用于性能评估的 BBOB 问题仿射组合

Diederick Vermetten 莱顿高级计算机科学研究所 荷兰莱顿
d.
l.ver
metten@liacs.leidenuniv.nl

Furong Ye 莱顿高级计算机科学研究所 荷兰莱顿
f.ye
@liacs.leidenuniv.nl

Carola Doerr 索邦大学、法国国家科学研究中心、LIP6实验室 法国巴黎
Carola.Doerr@lip6.fr

arXiv:2303.04573v1 [cs.NE] 8 Mar 2023

摘要
基准测试在优化算法的开发与分析中发挥着重要作用。因此，基准问题的定义方式会显著影响我们从特定基准研究中获得的见解。扩展现有基准函数范围的一种简便方法，是对成对的函数进行仿射组合。从景观分析的角度来看，这些函数组合能够在两个基函数之间实现平滑过渡。

在本研究中，我们展示了如何利用这些仿射函数组合来分析优化算法的行为。具体而言，我们指出，通过改变组合问题之间的权重，可以深入了解新增全局结构对优化算法性能的影响。通过分析更多函数组合上的性能轨迹，我们还表明，目标函数的尺度和最优点的位置等因素会极大影响对这些结果的解读。

关键词
黑箱优化，基准测试，性能分析

1 引言

基准测试是优化算法开发中的关键环节。基准问题不仅用于在标准化的问题集上比较不同优化器的效果，对算法在这些问题上的行为进行分析，也常常有助于深入了解算法的特性。因此，基准问题的设计对整个优化领域都有重大影响 [1]。

在单目标、连续、无噪声优化领域，最常见的基准套件之一恰如其名，称为黑箱优化基准（BBOB） [7]。该套件是COCO框架 [6], 的一部分，在过去十年中得到了广泛采用。该套件包含 24 个问题，每个问题旨在代表一组全局景观属性。通过一系列变换，可以为每个问题生成许多不同的实例，使研究人员能够测试其算法的不同不变性。鉴于 BBOB 套件广受欢迎，针对其具体特性的研究不胜枚举 [13, 16, 17]。

一种广受欢迎的连续优化问题研究方法是探索性景观分析（ELA） [15]。该技术旨在通过大量特征刻画景观的低层属性。将其应用于 BBOB 套件后发现，24 个函数的实例通常会聚集在一起，而不同函数之间的区分也相对稳健 [20]。这一观察引出了一个问题：如何探索问题之间的空间。

在最近的一项研究中，研究者提出了成对 BBOB问题之间的仿射组合，并使用 ELA [4] 对其进行了分析。分析结果表明，改变这些组合的权重对景观特征的影响相对平滑。因此，这些新函数有望用于研究不同景观之间的过渡，从而更深入地分析景观与算法行为之间的关系。

为了探究仿射函数组合在多大程度上可用于研究算法行为，我们开展了一项基准测试研究，考察仿射组合对五种数值黑盒优化算法性能的影响。我们使用了包含非球面模型的函数组合，以展示新增全局结构对算法相对排名的影响。此外，我们还表明，将具有不同全局性质的函数组合起来，并不总能带来平滑的性能过渡。我们举例说明，由两个函数组合而成的函数可能比其所基于的基函数难得多，也可能略微容易一些。

2 相关工作

2.1 BBOB问题套件

在连续优化基准测试中，最受欢迎的基准测试套件之一是 BBOB 系列，它是 COCO 框架的一部分。无噪声单目标套件由 24 个问题组成，每个问题都可以通过一组不同的变换生成多个实例。这些函数实例旨在保留函数的全局特性，同时改变全局最优点的位置等因素，使优化器无法直接利用这些特性。然而，这些变换对低层景观属性的确切影响并不那么简单，因此同一函数的不同实例可能会导致算法行为出现显著差异 [13]。

2.2 仿射函数组合

虽然使用函数实例能让 BBOB 套件覆盖比原始函数本身更广泛的问题景观，但通过这种方式能够创建的景观类型仍有限。最近，有人提出通过对成对的 BBOB 函数进行仿射组合来生成新的基准函数 [4]。研究表明，以一组 ELA特征进行测量时，这些组合能够平滑地填充低层景观属性空间。这些结果表明，即使是相对

简单的函数创建流程，也有望让我们对函数景观的运作方式获得新的认识。

3 实验设置

在本研究中，我们采用了 [4] 中仿射函数组合的一个略作修改的版本。具体而言，我们将 BBOB 套件中的两个函数组合定义如下：

$$C(F_1, I_1, F_2, I_2, \alpha)(x) = \exp\Big(\alpha \log\left(F_1(x) - F_1(O_1)\right) + (1-\alpha)\log\left(F_2(x - O_1 + O_2) - F_2(O_2)\right)\Big)$$

