欧盟欧洲区域发展基金的资助。我们还衷心感谢斯洛文尼亚研究与创新署通过计划资助项目 P2-0098、项目资助项目 J2-4460 和 GC-0001，以及授予 Ana Nikolikj 的青年研究人员资助项目 PR-12897 提供的经费支持。我们感谢 EC/EuroHPC JU 以及斯洛文尼亚高等教育、科学与创新部通过 SLAIF 项目（资助编号 101254461）提供的支持。

C.L. Camacho-Villalón 和 S. Džeroski 隶属于知识技术系。A. Nikolikj、T. Eftimov 和 E. Tuba 隶属于计算机系统系。这两个系均为位于斯洛文尼亚卢布尔雅那的约瑟夫·斯特凡研究所下属部门。电子邮箱：christian.camacho.villalon@ijs.si。K. Dost 隶属于新西兰坎特伯雷大学数学与统计学院。数字对象标识符 XXXXX.YYYYY

以及 PSO 中基础算法组件之间的交互作用，旨在帮助用户选择并调整 PSO 算法变体。为此，我们研究了 PSO-*X*框架 [17], ——一种粒子群优化的模块化实现，包含 11 个模块和 58 种实现选项，涵盖了 25 年以上的 PSO 研究。我们利用通过 PSO-*X*生成的 1,424 个 PSO 变体的性能数据，应用方差函数分析（f-ANOVA） [18, 19] ，量化各个组件及其交互作用对 CEC'05"单目标实参数优化特别会议"中 25 个函数性能的影响 [20]。本文补充了先前关于模块化优化框架中模块重要性的数据驱动研究，填补了文献中的一项重要空白，因为 PSO-*X* 迄今尚未纳入此类研究。

与传统的模块性能评估不同，传统评估侧重于单个模块或有限数量的模块组合的影响，而忽略更广泛的交互作用；本研究则考察了成对和三重交互作用。我们的结果表明，粒子群优化的性能在很大程度上取决于 omega1CS、randomMatrix 和 DNPP 模块的存在及其交互作用，不过后者的影响较弱。其他模块，例如 accelCoeffCS、拓扑结构和影响模型——尽管它们受到大量研究 [21, 22, 23]甚至引发争议 [24]——对粒子群优化-*X* 算法性能的贡献却微乎其微。

我们还研究了不同问题类别如何依据模块效应聚类，以确定高层次问题类别的特征（如多峰性、可分性和数学变换）与具体实现中所用模块之间的关系。我们发现，在大多数函数上，对粒子群优化-*X* 性能影响最大的模块是 randomMatrix，而在少数情况下则由 omega1CS 起主导作用。单个模块的效应在 10 维问题上最强，在 30 维问题上略有减弱。随着维度增加，randomMatrix 和 omega1CS 的单独影响效应逐渐减弱，而涉及 DNPP 模块的成对交互作用变得更加重要；该模块能够平衡这两个模块的作用。

本文其余部分安排如下：第二节回顾背景和相关工作；第三节介绍我们的模块重要性方法；第四节详述实验设置；第五节报告结果；第六节总结全文，并讨论局限性和未来工作。

II. 背景与相关工作

本文研究单目标连续优化问题（COP）。在单目标 COP 中，目标是找到一个实值向量 $\vec{x}^* \in \mathbb{R}^D$ ，使目标函数 f 最小化： $\mathbb{R}^D \to \mathbb{R}$，即$\vec{x}^* = \arg\min_{\vec{x}\in\mathbb{R}^D} f(\vec{x})$，其中 D 表示决策空间的维度。我们还考虑 $f(\cdot)$ 为非线性、多峰且可能不可分的情形，这使得通过解析方法计算梯度或闭式解变得不可行。元启发式算法是解决这类问题的首选方法。

A. 自动算法设计

元启发式算法的新变体通常以渐进的方式提出，即每次提出一个或少数几个，其设计是由算法设计者的直觉和专业知识指导的手动过程的结果。这种算法设计方法虽然在少数情况下取得了成功，但往往带有主观性、耗时且容易出错。为解决这些问题，自动设计方法利用了基于组件的优化框架的发展以及自动算法配置 [25]方面的进步。基于组件的优化框架（如 modCMAES [26], modDE [27], PSO-*X* [17]和 METAFO$\mathbb{R}$ [28]）为用户提供了一种灵活的方式，使其能够从离散的算法组件（模块）集合中生成许多不同的算法实现；这些组件可在算法设计中相互替换，并负责特定行为。另一方面，自动算法配置工具（AACT），例如 irace [29], ParamILS [30], 和 SMAC [31],，负责尝试算法组件的不同组合，并评估它们在不同问题上的性能。

B. PSO-X 框架

PSO-*X* 框架 [17]是粒子群优化算法的一种基于组件（即模块化）实现 [6, 7]。该框架整合了受多种先进 PSO 变体启发的广泛实现选项，并将关键设计选择转化为不同模块。PSO-*X* 架构将算法分解为可互换的模块，例如用于控制初始化、速度更新规则、位置更新规则、拓扑结构和影响控制策略模型等的模块。PSO-*X* 的核心是广义速度更新规则，该规则统一并扩展了多种 PSO 变体，可纳入或排除惯性、加速系数、扰动策略、随机矩阵、基于角度的旋转等不同组件。这一形式化方法使得在统一的算法模板下，能够灵活设计经典和新型 PSO 动力学。此外，将 PSO-*X* 与 AACT 配合使用，可以系统地探索不同设计方案，并评估各个组件及其交互作用对算法在不同优化问题上的影响。表 I 总结了 PSO-*X*框架实现的主要算法组件。有关 PSO-*X*的更多详情，请参阅文献 [32, 17]。下文中，我们使用无衬线字体表示 PSO-*X* 框架中实现的模块及其可用选项。

C. 方差函数分析

方差函数分析 (f-ANOVA) [18, 19]是一种方差分解技术，最初用于机器学习中的超参数优化。f-ANOVA 用于说明每个因素及因素组合对一组实验中结果变异性的贡献程度。给定许多

表 I 在 PSO-X 框架中实现的主要算法组件。每个组件定义了算法模板中的一个模块维度，并可与其他组件组合，以实例化完整的 PSO 变体。

算法组件	描述
广义速度更新规则（GVUR）	统一粒子群优化各类公式的核心运动学规则。粒子的位置（$\vec{x}^i$）代表优化问题的解，其更新方式为 $\vec{x}^i_{t+1} = \vec{x}^i_t + \vec{v}^i_{t+1}$。速度向量（$\vec{v}^i_{t+1}$）的计算使用如下定义的 GVUR: $\vec{v}^i_{t+1} = \omega_1 \vec{v}^i_t + \omega_2$ DNPP$(i,\ t) + \omega_3$ Pert$_{\text{rand}}(i,\ t)$，，其中 $\boldsymbol{v}^i_t$ 为惯性项，DNPP 和 Pert$_{\text{rand}}$ 是两个模块化组件，而 ω_1、 ω_2和 ω_3 是三个实值参数。
DNPP	下一可能位置分布（DNPP）定义了用于计算粒子位移的几何分布，影响旋转不变性和探索幅度。例如，DNPP-矩形采用超立方体分布，DNPP-球形使用超球面分布，DNPP-高斯则使用多元高斯分布。
扰动$_{\text{rand}}$ (perturbation2CS)	独立于群体信息和个体信息，应用可选的非信息性随机扰动，以增加探索。两种主要实现选项是扰动$_{\text{rand}}$-矩形和扰动$_{\text{rand}}$-含噪，二者均从随机均匀分布中采样。
扰动$_{\text{info}}$ (perturbation1CS)	在 DNPP 组件内，围绕感兴趣的方向施加可选的信息性扰动，以增强开发专注于对已知解进行局部细化。一些实现选项包括 Pert$_{\text{info}}$-高斯、Pert$_{\text{info}}$-Lévy 和 Pert$_{\text{info}}$-均匀分布。
PM	定义 Pert$_{\text{rand}}$ 和 Pert$_{\text{info}}$ 使用的扰动幅度（PM）。主要选项包括 PM-欧氏距离和 PM-目标函数距离，它们根据空间距离或适应度改进调整扰动；以及 PM 成功率，它根据成功扰动的比率增大/减小幅度。这些策略仅在启用 Pert$_{\text{info}}$ 时生效或启用 Pert$_{\text{rand}}$ 模块时生效。
omega1CS	控制 GVUR 中粒子的惯性（ω_1）。其值可通过多种策略计算，从随时间变化的策略（例如 IW 线性递减）到自适应更新策略（例如 IW-基于速度自适应）。
accelCoeffCS（AC）	用于控制认知与社会加速系数（φ_1, φ_2）。该组件在自我强化与协作之间取得平衡。主要选项包括 AC 外推和 AC 时变，可根据迭代次数和质量差异动态调整系数。
randomMatrix（Mtx）	该组件仅适用于 DNPP-矩形或 DNPP-球形模块。它会生成随机变换矩阵，用于旋转或缩放位移向量，例如 Mtx 随机对角和 Mtx 欧几里得旋转（角度为 α 遵循恒定、高斯或自适应调度），以及渐进对角化的分组矩阵，例如 Mtx-Increasing 基于群体。
拓扑结构（Top）	定义控制信息交换的邻域结构 $\mathcal{N}_i$。不同拓扑结构通过连通性来平衡探索与开发，其中环形拓扑的连通性最低，全连接拓扑的连通性最高。
影响模型 (MoI)	指定从 $\mathcal{N}_i$ 中选择哪些解，以及如何为它们对粒子运动的影响加权，例如 MoI-best-of-neighborhood、MoI-fully informed（取平均）、MoI-ranked fully informed 或随机告知者影响模型。
种群（Pop）	控制群体规模动态和初始化。例如，随时间变化种群策略和增量种群策略会根据搜索进度增加或移除粒子。本文考虑两种初始化方案：随机初始化，采用随机采样；以及水平初始化结合随机采样与水平学习，朝向全局最优进行搜索。