其中，F_1、I_1、F_2、I_2 分别是两个基函数及其在 BBOB [7] 中定义的实例编号。O_1 和 O_2 分别表示函数 F_1 和 F_2 最优点的位置。在评估 F_2 时，我们将其变换为 x，以确保最优点的位置为 O_1。与原始定义不同，我们在聚合前减去最优值，这样就可以计算这些问题的对数平均。通过这种方式，我们可以在不同问题之间使用一致的 α 值，而无需像 [4] 中那样进行基于熵的选择。此外，这还有一个好处：最优解的目标值始终为 0，从而简化了对不同实例和不同问题的性能比较。在图 1 中，我们展示了 F21 与 F1 组合后的景观变化，以及不同 α 值下的情况。

为了实现这些函数组合，我们使用 IOHexperimenter [3]框架。我们访问 BBOB问题，按照所述方式将它们组合起来，并将其封装为一个新问题。这样，我们便可以使用 IOHexperimter 的任何内置日志记录和跟踪选项。特别是，它允许我们将性能数据存储为可直接导入 IOHanalyzer [25] 进行后处理的文件格式。

对于我们的算法组合，我们使用 Nevergrad 工具箱，它为各种优化算法提供了统一的接口 [19]。在本研究中，我们对以下算法进行基准测试：

- 粒子群优化 (PSO) [10]
- 基于线性近似的约束优化 (Cobyla) [18]
- 差分进化 (DE) [21]
- 多元正态分布估计算法 (EMNA) [12]
- 对角协方差矩阵自适应进化策略（dCMA-ES） [8]

对于这些算法中的每一个，我们均采用 Nevergrad 中选定的默认参数。算法的每次运行预算为 $2\,000D$，其中 D 是问题的维度。我们针对每个实例进行 5 次独立运行。在本文的其余部分，我们设定 $I_2 = 1$。因此，当讨论仿射函数组合的实例 $C(F_1, I_1, F_2, I_2, \alpha)$时，我们指的是 I_1。

可复现性 为确保可复现性，我们将撰写本文所使用的全部代码发布在 Zenodo 仓库中 [24]。该仓库包含数据生成代码、生成的原始数据以及用于创建下文所述结果的后处理脚本，这些内容遵循 [14] 中提出的建议。此外，我们还提供了一个 Figshare 仓库，其中包含本文无法收录的其他图表和动画 [24]。

4 仿射组合与 F1 的性能比较

在第一组实验中，我们采用将每个函数分别与 F1（球体模型）进行组合的仿射组合。如图 1 所示，将球体模型添加到另一个函数中，会产生额外的全局结构，从而引导优化过程趋向全局最优点。因此，这类组合可能使我们能够研究新增全局结构对优化算法性能的影响。在一定程度上，通过比较原始 BBOB 的函数组在不同全局结构水平下的结果，已经可以对这一点进行研究；但仿射函数组合能够实现更为细致的探究。由于这些组合函数的景观特征似乎会随着 α 的变化而平滑地变化，我们可以假设算法性能也会表现出类似的行为。

图2展示了对角协方差矩阵自适应进化策略的性能，以经验累积分布函数（ECDF）曲线下面积衡量， [5], 针对不同的函数组合和 α值进行比较。与BBOB函数的常见做法一致，我们采用51个在10^2 至 10^{-8} 之间按对数间隔分布的目标值来计算ECDF。得到的曲线下面积（AUC）经过归一化，因此，若某个算法在首次评估时就达到所有目标值，其AUC将为1。图的上方（ $\alpha = 0$）展示了球体函数上的性能，CMA-ES在该函数上表现出色。不过，各列之间仍存在差异，因为仿射函数组合的位置设定为第二个函数的最优点。

从图2可以看出，CMA-ES的性能似乎确实会在球体函数与其组合的函数之间平滑过渡。不过，过渡速度的差异值得关注。例如，函数3和函数11上的最终性能似乎相近，但过渡速度却有显著差异。这似乎表明，对于F11而言，加入一些全局结构对CMA-ES所面临的景观挑战影响相对较弱；而对于F3，即使加入少量全局结构，也会显著简化其景观。