种配置的观测性能值，f-ANOVA 会对所有其他因素进行边际化，以估算单个因素的效应（主效应）以及高阶交互作用（两两组合、三项组合等）。由此得到的效应贡献具有可加性（存在估计误差），其总和等于观测总方差，从而得出可解释的重要性分数，依据解释能力对因素和交互作用进行排序。要应用 f-ANOVA 技术，必须提供一张设计表，其中包含每次运行涉及的因素及其相应水平，以及对应的响应值。随后，f-ANOVA 会确定相对贡献，从而识别主导驱动因素，以及协同或拮抗交互作用和情境依赖性（例如，按数据集或任务区分）。

D. 相关工作

以往关于模块化优化框架性能的研究，主要通过分析表现最佳的配置来研究模块重要性，通常依赖于所选组件的频数统计。例如，在 modCMAES [26] 和 modDE [27], 的研究中，使用自动配置工具 irace 识别“精英”配置，并根据某个模块在这些精英配置中的出现频率评估其重要性。为捕捉交互效应，研究人员逐步将模块加入配置空间，并观察精英配置频率的变化。这类研究只能粗略地反映模块的贡献，且没有系统地比较模块在不同问题类别中的重要性。

优化和机器学习领域已提出多种互补技术，用于评估超参数和模块的重要性。这些技术包括前向选择方法 [33], 性能影响模型 [34],消融分析 [35],以及函数ANOVA [18, 19],，各自提供了不同的视角。例如，消融会逐次禁用或替换一个组件，以衡量性能下降；而 f-ANOVA 则将性能方差分解到各算法、组件及其组合上。

研究人员还探索了用于根据问题特征预测有效模块选择的学习模型。在 [36],Prager 等人采用分类器链方法，根据景观特征预测每个模块的最优设置，从而隐式地对模块之间的依赖关系进行建模，但没有量化哪些交互作用最为重要。类似地，在[37, 38],Kostovska 等人训练了独立的机器学习模型，根据问题特征值为 modCMAES 的每个模块推荐最佳组件。尽管这些方法

承认问题性质会影响模块效能，但它们通常将模块独立处理，因此忽略了明确的模块间交互效应。在一项近期研究中，[39], Van Stein 等人将可解释人工智能技术应用于模块化算法，但仅关注单个模块的重要性，而忽略了高阶交互作用。

迄今为止，唯一明确量化模块化优化框架中各个模块及其组合的方差贡献的工作，是 [40, 41]中所报告的研究。在 [40], , Nikolikj等人将 f-ANOVA 应用于 modCMAES 和 modDE 算法变体，以测量主效应以及两两/三元交互效应；在 [41], , 他们研究了具有相似模块交互模式的问题类别簇，与基于高层次问题特征构建的问题类别簇之间的一致性。到目前为止，尚未针对不同问题景观下的粒子群优化-*X* 框架开展可比的方差分解分析。我们的工作填补了这一空白，首次针对粒子群优化提供了特定于问题的模块重要性与交互分析。

III. 量化模块效应的方法论

为了分析粒子群优化-*X* 模块在不同问题景观中的作用，我们采用了一种基于 f-ANOVA 的两步方法论。首先，对于每个问题类别（见第 IV-A 节），我们构建一个数据集，其中包含在该函数上评估的所有粒子群优化-*X* 算法变体。每个算法变体均以其模块选择作为分类特征进行编码，并以其在该函数上的性能作为目标变量。随后，我们对每个数据集应用 f-ANOVA，以量化性能方差中分别由单个模块（个体效应）、每对模块（两两交互效应）以及每组三个模块（三元交互效应）所解释的比例。由此，对于每个问题类别，都会得到一个维度为 $n+\binom{n}{2}+\binom{n}{3}=92$ 的效应贡献向量（粒子群优化-*X*中包含 $n=8$ 个模块）。需要注意的是，尽管粒子群优化-*X* 总共包含11个模块，但为了控制算法变体的数量，我们固定了以下参数：GVUR中的 $\omega_2=1$ 和 $\omega_3=1$、PM $=0.5$，两个 $\text{Pert}_{\text{rand}}$ 和 $\text{Pert}_{\text{info}}$中的参数，以及 Pop 中采用 pop $_{size}=20$的 Pop-constant。

A. 用于模块交互作用分析的 *f-ANOVA*

鉴于 PSO-*X* 可通过不同模块（如速度更新、拓扑结构、惯性控制、扰动策略）的组合实例化出大量算法配置，f-ANOVA 使我们能够剖析各模块对观测到的算法性能的影响。令 $g(\vec{m})$ 表示由模块向量 $\vec{m}=(m_1,\ldots,m_q)$ 定义的 PSO-*X* 配置的期望性能，其中每个 m_i 代表一个用于控制特定模块或实现选项的类别型或数值型超参数。假设 $g(\vec{m})$ 可平方积分，则可将其分解为主效应与交互效应之和，如下所示：

$$g(\vec{m})=g_0+\sum_{i=1}^{q}g_i(m_i)+\sum_{i<j}g_{i,j}(m_i,m_j)+\ldots+g_{1,\ldots,q}(m_1,\ldots,m_q), \quad (1)$$

其中，g_0 表示所有配置的平均性能，$g_i(m_i)$ 表示模块 i 的主效应，而高阶项$g_{i,j}(\cdot)g_{i,j,k}(\cdot)$, $g_{i,\ldots,q}(\cdot)$ 表示模块或参数组之间的交互效应。同理，$g(\vec{m})$ 的总方差可表示为：

$$\mathbb{V}[g(\vec{m})]=\sum_{U\subseteq\{1,\ldots,q\}}\mathbb{V}_U, \quad (2)$$

其中，$\mathbb{V}_U=\text{Var}[g_U(\vec{m}_U)]$ 表示子集 U （即一组模块）的方差贡献。随后，U 的相对重要性定义为：

$$\mathbb{I}_U=\frac{\mathbb{V}_U}{\mathbb{V}[g(\vec{m})]}, \quad (3)$$

其中 $\sum_U\mathbb{I}_U=1$。$\mathbb{I}_U$ 的值越高，表明相应模块或模块组合对优化性能的影响越大。

直接计算这些方差分量并不可行，因为 PSO-*X* 包含一个高维组合设计空间，其中既有大量类别型模块选择，也有连续控制参数。作为替代方案，我们根据多种 PSO-*X* 配置获得的实证结果训练一个替代回归模型。遵循 Hutter 等人 [18], 的方法，我们采用随机森林回归器来近似性能函数，并高效估计每个模块及其交互作用的边际贡献。估计得到的方差分量及其归一化重要性，可用于量化每个 PSO-*X* 模块（及其与其他模块的交互作用）对整体性能变异的贡献。这些结果可以按模块类别（例如拓扑结构、惯性、扰动机制）汇总，并在不同问题类别之间进行比较，从而系统地识别算法设计选择与问题景观特征之间的依赖关系。

B. 基于模块重要性的问题聚类

我们使用方差效应向量来识别模块重要性模式相似的问题类别组。我们将 f-ANOVA 结果中的每个问题类别视为该类别的一个 92 维嵌入（即元特征向量）。随后，我们对这 25 个嵌入进行聚类，每个嵌入对应一个 CEC'05 函数，以发现以相似方式"激活"PSO-*X* 模块的问题簇。由于我们希望保留方差贡献中的所有信息，因此直接在 92 维空间中进行聚类，不进行降维。我们尝试了不同的簇数和连接准则，并使用轮廓系数来指导合适簇数的选择。确定聚类结果后，我们根据已知的问题特征考察各簇的组成。具体而言，我们将数据驱动的分组与 CEC'05 基准测试的既定类别进行比较。这种比较揭示了常见问题特征（如模态性或可分性）与相似模块重要性概况之间的吻合程度。

表II CEC'05 基准测试问题：性质、定义域和全局最优点

Func	Name	Type	性质	定义域	最优值
f_1	平移球函数	单峰	可分离，移位	$[-100,100]^D$	0
f_2	平移的 Schwefel 1.2 函数	单峰	不可分，移位	$[-100,100]^D$	0
f_3	移位旋转高条件数椭圆函数	单峰	不可分、旋转、移位	$[-100,100]^D$	0
f_4	移位 Schwefel 1.2 + 噪声	单峰	不可分、含噪	$[-100,100]^D$	0
f_5	Schwefel 2.6（边界）	单峰	不可分，边界上最优	$[-100,100]^D$	0
f_6	移位 Rosenbrock	多峰	不可分，狭窄山谷	$[-100,100]^D$	0
f_7	平移旋转 Griewank 函数（无边界）	多峰	不可分、旋转	$[-600,600]^D$	0
f_8	平移旋转 Ackley 函数（边界）	多峰	不可分、旋转、边界	$[-32,32]^D$	0
f_9	平移 Rastrigin 函数	多峰	可分离，多个局部最优解	$[-5,5]^D$	0
f_{10}	平移旋转 Rastrigin 函数	多峰	不可分、旋转、多个最优解	$[-5,5]^D$	0
f_{11}	平移旋转 Weierstrass 函数	多峰	不可分，分形景观	$[-0.5,0.5]^D$	0
f_{12}	Schwefel 2.13	多峰	不可分，移位	$[-\pi,\pi]^D$	0
f_{13}	扩展 Griewank+Rosenbrock (f_8,f_2)	多峰	不可分，扩展混合	$[-3,1]^D$	0
f_{14}	移位旋转的 Scaffer's f_6	多峰	不可分，旋转	$[-100,100]^D$	0
f_{15}	混合组合 1	混合函数	混合，部分可分	$[-5,5]^D$	0
f_{16}	旋转混合函数 1	混合函数	不可分，旋转	$[-5,5]^D$	0
f_{17}	旋转混合函数 1 + 噪声	混合函数	不可分、含噪	$[-5,5]^D$	0
f_{18}	旋转混合函数 2	混合函数	不可分、陷阱、平坦区域	$[-5,5]^D$	0
f_{19}	旋转混合函数 2（窄盆地）	混合函数	不可分，窄盆地	$[-5,5]^D$	100
f_{20}	旋转混合函数 2（边界上最优）	混合函数	不可分，边界上最优	$[-5,5]^D$	0
f_{21}	旋转混合函数3	混合函数	不可分，旋转	$[-5,5]^D$	200
f_{22}	旋转混合函数3（病态）	混合函数	不可分，病态	$[-5,5]^D$	300
f_{23}	非连续旋转混合函数	混合函数	不可分，非连续	$[-5,5]^D$	300
f_{24}	旋转混合函数4	混合函数	不可分、旋转、复杂	$[-5,5]^D$	200
f_{25}	旋转混合函数4（无边界）	混合函数	不可分，最优点位于初始化范围之外	$[-5,5]^D$	200

IV. 实验设计

为了收集 PSO-X 性能数据，我们使用了 CEC'05 基准测试套件 [20]。由于问题的复杂性和多样性，CEC'05 测试套件仍然是测试元启发式算法的实用基准。下文将进一步介绍 CEC'05 测试套件、PSO-X 配置空间，以及 f-ANOVA 使用的数据集。

A. 基准函数集

CEC'05 基准测试套件 [20] 包含 25 个问题，根据其高层特征可分为三大类：单峰函数（f_1-f_5）、基本多峰函数（f_6-f_{12}）、扩展多峰函数（f_{13}-f_{14}）和混合组合函数（f_{15}-f_{25}）。在 CEC'05 基准测试中，所有问题类别都具有若干特征，使该套件呈现出高度异质性。例如，其中包括：非可分函数，即无法逐维进行优化的函数；数学变换，即搜索空间发生移位、旋转和缩放的函数；以及难以访问的全局最优点，这些全局最优点位于狭窄盆地、平坦区域、搜索空间的边界或初始化范围之外。表II展示了这 25 个函数、它们的性质、定义域和全局最优点。我们考虑问题的 10 维和 30 维版本，即 $D \in \{10, 30\}$。

B. 配置空间

为系统地生成本研究考察的 PSO 变体，我们对八个模块与共 26 种实现选项（见表 III）进行笛卡尔积运算，最终得到 1,424 种变体。每种算法变体都在 CEC'05 基准测试套件中的 25 个问题类别上分别独立运行 10 次。评估预算设为 $5000D$ 次函数评估，其中 D为优化问题的维度。性能根据已找到的最优解与全局最优解之间的距离（距离）进行评估。为确保数值稳定性并进行有意义的比较，小于 10^{-9} 的距离均截断为该阈值。对于每个问题实例，取 10 次运行所得距离的中位数作为最终性能指标。随后，对这些数值进行以 10 为底的对数变换。因此，距离指标的下限为 10^{-9}，数值越低表示优化性能越好。

C. f-ANOVA 数据集与层次聚类

由于本研究旨在量化 25 个问题类别中每个类别的各个模块的重要性，因此我们将数据整理为 25 个独立的数据集，每个问题类别对应一个数据集。每个数据集包含 1,424 种算法变体（数据实例），每个实例由八项模块设置（特征）描述，目标变量则对应这些变体在相应问题类别上的表现。分别对每个数据集运行 f-ANOVA，得到 25 个向量表示，每个向量都通过 $8 + \binom{8}{2} + \binom{8}{3}$ 项来体现模块效应。为了根据这些效应识别问题类别之间的相似性，我们采用层次聚类（HC） [42]。鉴于数据集规模相对较小（25 个实例），且我们关注可解释性，因此选择 HC，原因是它能够通过树状图提供清晰、透明的聚类过程，并

表 III PSO-X 框架中的 8 个模块和 26 种实现选项。实现选项按照其在 PSO-X 框架中的编号列出。特定实现选项对应的参数默认值以括号标示。

模块	实现选项和参数值
DNPP	0=DNPP-矩形， 1=DNPP-球形， 2=DNPP-加性随机（参数_r = 0.5）
accelCoeffCS	0=AC-常数 ($\phi_1 = 1.4$, $\phi_2 = 1.4$)，1=AC-随机 ($\phi_{1,t_0} = 2.4$, $\phi_{1,t_{max}} = 0.5$, $\phi_{2,t_0} = 0.5$, $\phi_{2,t_{max}} = 2.4$)
拓扑结构	0=环形拓扑，1=全连接拓扑
影响模型	0=邻域最优影响模型，1=MoI-fully informed
randomMatrix	0=Mtx-单位矩阵，1=Mtx 随机对角，4=Mtx 欧几里得旋转 (α-自适应, par_alpha = 30, par_beta = 0.01)，6=递增群组型矩阵和无
omega1CS	0=IW-恒定 ($\omega_1 = 0.0$)，0=IW-恒定 ($\omega_1 = 0.75$)，12=IW-基于速度自适应 (λ0.5, $\omega_{t_0} = 0.15$, $\omega_{t_{max}} = 0.95$)，14=IW-基于排名 ($\omega_{t_0} = 0.15$, $\omega_{t_{max}} = 0.95$)，15=IW-基于成功 ($\omega_{t_0} = 0.15$, $\omega_{t_{max}} = 0.95$)
perturbation1CS	0=无， 1=扰动$_{info}$-高斯（PM 成功率，扰动幅度 = 0.5，成功 = 40，失败 = 20）
perturbation2CS	0=无， 1=扰动$_{rand}$-矩形（PM 成功率，扰动幅度 = 0.5，成功 = 40，失败 = 20）

直观展示每个阶段簇的形成与合并过程。为了确定最合适的簇数，我们对聚类算法的超参数进行网格搜索，并使用轮廓系数 [43] 评估结果。该指标的取值范围为 -1 至 1，反映聚类质量——数值越高，表示簇越紧密且分离度越好。

五、结果与讨论

我们将结果与讨论分为四个部分。首先，我们考察粒子群优化-X 算法变体在各问题类别上的总体性能分布。其次，我们分析排序模块效应的累积方差贡献，即 8 个单独模块效应，加上 $\binom{8}{2} = 28$ 个成对模块效应，再加上 $\binom{8}{3} = 56$ 个三元模块效应。第三，我们根据模块效应向量对问题类别进行聚类，并考察这些簇与 CEC'05 问题特征之间的对应关系。为了了解某些模块选项及其交互作用如何影响粒子群优化-X 在不同簇中的性能，我们采用边际性能剖面（即与最优值的距离较小），并讨论具有代表性的案例。受篇幅限制，此处仅展示代表性图表。不过，本文的完整图表集、原始数据和处理后数据，以及用于处理和可视化的脚本，均作为补充材料提供。

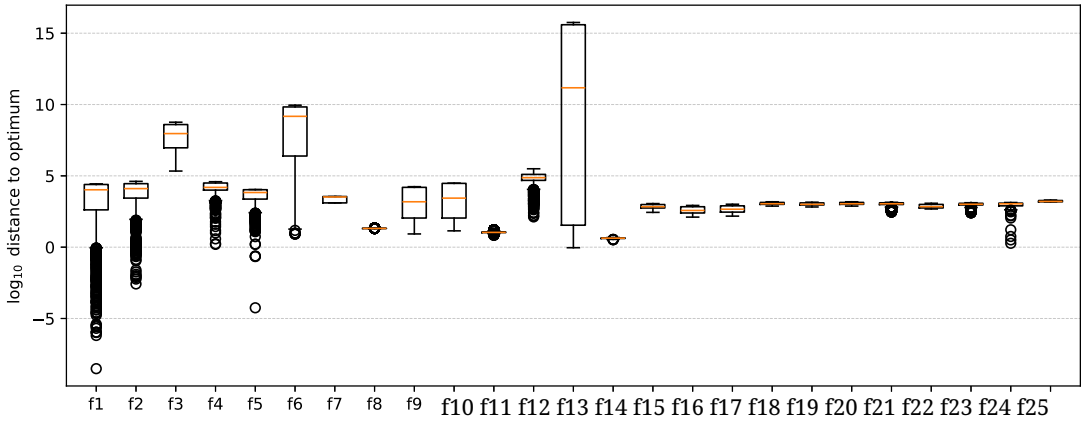
A. 算法性能分布

我们首先评估 1，424 个 PSO-X 变体在 25 个基准函数上的性能分布。性能采用对数变换后的距离指标进行衡量，即某个变体在 $5000D$ 次评估预算内找到的最佳目标值与全局最优解之间的差值。较低的 距离 值表示结果更好，可能达到的最小值为 -9 （即达到 10^{-9} 精度）。所有 PSO-X变体的性能分布以箱线图形式展示，分别对应 $10D$ （图 1a）和 $30D$ （图 1b）问题——函数 f_1 至 f_{25}。每个箱线图的离散程度反映了 PSO-X 算法使用不同模块时所达到性能的变异性。箱线图分布较为宽广的类别（即 f_1、f_2、f_3、f_5、f_6、f_9 和 f_{13}）显示出最佳与最差 PSO-X 变体之间存在较大差距，表明其对模块选择高度敏感，并说明通过重新设计算法和调整参数配置仍有显著的性能提升潜力。相比之下，若干问题类别（尤其是 f_7、f_8、f_{11} 以及大多数来自 f_{15}–f_{25} 的函数）呈现出非常紧密的性能分布，几乎所有变体的表现都相近。在这些情况下，即使对算法设计进行大幅调整，结果差异也很小，这意味着问题要么足够简单，以至于大多数配置都能成功；要么本身难度较高，使得所有配置都或多或少面临同样的困难。