我们可以对其他优化算法进行类似的分析。在图3和图4中，我们分别展示了差分进化和 Cobyla 对应的热力图，与图2相同。从这些热力图可以清楚地看出，差分进化的性能比 CMA-ES 更不稳定，而 Cobyla 的性能下降得快得多。差分进化的整体趋势线确实与对角协方差矩阵自适应进化策略的趋势线有些相似：图3中高、低曲线下面积之间的转变点，与图2中的转变点相近。不过，算法表现仍存在一些差异，尤其是与 Cobyla 相比时。这些差异也引出了一个问题：算法之间的排名是否也存在转变点？具体来说，如果某个算法在 $\alpha = 0$ 时表现良好，但在 $\alpha \to 1$时被其他算法超越，那么探究排名的这一变化将有助于进一步了解所考察算法各自的优势与不足。

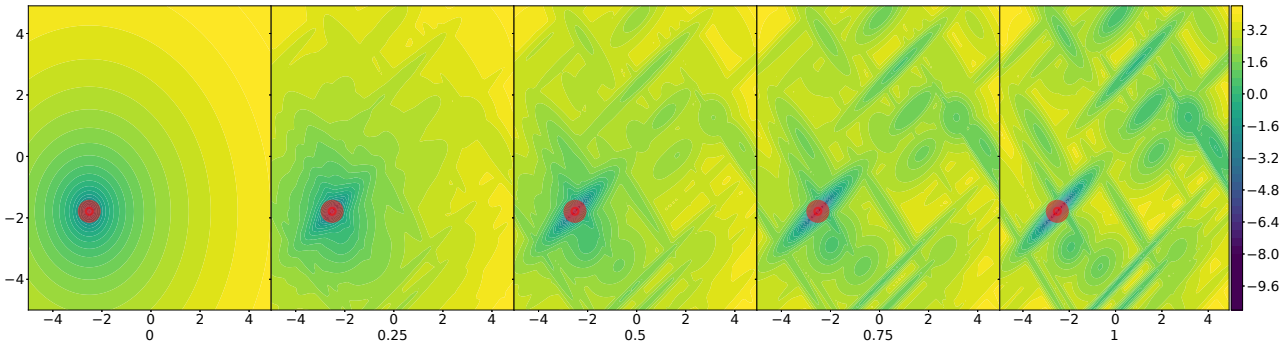

图 1：F21（$\alpha = 1$）与 F1（$\alpha = 0$）的仿射组合景观（函数值取对数尺度）随 α变化的演变。两个函数均使用实例 1。红色圆圈标出了全局最优点的位置。

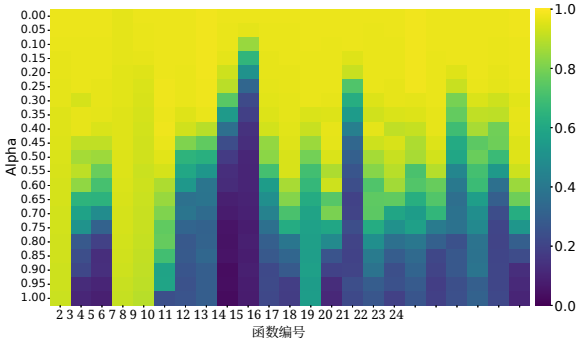

图2：给定 α（y 轴）时，针对每个 BBOB 函数（x 轴）与球体模型的组合，Diagonal CMA-ES 的 ECDF 曲线下归一化面积。AUC 根据 10 个实例上的 50 次运行，在10000 次函数评估后计算。

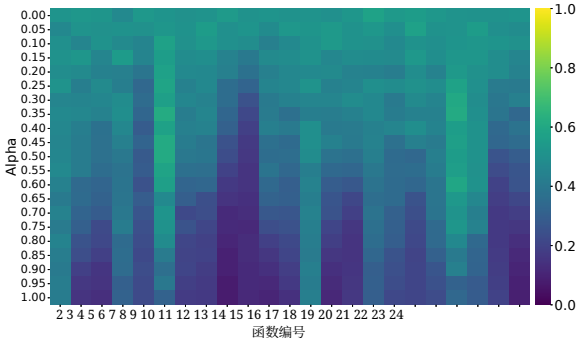

图3：给定 α（y 轴）时，针对每个 BBOB 函数（x 轴）与球体模型的组合，差分进化的 ECDF 曲线下归一化面积。AUC 根据 10 个实例上的 50 次运行，在10000 次函数评估后计算。