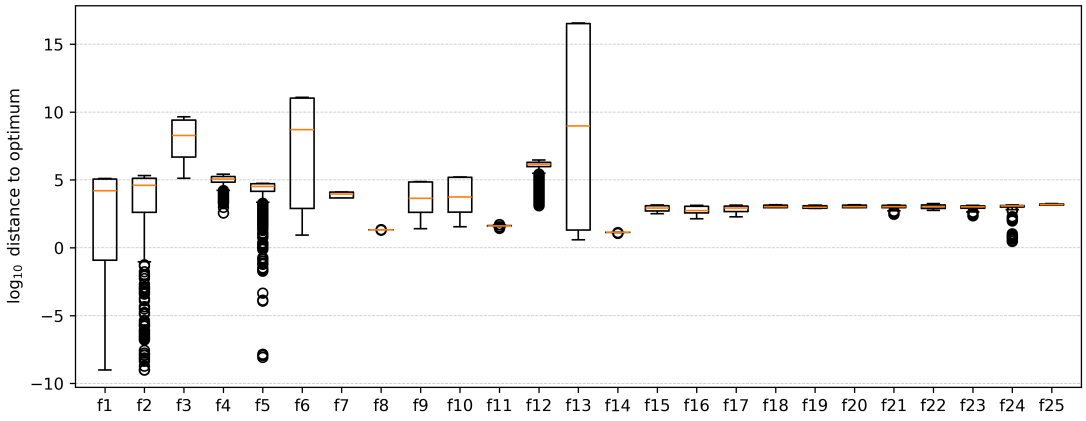
在 $D = 30$比较结果时，问题维度对性能变异性的影响似乎并不特别显著。尽管结果的变异性有所增加，但大多数PSO-X配置都能够扩展到更高维度的问题版本。对于函数f_4-f_{10}、f_{12} 和 f_{13}，解质量的下降更加明显，最佳变体与最差变体之间的差距也随之扩大。然而，对于函数 f_1、f_2 和 f_5，有更多PSO-X 配置能够达到更优的近似最优解；而对于函数 f_{15}-f_{25}，性能基本保持不变。这一趋势与其他模块化算法研究（例如 [40, 41]）中的观察结果一致，即随着问题复杂度的增加，性能离散程度也会增大。尽管CEC'05测试套件中的大多数混合函数对配置相对不敏感（大多数PSO-X变体的表现相近），但单峰和多峰问题仍有很大的改进空间，可以通过适当的模块选择来提升性能——尤其是在维度增加时。

B. 累积模块重要性

图 2a 和 2b 分别展示了 PSO-X 框架在 $10D$和 $30D$ 问题中排序模块效应的累积方差贡献。x轴表示纳入的模块效应数量（从影响最大者到第 92 个），而



(a) 10D 问题。



(b) 30D 问题。

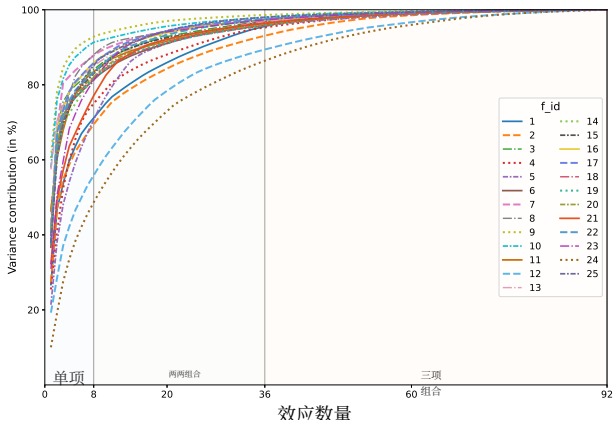
图1. 由包含CEC'05测试套件的25个问题类别组成的测试集中，1,424种粒子群优化-X 变体的性能分布。性能值（即在5,000D 次评估内已找到的最优解与全局最优解之间的绝对差值）以对数尺度呈现。

y轴表示这些效应所解释的性能方差累积百分比。每条曲线对应 25 个 CEC'05 基准函数之一，曲线的不同区段分别反映一阶（单个模块）效应、二阶（两两）交互效应和三阶（三项组合）交互效应的贡献。从图中可以看出，通常少数排名靠前的效应就能解释大部分性能方差，但这一累积速度在不同函数之间差异显著。

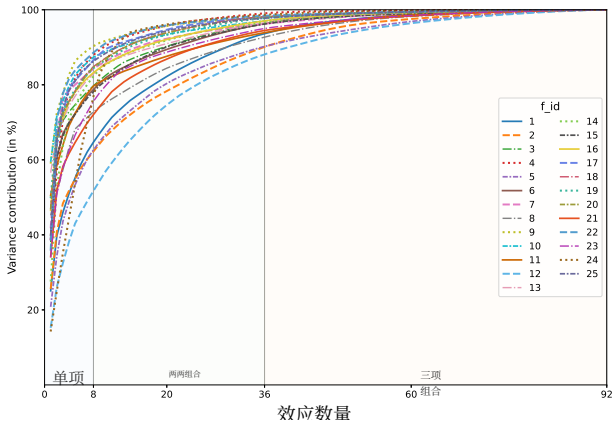
对于大多数问题，累积重要性曲线的初始斜率都很陡，表明单个模块的影响占方差的很大一部分。例如，在 10D 和 30D 图中，若干函数的曲线在前5个效应内迅速上升，方差解释率达到50–60%。这表明，对于这些问题，性能差异主要由少数几个重要模块及其交互作用所主导。在 10D 结果（图2a）中，f_7 、 f_9 和 f_{10}等函数体现了这一特征。它们的曲线在排名靠前的约5–10个效应内飙升至较高比例（占总方差的60–70%以上），之后增加效应带来的收益便逐渐递减。在 30D 结果（图2b）中，我们也观察到类似趋势，函数 f_7 、 f_9 和 f_{10}也呈现这一趋势，但曲线斜率略低于 10D。无论维度如何，只需相对较少的效应，累积方差就会达到较高水平。这表明，大多数函数不会引发极端的模块交互响应。在V-C节中，我们将针对每个问题类别，确定PSO-X 所使用的关键模块，并解释这些模块如何以及为何有助于提升其性能。

尽管总体趋势是少数模块效应影响显著，但我们也注意到不同函数之间存在明显差异。整个基准测试中，少数函数的曲线呈现出更为平缓的上升趋势。对于这些函数，没有单个模块或模块对能主导性能；相反，随着纳入的效应增多，方差缓慢累积。例如，在 10D 结果中，基准测试中段和后段的函数（如多峰函数 f_{12} 和复合函数 f_{24}）呈现出近乎线性的增长曲线——即使是前 10 个效应，也只能解释一半的方差。在 30D 结果中，多峰函数 f_{12} 的曲线表现出类似的趋势，但单峰函数 f_1-f_2 和混合组合函数 f_{25} 在前几十个效应中也仍然相对较低。这表明，这些函数的性能取决于多种模块及其交互作用的广泛组合。

通过比较 10D 和 30D 的结果，我们观察到



(a) 10D 问题。



(b) 30D 问题。

图2. 属于CEC'05测试套件的25个问题类别中，每个类别的92个模块效应的效应数量与累积方差贡献。

提高问题维度往往会使整体重要性曲线略微变平。事实上，尽管效应的排序相似，但图 2b 中的结果显示，其方差累积过程比图 2a 更为渐进。换言之，更高维度的实例需要考虑更多模块效应，才能达到相同的累积方差阈值。例如，在 f_1 中， $10D$ 曲线在 5 个效应以内便超过 60%；而在 $30D$ 中，其上升速度稍慢，因为前 5 个单一效应仅覆盖了约 40–45% 的方差。对于若干多峰函数，也存在类似趋势：在 30D中，一阶效应的贡献有所降低，这意味着在更高维度下，二阶和三阶交互作用对性能变异的贡献相对更大。最后，值得注意的是，通过考虑最多三个模块的组合效应，便可以解释整个 CEC'05测试套件中的性能方差。这表明，尽管可能存在高阶交互作用，但它们对 PSO-X 算法性能的影响可以忽略不计。

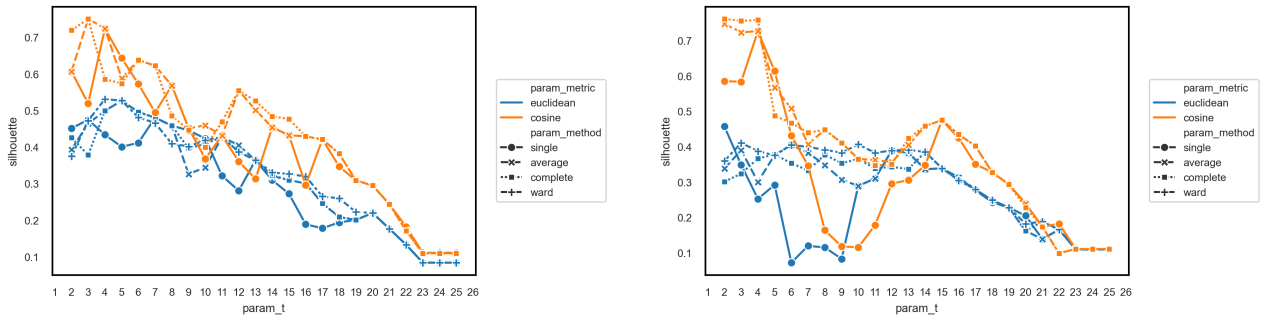
C. 按模块效应对问题类别进行聚类

我们对通过 f-ANOVA 获得的 25 维模块效应向量应用层次聚类（HC），以将模块重要性模式相似的问题类别归为一组。聚类前，我们通过评估各种设置下的聚类质量，对 HC 算法的超参数（簇数（k）、连接方法和距离度量）进行调优。图 3 展示了以下条件下的轮廓系数：(i) k 的取值从 2 到 25 变化；(ii) 两种距离度量（欧几里得和余弦）；以及 (iii) 四种连接方法（单、完全、平均、*Ward*）。图 3a（$10D$ 个问题）和图 3b（$30D$ 个问题）中的每条线代表一种距离度量与连接方法的组合。

我们发现，适中的簇数——5 到 7 个——能够获得最高的轮廓系数，其中余弦距离和完全链接法的整体表现最佳。基于此，在最终聚类中，我们为 $10D$ 问题选择了 $k = 6$ 个簇，为 $30D$ 问题选择了 $k = 5$ 个簇，并为 $10D$ 和 $30D$问题均采用余弦 距离和完全 链接法。在这些设置下，轮廓系数约为 0.6，表明簇的定义较为明确；然而，相对较低的最大簇间距离（在归一化方差尺度上， $10D$ 问题约为 0.065， $30D$ 问题约为 0.1）表明，所有问题类别的模块重要性概况总体上相似。这意味着，尽管可以识别出不同的簇，但簇间差异较为细微，而且在整个基准测试中，模块贡献的总体模式并无显著变化。