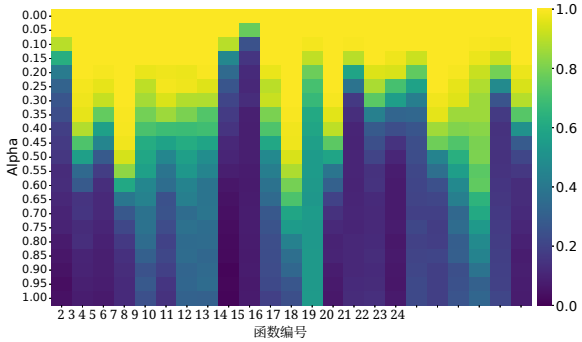

图4：对于给定的 α 值（y轴），Cobyla 在 BBOB 函数与球体模型的每种组合（x轴）上的 ECDF 曲线下面积归一化值。在10000 次函数评估后，根据在 10 个实例上各运行 50 次的结果计算曲线下面积。

为回答算法相对排名这一问题，我们采用由5种算法组成的算法组合，并根据它们在每种仿射函数组合上的曲线下面积进行排名。随后，我们在图5中展示每种设定下排名第一的算法。需要注意的是，在所选预算下，粒子群优化和估计分布算法从未排名第一，因此不会显示在图中。

从图5中可以清楚地看到，Cobyla 能很好地处理球体模型；当球体的权重相对较高时，它的表现优于其他算法。随后，超过某个阈值后，CMA-ES 的表现始终优于算法组合中的其他算法。然而，随着 α 进一步增大，球体模型的影响减弱，似乎出现了一种有趣的模式。对于若干问题，算法还会第二次转换为差分进化或 Cobyla。对于某些函数，例如 F3 和 F4，这种现象的一个可能解释因素是局部最优值的强度增加，使 CMA-ES 更难探索整个景观；而差分进化采用均匀初始化，因此受到的影响略小。

为了更好地理解算法排名的变化情况，我们可以聚焦于其中一个函数，并

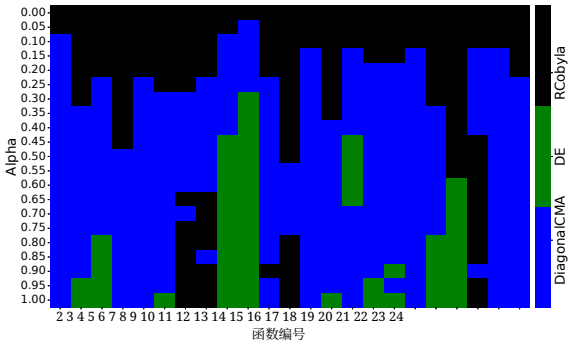

图5：对于给定的 α 值，针对 BBOB 函数与球体模型的每种组合（横轴），ECDF 曲线下面积最高的算法（纵轴）。在进行10000 次函数评估后，根据 10 个实例上的 50 次运行计算 AUC。PSO 和 EMNA 未显示，因为它们从未排名第一。

绘制若干个 α取值下的期望运行时间（ERT）。图6展示了这一结果，其中我们考察了F10与球体模型的组合。可以清楚地看到，Cobyla在优化球体模型方面非常有效，其求解速度几乎比排名第二的算法DiagonalCMA快一个数量级。然而，当 α 增大时，Cobyla很快便开始失效，而DiagonalCMA仍能在 $\alpha = 0.25$ 下以相近的评估次数求解大多数实例。不过，从图中的分岔可以明显看出，对于某些实例，DiagonalCMA已无法在分配的预算内找到最优点。当 α 进一步增大时，这三个算法都再也无法求解任何实例。当$\alpha \geq 0.75$时，我们看到差分进化超过了另外两种算法，这也解释了图5中观察到的排名优势。

5 不同函数组之间的组合

将函数与球体模型组合，可以看作是为问题添加全局结构；而其他函数之间的组合，则能为不同类型问题之间的转变点提供有趣的见解。为了展示这些组合能够带来哪些见解，我们选取 5 个函数子集，并收集每种组合在相同 21 α 值下的性能数据（包括函数的两种排列顺序）。图 7 展示了对角协方差矩阵自适应进化策略在这些函数组合上的性能，以归一化曲线下面积表示。请注意，对于 $\alpha = 1$，我们使用列标签中指定的函数；对于 $\alpha = 0$，我们使用行标签中指定的函数，但采用列函数的最优点。

从图 7 可以看出，两个极端 α 值之间的性能变化大多较为平滑。虽然也存在一些变化相当迅速的情况，例如 F2 与 F11 之间的过渡，但这些似乎只是例外，而非常态。特别值得关注的是，在某些设置下，两个函数的仿射组合的性能明显比参与组合的函数更容易或更难。这种情况例如出现在 F21 和 F9 的组合中。值得注意的是，该函数组合沿对角线镜像后的组合并未表现出类似的行为。事实上，总体而言，图 7 并非完全关于对角线对称。