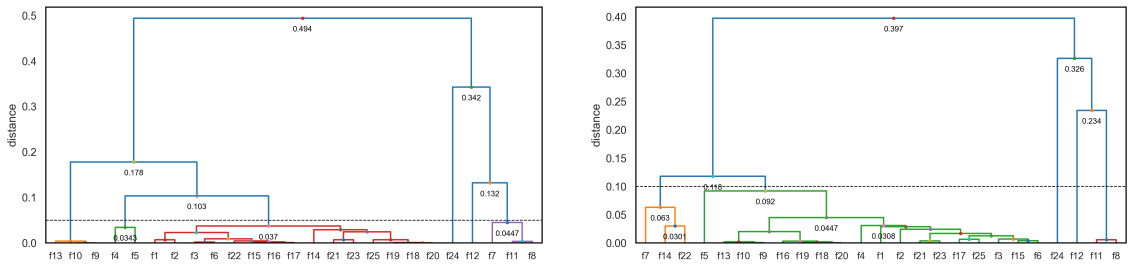
图 4 展示了每个 $10D$ 和 $30D$ 问题的树状图，这是一种表示层次聚类过程的树形图。纵轴表示每次合并时的簇间距离。虚线表示选定的切分位置，由此将 $10D$ 结果划分为 6 个簇（图 4a），将 $30D$ 结果划分为 5 个簇（图 4b）。分支上的红点标记了簇的合并距离。最高合并距离（图 4a 中为 0.494，图 4b 中为 0.397）以及整体距离相对较低，这证实了不同类别间的模块重要性模式总体相似。下文将基于 HC 过程的结果，分析依据模块效应对问题类别进行的分组。

1) 关于 *10D* 问题的结果：图5展示了 $10D$ 问题的向量表示聚类热图。行对应问题类别（f_{id}），列对应模块效应（效应$_{\text{id}}$）。颜色深浅表示各模块效应的方差贡献，范围从 0（不重要）到 1（非常重要）。第一列显示聚类分配，每种颜色代表一个不同的簇。结果表明，大多数问题类别集中在三个主要簇中（深绿色、红色和青色），少数则单独分布在较小且不同的簇中（紫色、粉色和浅绿色）。从聚类热图中可以看出，在所有问题类别中，影响最大的模块（即产生较强个体效应的模块）是 randomMatrix 和 omega1CS 模块，其次是 DNPP 模块。modelOfInfluence 和 perturbation1CS 模块也会在某些函数中激活，但其单独及组合贡献都较小。有趣的是，聚类热图还显示，拓扑结构、accelCoeffCS 和 perturbation2CS 的个体效应，以及它们绝大多数成对和三重交互作用



(a) 10D 问题。 (b) 30D 问题。

图3。 层次聚类算法调优的性能结果。使用轮廓系数作为聚类质量度量。



(a) 10D 问题。 (b) 30D 问题。

图 4. HC 聚类过程的树状图。它展示了问题类别如何依据相似性逐步分组，分支高度表示簇间差异。颜色编码突出显示了最终形成的簇。

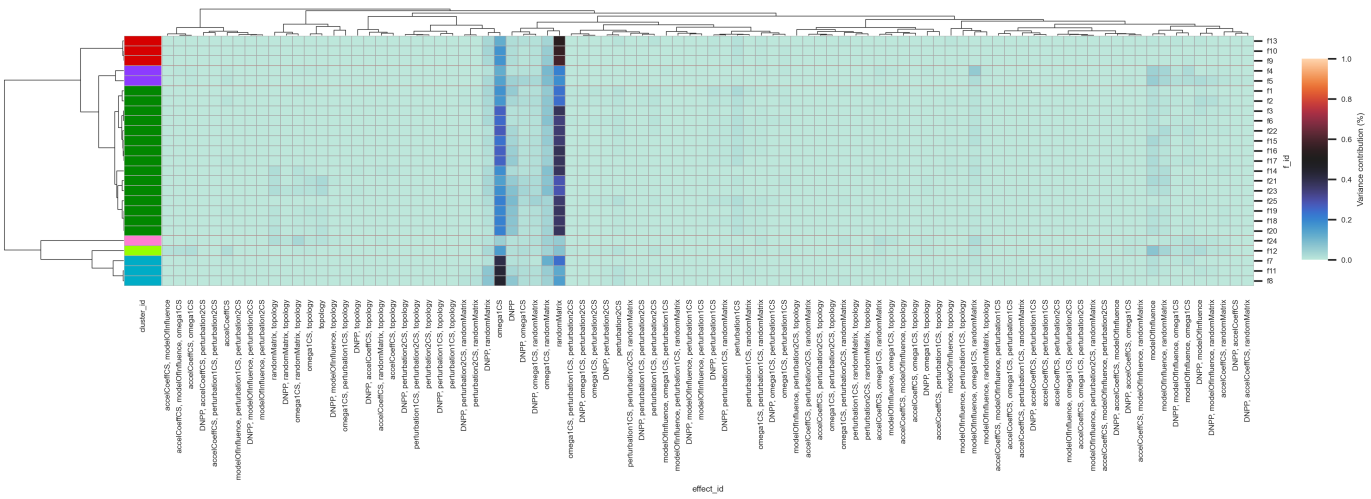


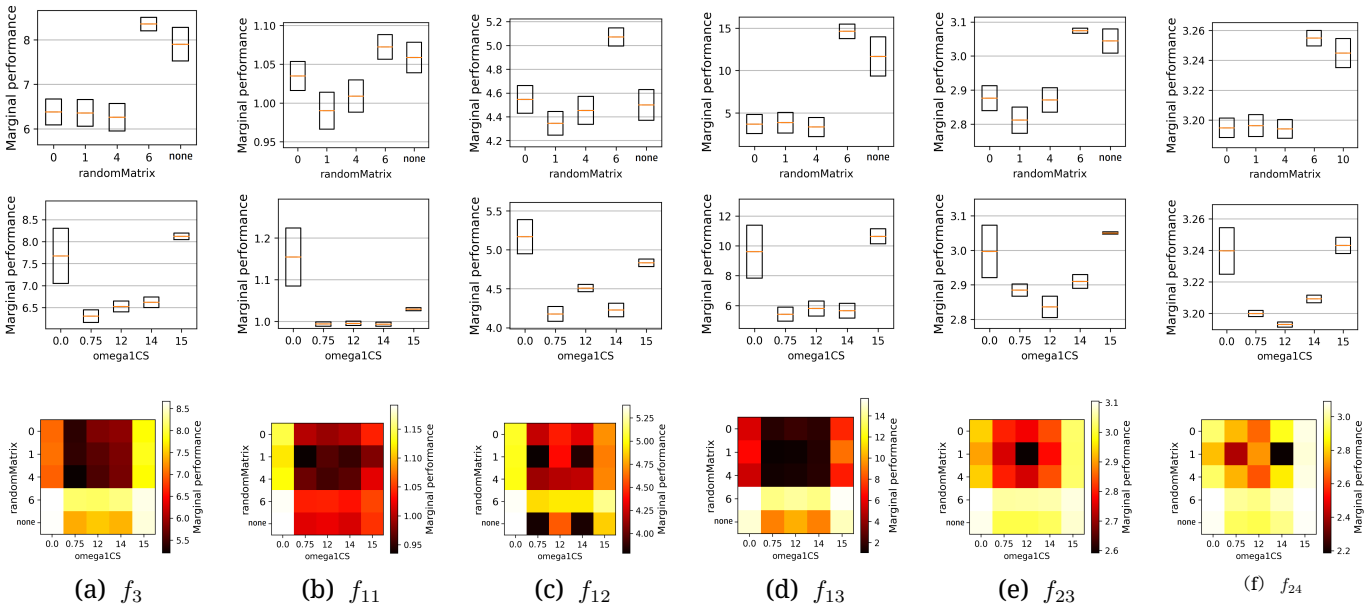
图 5. 10D 问题的聚类热图。

对 PSO-X 的性能没有产生任何显著影响。在交互效应中，涉及 omega1CS + randomMatrix、DNPP + omega1CS、DNPP+ randomMatrix 的组合，以及 DNPP + omega1CS +randomMatrix 这一组三元组合，影响尤为显著。这表明，PSO-X 变体的行为主要由这三个模块之间的相互作用驱动。

图6展示了 randomMatrix 和 omega1CS 模块的实现选项在不同簇函数的代表性示例中的边际性能，簇分别为紫色（f_3）、青色（f_{11}）、浅绿色（f_{12}）、红色（f_{13}）、深绿色（f_{23}）和粉色（f_{24}）。

图6上方的前两行以箱线图展示单个因素的边际效应， x 轴表示模块的不同选项， y 轴表示边际性能；数值越低，边际性能越好（即平均而言，与最优值的距离越小）。图6底部一行的热图展示模块选项组合的边际性能， x 轴和 y 轴表示模块的不同实现选项，颜色则表示边际性能；数值越低，性能越好。

randomMatrix 和 omega1CS 模块定义了粒子在搜索空间中的移动方式。因此，它们对



(a) f_3 (b) f_{11} (c) f_{12} (d) f_{13} (e) f_{23} (f) f_{24}

图6. randomMatrix和omega1CS模块选项的边际性能 在问题类别中的表现： f_3 （紫色簇）、 f_{11} （青色簇）、f_{12} （浅绿色簇）、 f_{13} （红色簇 f_{23} （深绿色簇）和 f_{24} （粉色簇），其中 $D = 10$。

粒子群优化-X 性能的影响在涉及搜索空间旋转和多个局部最优点的问题中尤为明显。实际应用中，randomMatrix 模块提供了多种选项，可生成随机变换矩阵，以旋转和/或缩放粒子的速度向量（$\vec{v}_{t+1}^{i}$）；omega1CS 则提供多种选项，可控制广义速度更新规则中的粒子惯性（ω_1）——参见表 I。结合这两个模块，粒子便能探索搜索空间中的不同超平面，并将步长从极长调整到极短的位移。在所有问题类别中始终带来最佳边际性能的实现选项包括：randomMatrix 模块的 0 = Mtx-单位矩阵、1 = Mtx 随机对角和 4 =Mtx 欧几里得旋转；以及 omega1CS 模块的 0 = IW-恒定、 12 =IW-基于速度自适应和 14 = IW-基于排名。相反，omega1CS 模块表现最差的选项似乎是两个极端： 0=IW- 恒定（$\omega_1 = 0.0$）它会消除惯性项；以及 15=IW-基于成功，它会在观察到改进后迅速调整惯性项，从而导致过早收敛。