我们可能会预期（F_1，F_2，α）与（F_2，F_1，$1-\alpha$）相似。然而，F9 与 F21 的组合表明，情况并非总是如此。具体来说，组合（F_{21}，F_9，1）的曲线下面积显著劣于（F_9，F_{21}，0），尽管 F21 并未直接影响仿射组合的函数值。这两个问题唯一的区别在于最优点的位置。对于 F21，最优点的默认位置被硬编码为距最优点 [13]，1 个单位，而 F9 并非如此。由于 CMA-ES 会将质心初始化在空间原点，并使用默认初始步长 0.3 [19]，因此它能够在默认设置下找到最优点；而函数的平移版本则更具挑战性。这凸显了传统 BBOB问题性能分析中一个潜在的问题：如果分析时未考虑最优点位置等内置限制，就有可能误读具有结构偏差的算法 [23] 的结果，将其视为在这类多模态问题上表现最优，尽管它无法求解同一函数的平移版本。

为了了解这种初始化对性能差异的影响有多大，我们使用 CMA-ES 的另一个版本进行了额外实验。我们选择使用模块化协方差矩阵自适应进化策略 [2]，并将初始步长设为定义域范围的 0.2 倍，因此在本例中为 2。实验结果如图8所示。从图中可以清楚地看出，这种 CMA-ES 设置的整体表现更好。尤其值得注意的是，不对称性有所减弱，尽管并未完全消失。

此外，图8还显示了若干对角协方差矩阵自适应进化策略中未曾出现的有趣性能趋势。例如，F2 和 F9 的组合在中心附近的曲线下面积大幅下降，尽管对于这个版本的 CMA-ES 而言，这两个函数单独看来都相对容易求解。两个版本的 CMA-ES 之间存在明显差异，但许多趋势在某种程度上也同样出现在图7中，例如 F11 和 F16 组合的性能下降。

作为最后一种算法，我们在同一组函数组合上运行差分进化。结果如图9所示。从图中可以看出，差分进化的整体性能确实不如两个版本的 CMA-ES。值得注意的是，其对角线方向上的不对称程度低于 DiagonalCMA-ES。这可能是初始化方式发生变化所致（CMA-ES 使用高斯分布，差分进化使用均匀分布），从而减小了初始时向空间中心偏移的偏差。另一个需要考虑的因素是性能方差。对于 CMA-ES，性能会随着α的变化而显著波动；而对于差分进化，AUC 的变化似乎小得多。

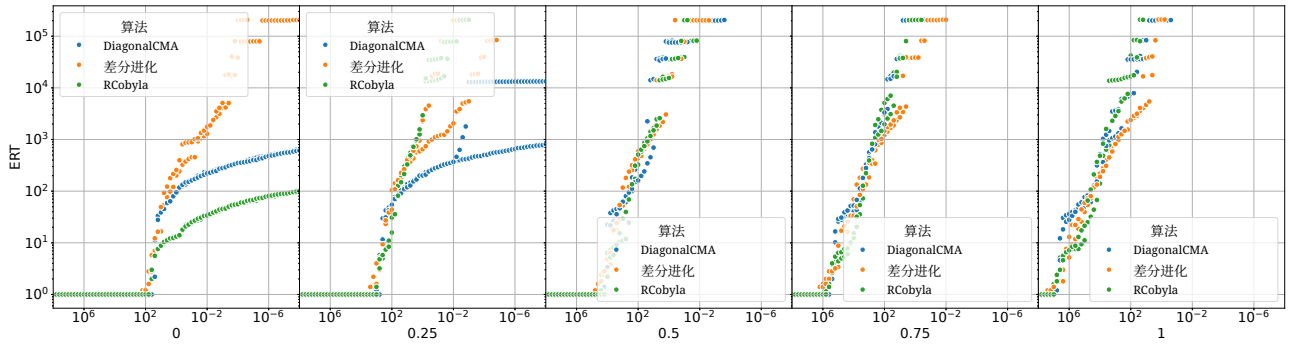

图 6：在 F10 ($\alpha = 1$) 与 F1 ($\alpha = 0$) 之间的仿射组合上，三种算法针对选定的 α 值的每实例期望运行时间。每个点表示基于 1 个实例上的 5 次运行计算得出的期望运行时间，共计 10 个实例。