如图6所示，红色簇中的函数（f_9、 f_{10}和 f_{13}）构成一个紧凑的群组，其中 randomMatrix 的边际重要性始终较高，其次是 omega1CS，而它们之间的交互作用较弱。在 CEC'05 基准测试的25个函数中，函数 f_9、f_{10}和 f_{13} 的边际性能受 randomMatrix模块影响最大。该簇中的三个函数具有相似的碗状形状，且与其他多峰函数（例如罗森布洛克函数 f_6）或较"容易"的单峰集合（f_1-f_5）相比，都表现出明显的多峰性。此外，函数 f_{10} 是函数 f_9的旋转版本，二者都有规律重复的局部最优点。这些模式表明，PSO-X 在这类问题上的性能很大程度上取决于搜索基底。不同旋转基使粒子能够在不同的超平面中观察搜索空间，从而更容易区分质量不同的谷地。此外，PSO-X 的性能似乎还取决于惯性降低的速度：优化过程的早期阶段侧重于探索，之后则侧重于开发。

青色簇由函数 f_7、 f_8和 f_{11}组成，在randomMatrix和omega1CS的影响平衡方面处于光谱的另一端。青色簇中的三个函数是omega1CS影响最强、而randomMatrix模块影响不那么重要的函数。该簇涵盖了具有复杂全局结构的函数，例如最优点位于初始化范围之外、具有边界最优点或崎岖度较高的函数，这表明在这些情形下算法行为一致。omega1CS模块的选项可通过限制速度向量 $\vec{v}_{t+1}^{i}$的大小，提供惯性步长控制和边界处理选项。较大的 $\vec{v}_{t+1}^{i}$使粒子能够迅速从搜索空间的一端移动到另一端，而较小的 $\vec{v}_{t+1}^{i}$ 则让粒子对其位移进行小幅调整，以找到难以到达的全局最优点。对于函数 f_7、 f_8和 f_{11}中普遍存在的细尺度崎岖度/噪声以及无界搜索空间，omega1CS的选项至关重要；这些选项使粒子能够根据景观特征自适应地调节速度。

最大的簇是深绿色簇，其中包含函数 f_1-f_3, f_6, f_{14}-f_{23} 和 f_{25}。如表II所示，这些函数具有多种特征，但我们可以区分出两种显著特征：碗状和椭圆景观（f_1-f_3, f_6, 以及 f_{14}）以及含噪景观（f_{15}-f_{23} 和 f_{25}）。深绿色簇是唯一同时包含单峰、多峰和混合函数这三类函数的簇，但其中多峰函数所占比例偏低（9 个多峰函数中仅有 2 个

个来自 CEC'05测试套件），相比之下，混合组合函数所占比例较高（CEC'05测试套件中的 11 个混合函数里有 10 个）。聚类热图（图5）显示 omega1CS 和 randomMatrix 模块的贡献较为均衡，但后者始终更为重要。这并不完全出乎意料，因为该簇中的大多数函数都具有旋转搜索空间。然而，该簇中的大多数函数不可分离，这一事实表明 randomMatrix 对处理这一特征也很有用。另一个值得注意的现象是，在深绿色簇中（尤其是函数 f_{15}-f_{23} 和f_{25}），与红色簇和青色簇相比，DNPP 模块发挥的作用更大；在后两个簇中，DNPP 的贡献微乎其微。omega1CS 与 randomMatrix 之间的成对交互作用也比在红色簇和青色簇中更常见，但并不突出。

我们观察到模块交互作用较广泛的函数，位于紫色簇（函数 f_4 和 f_5）、浅绿色簇（函数 f_{12}）和粉色簇（函数 f_{24}）中。紫色簇中的两个函数 f_4 和 f_5 都是单峰函数，其主要区别特征是：函数 f_4 由于加入了适应度噪声而具有崎岖景观，而函数 f_5 的景观则较为平滑。除了 omega1CS 和 randomMatrix，影响模型模块及其与 randomMatrix 的两两交互作用 modelOfInfluence +randomMatrix 对粒子群优化-X 的性能也有显著影响；不过，在函数 f_4中，还出现了三元组合 modelOfInfluence + omega1CS + randomMatrix，这表明它或许也会对崎岖的单峰函数产生一定影响，尽管影响较小。虽然我们的结果表明，在 CEC'05 基准测试的大多数函数中，modelOfInfluence 的影响都很小，但函数 f_4 和 f_5 中邻域最优影响模型的作用说明，在单峰函数中，粒子可以从跟随单一最优粒子中获益，而不是像 MoI-fully informed 那样同时跟随许多个体。

浅绿色簇中的函数 f_{12} 和粉色簇中的函数f_{24}，是 PSO-X 展现出最复杂模块交互的函数。具体而言，在函数f_{12}中，accelCoeffCS 和 modelOfInfluence 各自的贡献较小，但它们与 omega1CS 和 randomMatrix 存在交互作用。与其他多峰函数相比，函数 f_{12} 是唯一具有平滑景观且全局最优解所在的谷地界定清晰的函数。我们的结果表明，面对平滑景观的问题时，accelCoeffCS 和 modelOfInfluence 模块可能会影响 PSO-X 的性能。然而，对于函数 f_{24}，没有单一模块能够解释性能方差。如聚类热图——图5——和累积方差图——图2a——所示，多个模块都会影响 PSO-X 的性能，但没有任何一个模块占主导地位；相反，性能方差会随着时间推移，通过不同模块各自的作用及其交互作用不断累积。

2) 结果：关于 *30D* 问题： 本节重点介绍在这些问题的 $30D$ 变体上获得的结果，旨在确定更高维度在粒子群优化-X 模块激活及其边际性能效应中所起的作用。从图7所示的聚类热图中可以看到，存在三个主要簇（红色、紫色和深绿色），以及两个仅包含一个函数的小簇（浅绿色和青色）。与$10D$ 结果类似，方差分解证实，omega1CS、randomMatrix 和 DNPP 的主效应解释了大部分性能差异。在 V-B节中，我们观察到， $30D$问题的累积方差曲线在最初几个效应中陡升，随后逐渐趋于平缓，这表明维度增加使模块激活略有但明显增加。图8展示了各模块单独及组合后的边际性能。为此，我们选取了函数 f_1（紫色簇）、 f_8 （深绿色簇）和 f_{14} （红色簇）中的 randomMatrix 和 omega1CS 模块，函数 f_{12} （青色簇）中的 modelOfInfluence 和 omega1CS 模块，以及函数 f_{24} （浅绿色簇）中的 DNPP 和 omega1CS 模块。

在 $30D$ 问题上取得最佳性能的 omega1CS 和 random- Matrix 实现选项，与在 $10D$ 问题上的相同。此外，与 $10D$ 问题上的结果类似，在 $30D$ 问题中影响最大的交互效应涉及以下组合：omega1CS + randomMatrix、DNPP+ omega1CS、DNPP + randomMatrix，以及三元组合 DNPP + omega1CS + randomMatrix。然而，在30D 问题中，我们观察到 perturbation1CS模块的激活略强，而影响模型模块的激活略弱。这两个模块协同工作。perturbation1CS模块通过计算一个以邻近解为中心的随机向量，为粒子运动增添额外的多样性；而影响模型模块则决定粒子可以从中选择的邻近解集合。perturbation1CS激活增强表明，随着维度数量增加，粒子会受益于社会组件中更高水平的随机性，从而降低影响模型实现选项具体选择的重要性。

对于 DNPP 模块，最佳选项是 0 =DNPP-矩形和 1 = DNPP-球形：在单峰函数中，DNPP-矩形的单独效果更好 f_1-f_5 ，而在多峰函数和混合函数中，DNPP-球形表现更佳。这一趋势在$10D$上的结果中也有所体现，但在 $30D$ 问题中更为明显。值得注意的是，尽管 DNPP 决定了粒子用于采样新位置的映射类型，以及实现中是否使用 randomMatrix 和 perturbation1CS，但 DNPP 模块本身对粒子群优化-X的性能贡献甚微。这是因为 DNPP 模块充当其他模块交互所依托的模板；因此，它对粒子群优化-X 的性能产生间接影响，而 randomMatrix 等模块则承担主要工作负载。

在图7中，红色簇包含函数 f_7、 f_{14}和 f_{22}，它们对 omega1CS、randomMatrix 及其两两交互作用表现出相似的强敏感性。这三个函数都具有多峰、不可分且旋转的景观，而函数 f_7 还是无界的，函数f_{22} 则是病态的。这些特性使这三个函数都尤其难以求解。在更高维度下，吸引域会变得更狭长、重叠更多，因此