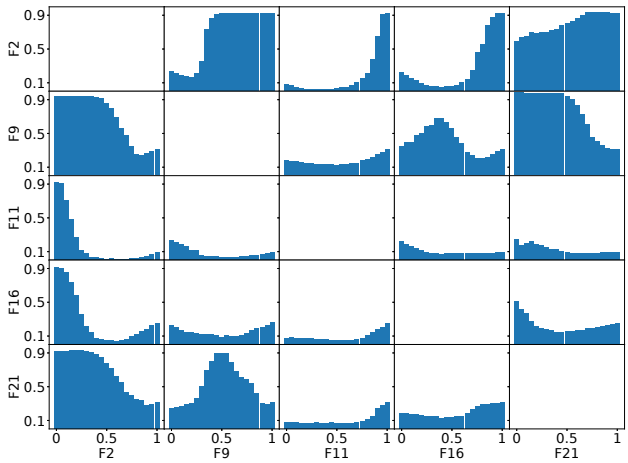

图 7：Diagonal CMA- ES 在所选 BBOB 问题之间的各个仿射组合上的 ECDF曲线下面积。每个分面对应行函数与列函数的组合，x 轴表示所用的 α。AUC值基于 5 个实例上的 50 次运行计算，预算为10000 函数评估次数。

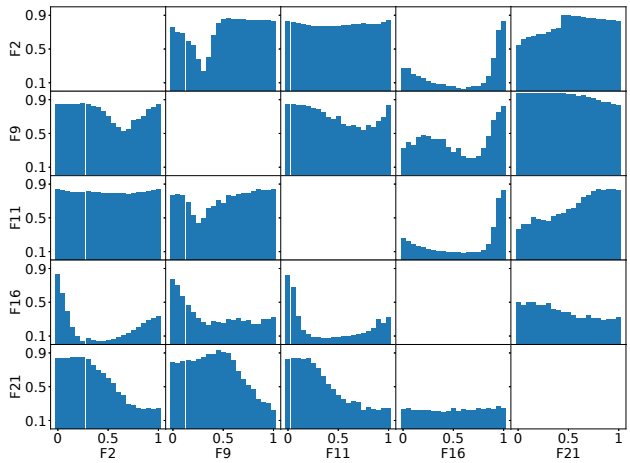

图8：在所选 BBOB问题之间的每种仿射组合上，模块化协方差矩阵自适应进化策略的 ECDF曲线下面积。每个分面对应行函数与列函数的组合，x 轴表示所用的 α。AUC值基于 5 个实例上的 50 次运行计算，预算为10000 次函数评估次数。

6 深入考察单函数组合

为了进一步分析改变函数组合权重的影响，我们可以聚焦于某一个特定组合，并对其进行更详细的研究。首先，我们评估使用不同实例衡量性能所产生的影响。具体做法是，在图10中按实例分别考察 F2 至 F16 这一特定函数组合的 AUC值分布。从图中可以看出，总体而言，AUC值分布相当稳定。不过，在 CMA-ES 变体的转折点附近，即 $\alpha \approx 0.8$左右，方差明显增大。为确认其他函数组合中是否也存在这种现象，我们针对 F21 和 F9 的组合绘制了相同的图，如图 11 所示。从图中可以看到类似的模式

对于对角协方差矩阵自适应进化策略，高 α 值处的 AUC 分布范围几乎从 0 到 1。

图 10 中观察到的方差可能表明，为了稳定地了解这一转折点处的确切行为，应使用更多样化的实例，以获得更稳健的性能估计。不过，考虑到图 11 中观察到的 AUC 差异极大，这种方差促使我们更详细地研究实例生成过程（例如，最优解的位置）与所用算法的搜索行为之间的相互作用。

除了实例生成过程之外，分析优化算法在这些仿射函数组合上的性能时，还需考虑另一个重要因素：目标值的尺度。虽然通常会忽略

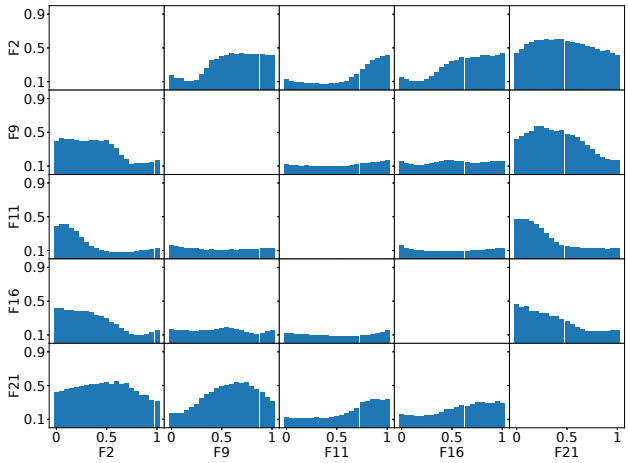

图9：差分进化在所选 BBOB问题之间各仿射组合上的ECDF曲线下面积。每个分面对应行函数与列函数的组合，x 轴表示所用的 α。AUC值基于 5 个实例上的 50 次运行计算，预算为 10000 次函数评估。

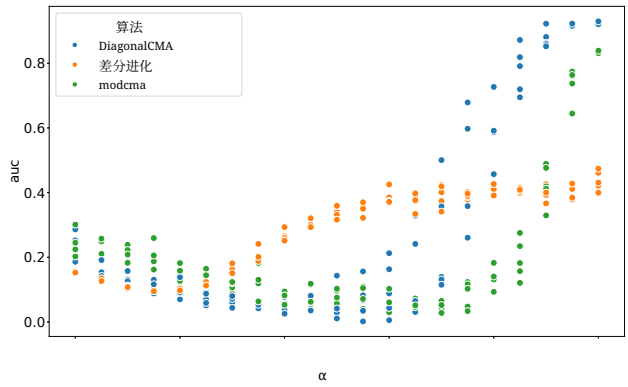

图 10：所选算法在 F2 与 F16 的仿射组合上的各实例归一化曲线下面积值分布。AUC值基于 5 个实例上的 50 次运行计算，预算为 10000 次函数评估。

尺度，以便使用相同的目标值（与最优点的精度）——例如计算汇总的 ECDF 曲线，但不同问题对目标值进行尺度变换的方式，确实会影响我们对结果的解读。在考察这些问题的仿射组合时，这一点会变得愈发明显。图 12 展示了对角协方差矩阵自适应进化策略在 F16 与 F11 组合上的收敛图。从曲线左侧可以清楚地看到，不同组合得到的初始值差异很大：$\alpha = 0$ 时为 10_7，而 $\alpha = 1$ 时为 10_2。然而，尺度变化并非影响

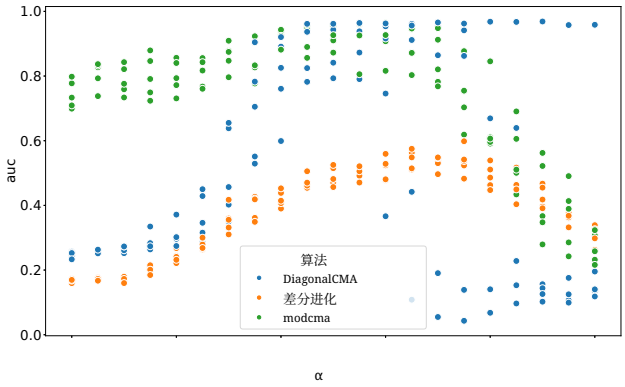

图 11：所选算法在 F21 与 F9 的仿射组合上的各实例归一化曲线下面积值分布。AUC值基于 5 个实例上的 50 次运行计算，预算为 10000 次函数评估。

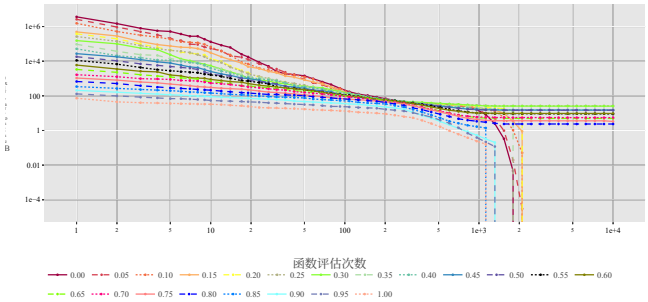

图 12：模块化协方差矩阵自适应进化策略在 F16 和 F11 的仿射组合上找到的函数值几何平均数的演变；两个函数均为实例 1。每条曲线对应指定 α 下的 50 次运行。

性能。初始化后，曲线的形状发生了明显变化，这与图 7 中观察到的曲线下面积变化一致。

为了探究这种行为变化的原因，我们可以研究对角协方差矩阵自适应进化策略在这些函数上的优化轨迹。由于无法在原始的 5 维空间中将其可视化，我们改为在这些函数的二维版本上重复收集数据。在图 13 中，我们展示了 F11 与 F16 在若干个 α取值下的仿射组合景观。我们标出了对角协方差矩阵自适应进化策略在该实例的 50 次运行中各自找到的最佳点。这张图清楚地展示了原始问题之间的尺度差异。此外，我们看到，随着 α 越来越接近 1，算法陷入局部最优点的次数越来越少。F11 所带来的全局结构足以引导 CMA-ES 到达包含全局最优点的区域。然而，当 F11 的影响过大时，找到正确搜索方向的困难会显著影响收敛行为。因此，α 更接近