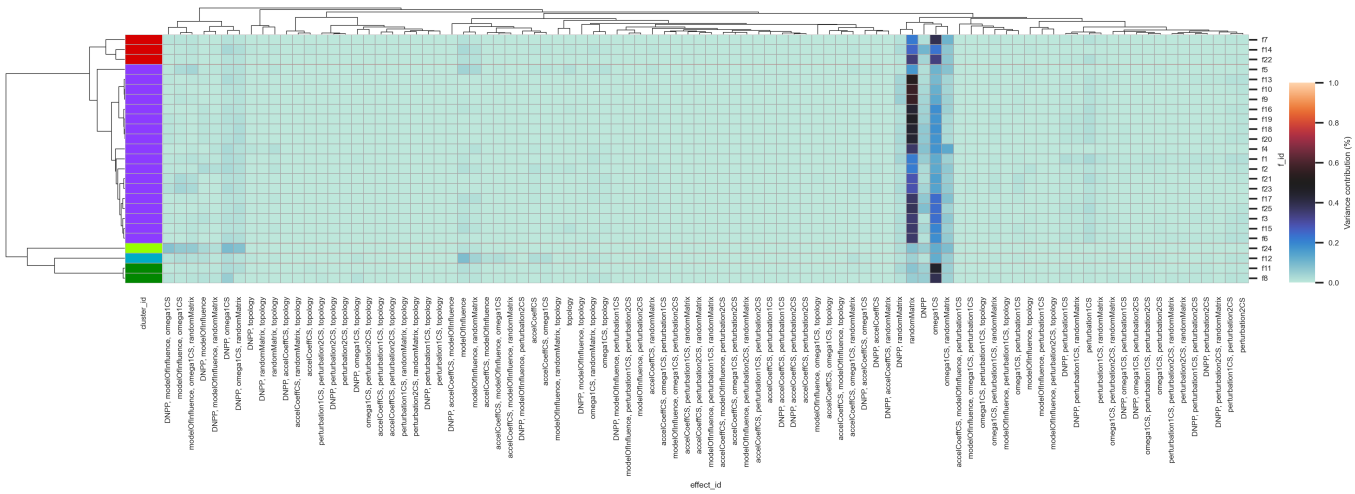


图7。 $30D$ 问题的聚类热图。

omega1CS 对粒子逃离这些吸引域并移动到更高质量区域至关重要。另一方面，要处理多峰且旋转的景观，需要为 randomMatrix 模块选择合适的选项，因为随着维度数量增加，估计其速度向量（$\vec{v}_{t+1}^{i}$）的旋转基会变得更加复杂。在此类问题中，omega1CS 与 randomMatrix 之间错综复杂的相互作用，可通过函数 f_{14}的成对交互热图直观地呈现出来，见图8d；该图显示，只有将选项 $1 =$ Mtx 随机对角与12 =IW-基于速度自适应结合使用，才能获得更高的边际贡献效应。

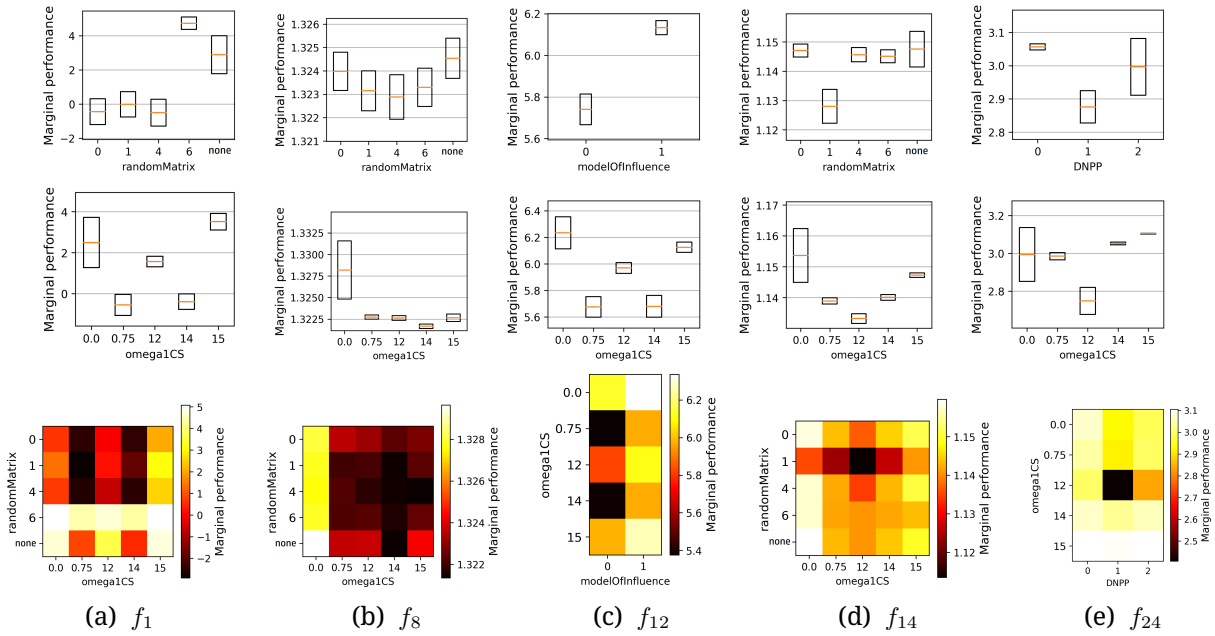
深绿色簇中的函数——f_8 和 f_{11}——是多峰函数，其景观含噪，且全局最优点位于难以到达的位置。这两个函数都表现出较强的 omega1CS 效应，而 randomMatrix 和 DNPP 模块及其组合交互作用的贡献虽较为适中，但同样重要。对于函数 f_8，DNPP 模块与 omega1CS 模块之间的交互作用也有显著影响。这两个函数在 $30D$ 问题上的结果与在 $10D$上的结果相似，说明维度并非决定函数 f_8 和 f_{11}复杂性的关键因素。在这类问题中，粒子通过控制惯性来应对含噪且高度崎岖的结构的能力，是影响性能的最大因素。从函数 f_8的箱线图和热图（图8b）中可以看出，使用选项 $12 =$ 基于速度自适应的 IW 自适应，是影响粒子群优化-X性能的主要因素。

图7中最大的簇是紫色簇，其中包含函数 f_1-f_6、f_9、f_{10}、f_{13}、f_{15}-f_{21}、f_{23} 和 f_{25}。在这个簇中，影响最大的模块是 randomMatrix，在该簇的18个函数中，有13个函数的性能方差有50-60%归因于它。尽管randomMatrix在绝大多数多峰函数和混合函数中的单独效应都很明显，但我们发现，在三个单峰函数（f_1、f_2、f_5）和两个混合组合（f_{21} 和 f_{23}）中，其效应有所减弱（对方差的贡献低于30%）。事实上，在由 f_1、f_2、f_5、f_{21} 和 f_{23}组成的函数子集中，omega1CS的单独效应、omega1CS与randomMatrix的两两交互效应、DNPP与omega1CS的两两交互效应，以及DNPP、omega1CS和randomMatrix的三重交互效应影响最大。与其他簇相比，perturbation1CS模块在紫色簇中最为活跃，但其贡献至多只是增量式的。

青色簇中的函数 f_{12} 和浅绿色簇中的函数 f_{24} 在 $30D$ 中仍然是离群值。它们的模块激活与 $10D$ 结果中的情况相似，但边际效应更强。函数 f_{12} 是唯一一个我们观察到其他模块对性能的贡献与 randomMatrix 或 omega1CS 相近的函数。在函数 f_{12} 中，modelOfInfluence 模块与 randomMatrix 和 accelCoeffCS 模块的组合效应也占主导地位，而 DNPP 模块则似乎完全没有影响。对于 f_{24}，randomMatrix、omega1CS 和 DNPP 的成对交互作用及三重交互作用所产生的方差贡献，与它们各自的独立效应相近。这表明函数 f_{24} 可能对实现设计尤为敏感，因为它涉及三个模块协同工作。

VI. 结论与展望

我们量化了 PSO-X框架中的算法模块对 CEC'05 问题类别性能的影响。我们在包含 1,424 种自动生成的 PSO-X 变体的大型设计空间上应用 f-ANOVA，将性能变异分解为关键模块的独立效应、两两交互效应和三重交互效应。在 $10D$ 和 $30D$两种场景下，累积方差分析均显示，少数模块效应（即 randomMatrix、omega1CS 和 DNPP 模块）解释了大部分可解释方差；单项效应在初期贡献最大，随后两两交互作用进一步提高了解释力。虽然存在三重交互作用，但其影响通常较弱，这表明性能方差主要由主效应和两两交互作用所解释。



(a) f_1 (b) f_8 (c) f_{12} (d) f_{14} (e) f_{24}

图 8. randomMatrix 和 omega1CS 模块在问题类别 f_1 （紫色簇）、 f_8 （深绿色簇）和 f_{14}（红色簇）中的边际性能；影响模型和 omega1CS 模块在问题类别 f_{12} （青色簇）中的边际性能；以及 DNPP 和 omega1CS 模块在问题类别 f_{24} （浅绿色簇）中的边际性能。所有结果均采用 $D = 30$。

对各函数重要性剖面进行聚类分析后发现，某些问题类别具有相似的模块效应模式，即使这些类别与 CEC'05 高级特征分类仅部分吻合。randomMatrix 模块在大多数函数上表现出较高的边际效应，其次是 omega1CS 模块，最后是 DNPP 模块。我们的结果为 PSO 提供了两方面的指导。从设计角度看，在构建 PSO-X 变体时，选择并配置 randomMatrix 和 omega1CS 模块至关重要；而 DNPP 模块则可根据预期复杂度固定——问题简单时选择 DNPP-矩形，问题困难时选择 DNPP-球形。从选择角度看，观察到的簇表明，在重要性模式相似的函数组内，模块选择具有可迁移性，即便这些组与传统的基于属性的划分并不完全一致。实践中，这意味着可以采用组合策略来实现 PSO：使用一组 PSO-X 配置，主要通过 randomMatrix 和 omega1CS 选项、其次通过 DNPP 和 modelOfInfluence 选项进行多样化，从而以最少的冗余覆盖大多数问题类别。

我们研究的主要局限在于，其范围局限于 CEC'05测试套件以及所考察的特定 PSO-X 模块。显然，不同的问题族、扩展的模块空间以及更大的参数配置范围可能会重塑重要性格局。此外，尽管我们测量了三重交互作用，但高阶效应仍有待探索，并且它们可能存在于截然不同的实验条件下。未来的研究将：(i) 扩大 PSO-X 模块集，以测试更多模块，重点关注旋转和种群控制方案；(ii) 在其他基准测试族（包括 BBOB 和 CEC 基准套件）以及其他模块化框架（例如 METAFO$\mathbb{R}$）上重复该分析；以及 (iii) 加强问题与算法之间的联系，创建能够实现有原则、数据驱动的模块选择的增强型元特征。

参考文献

[1] D. G. Luenberger, Y. Ye 等，《线性与非线性规划》。Cham：Springer，2016年。[2] C. Audet 和 W. Hare，《无导数与黑箱优化》。柏林、海德堡，德国：Springer，2017年。[3] I. Rechenberg，"进化策略：按照生物进化原理优化技术系统"，博士论文，过程工程系，柏林工业大学，1971年。[4] H.-P. Schwefel，《基于进化策略的计算机模型数值优化》。巴塞尔，瑞士：Birkhäuser，1977年。[5] R. Storn 和 K. Price，"差分进化——一种在连续空间中进行全局优化的简单高效启发式方法"，《全球优化期刊》，第11卷，第4期，第341–359页，1997年。[6] J. Kennedy 和 R. Eberhart，"粒子群优化"，载于*ICNN'95*——国际神经网络会议论文集，第4卷。新泽西州皮斯卡塔韦：IEEE，1995年，第1942–1948页。[7] R. Eberhart 和 J. Kennedy，"一种采用粒子群理论的新型优化器"，载于第六届微型机器与人类科学国际研讨会论文集。新泽西州皮斯卡塔韦：IEEE 出版社，1995年，第39–43页。[8] M. Birattari、T. Stützle、L. Paquete 和 K. Varrentrapp，"一种用于配置元启发式算法的竞速算法"，载于遗传与进化计算会议论文集，*GECCO 2002*，W. B. Langdon 等编。

Morgan Kaufmann 出版社，旧金山，加利福尼亚州，2002年，第11–18页。[9] M. Birattari，《元启发式调优：机器学习视角》，计算智能研究丛书。柏林/海德堡，德国：Springer，2009年，第197卷。[10] M. López-Ibáñez 和 T. Stützle，“多目标蚁群优化算法的设计选择对性能的影响：双目标旅行商问题的实验研究”，载于遗传与进化计算会议论文集，*GECCO 2010*，M. Pelikan 和 J. Branke 编。纽约：ACM Press，2010年，第71–78页。[11] J. N. Hooker，“启发式方法测试：我们全都做错了”，《启发式方法期刊》，第1卷，第1期，第33–42页，1996年。[12]C. García-Martínez、P. D. Gutiérrez、D. Molina、M. Lozano 和 F. Herrera，“自 CEC 2005 实参数优化竞赛以来：十年的研究与进展，以及比较分析的不足”，《软计算》，第21卷，第19期，第5573–5583页，2017年。[13]A. Nikolikj、M. A. Munoz 和 T. Eftimov，“连续黑箱优化算法的基准测试表现：深入解析算法成功与失败的可解释洞见”，《群体与进化计算》，第94卷，第101895页，2025年。[14] C. Doerr、H. Wang、F. Ye、S. Van Rijn 和 T. Bäck，“IOH-profiler：一种用于迭代优化启发式方法的基准测试与分析工具”，《arXiv 预印本 arXiv:1810.05281》，2018年。[15] T. Bartz-Beielstein、C. Doerr、D. v. d. Berg、J. Bossek、S. Chandrasekaran、T. Eftimov、A. Fischbach、P. Kerschke、W. La Cava、M. Lopez-Ibanez 等，“优化基准测试：最佳实践与待解决问题”，《arXiv 预印本 arXiv:2007.03488》，2020年。[16]C. L. Camacho-Villalón、T. Stützle 和 M. Dorigo，“设计新型元启发式算法：人工方法与自动方法”，《智能计算》，第2卷，第0048期，第1–15页，2023年。[17] C. L. Camacho-Villalón、M. Dorigo 和 T. Stützle，“PSO-*X*：用于自动设计粒子群优化算法的基于组件的框架”，《*IEEE* 进化计算汇刊》，第26卷，第3期，第402–416页，2022年。[18] F. Hutter、H. H. Hoos 和 K. Leyton-Brown，“一种高效评估超参数重要性的方法”，载于第*31*届国际机器学习会议论文集，第32卷，2014年，第754–762页。 [在线]。网址：http://jmlr.org/proceedings/papers/v32/hutter14.html[19] J. N. Van Rijn 和 F. Hutter，“跨数据集的超参数重要性”，载于第*24*届 *ACM SIGKDD* 知识发现与数据挖掘国际会议论文集，2018年，第2367–2376页。[20] P. N. Suganthan、N. Hansen、J. J. Liang、K. Deb、Y. P. Chen、A. Auger 和 S. Tiwari，“CEC 2005 实参数优化专题会议的问题定义与评估标准”，南洋理工大学，新加坡，技术报告，2005年。[21]A. Banks、J. Vincent 和 C. Anyakoha，“综述粒子群优化。第一部分：背景与发展，” *Natural Computing*，第6卷，第4期，第467–484页，2007年。[22] ——，“粒子群优化综述。第二部分：混合化、组合、多准则与约束优化，以及指示性应用，”*Natural Computing*，第7卷，第1期，第109–124页，2008年。[23] M. R. Bonyadi 和 Z. Michalewicz，“单目标连续空间问题的粒子群优化：综述”，2017年。[24] A. P. Engelbrecht，“粒子群优化：全局最优还是局部最优？”载于*2013*年*BRICS*计算智能大会暨第*11*届巴西计算智能大会，2013年，第124–135页。[25] T. Stützle 和 M. López-Ibáñez，“元启发式算法的自动化设计”，载于元启发式手册，丛书：运筹学与管理科学国际丛书，M. Gendreau 和 J.-Y. Potvin 编。纽约：Springer，2019年，第272卷，第541–579页。[26]J. de Nobel、D. Vermetten、H. Wang、C. Doerr 和 T. Bäck，“通过调优评估新思路与现有算法模块相互作用时的益处”，载于*GECCO'21 Companion*，F. Chicano 编。纽约：ACM Press，2021年，第1375–1384页。[27] D. Vermetten、F. Caraffini、A. V. Kononova 和 T. Bäck，“模块化差分进化”，载于遗传与进化计算会议论文集。纽约：ACM Press，2023年，第864–872页。[28] C. Camacho-Villalón、M. Dorigo 和 T. Stützle，“METAFOR：用于单目标连续优化问题的混合元启发式软件框架”，*arXiv*预印本 *arXiv:2502.11225*，2025年。[29] M. López-Ibáñez、J. Dubois-Lacoste、L. Pérez Cáceres、T. Stützle 和 M. Birattari，“irace软件包：用于自动算法配置的迭代竞速”，*OperationsResearch Perspectives*，第3卷，第43–58页，2016年。[30] F. Hutter、H. H. Hoos、K. Leyton-Brown 和 T. Stützle，“ParamILS：一种自动算法配置框架”，*Journal of ArtificialIntelligence Research*，第36卷，第267–306页，2009年10月。[31] F. Hutter、H. H. Hoos 和 K. Leyton-Brown，“面向通用算法配置的序贯模型优化”，载于*Learning and Intelligent Optimization*，第五届国际会议，*LION 5*，丛书：Lecture Notes in Computer Science，C. A. Coello Coello 编。德国海德堡：Springer，德国海德堡，2011年，第6683卷，第507–523页。[32]C. L. Camacho-Villalón、M. Dorigo 和 T. Stützle，“PSO-*X*：用于自动设计粒子群优化算法的基于组件的框架：补充材料”，http://iridia.ulb.ac.be/supp/IridiaSupp2021-001/，2021年。[33] F. Hutter、H. H. Hoos 和 K. Leyton-Brown，“利用前向选择识别关键算法参数和实例特征”，载于*Learning and Intelligent Optimization*，第七届国际会议，*LION 7*，丛书：Lecture Notes in Computer Science，P. M. Pardalos 和 G. Nicosia 编，第7997卷。Springer，德国海德堡，2013年，第

364–381。[34] N. Siegmund、A. Grebhahn、S. Apel 和 C. Kästner，“高度可配置系统的性能影响模型”，载于*2015*年第*10*届软件工程基础联合会议论文集，2015年，第284–294页。[35] A. Biedenkapp、M. Lindauer、K. Eggensperger、F. Hutter、C. Fawcett 和 H. H. Hoos，“通过消融和替代模型高效分析参数重要性”，载于*AAAI*人工智能大会，S. P. Singh 和 S. Markovitch 编。AAAI出版社，2017年2月。[在线]。参见：
h
t
t
p
s://
a
aai.org/ocs/index.php/AAAI/AAAI17/paper/view/14750
[36] R. P. Prager、H. Trautmann、H. Wang、T. H. Bäck 和 P. Kerschke，“通过分类器链和探索性景观分析对模块化CMA-ES进行逐实例配置”，载于*2020*年*IEEE*计算智能研讨会系列（*SSCI*）。IEEE，2020年，第996–1003页。[37]A. Kostovska、D. Vermetten、S. Džeroski、C. Doerr、P. Korosec 和 T. Eftimov，“景观特征对模块化CMA-ES变体性能预测的重要性”，载于遗传与进化计算会议论文集，2022年，第648–656页。[38] A. Kostovska、C. Doerr、S. Džeroski、P. Panov 和 T. Eftimov，“黑箱优化中的几何学习：用于算法性能预测的图神经网络框架”，载于遗传与进化计算大会附属会议论文集，2025年，即将出版。[39] N. Van Stein、D. Vermetten、A. V. Kononova 和 T. Bäck，“迭代优化启发式方法的可解释基准测试”，*ACM*进化学习汇刊，2024年。[40] A. Nikolikj、A. Kostovska、D. Vermetten、C. Doerr 和 T. Eftimov，“量化模块化优化框架中单个模块及联合模块的影响”，载于*2024*年*IEEE*进化计算大会（*CEC*）。IEEE，2024年，第1–8页。[41] A. Nikolikj 和 T. Eftimov，“探索模块化CMA-ES在不同问题类别中的模块交互”，《群体与进化计算》，第98卷，第102116页，2025年。[42] D. Müllner，“现代层次凝聚聚类算法”，*arXiv*预印本 *arXiv:1109.2378*，2011年。[43]P. J. Rousseeuw，“轮廓系数：用于解释和验证聚类分析的图形辅助工具”，计算与应用数学杂志，第20卷，第53–65页，1987年。