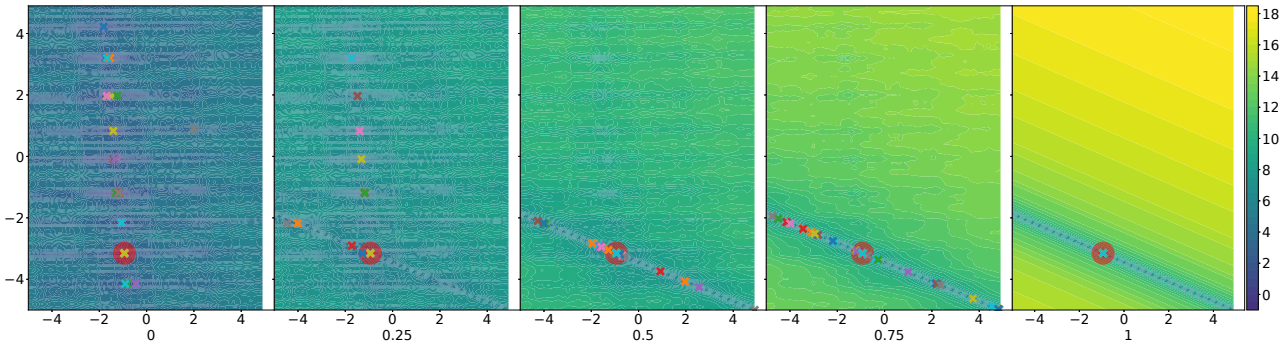

图 13：F11 ($\alpha = 1$) 与 F16 ($\alpha = 0$) 的仿射组合景观（函数值取对数尺度）随 α 变化的演变情况。两函数的实例均为 1。红色圆圈标出了全局最优点的位置。十字标记表示模块化协方差矩阵自适应进化策略在 50 次运行中各自找到的最佳点。

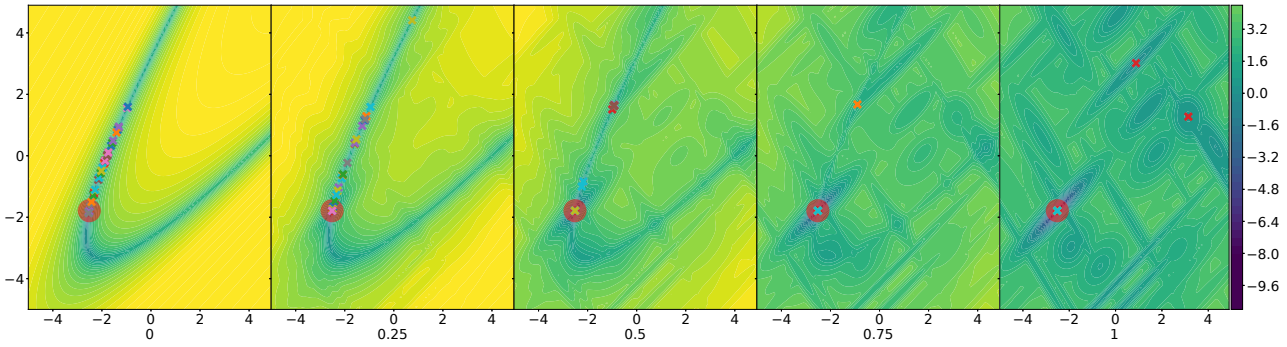

图 14：F21 ($\alpha = 1$) 与 F9 ($\alpha = 0$) 的仿射组合景观（函数值取对数尺度）随 α 变化的演变情况。两函数的实例均为 1。红色圆圈标出了全局最优点的位置。十字标记表示对角协方差矩阵自适应进化策略在 50 次运行中各自找到的最佳点。

0.5 的取值似乎能兼顾 F16 的多峰性与 F11 带来的挑战，使其成为 CMA-ES 难以解决的问题。

虽然 F11 与 F16 的组合似乎会生成更具挑战性的函数，但图 7 显示，也存在相反的函数组合。F9 与 F21 的组合呈现出有趣的行为。尽管初始化的执行方式可能解释了（F_9，F_{21}，0）与（F_{21}，F_9，1）之间的不对称性，但这无法解释 α 接近 0.5 时曲线下面积的增加。我们在图 14 中展示了景观的变化，以及对角协方差矩阵自适应进化策略找到的相应解。在该图中，我们可以看到，当 $\alpha = 0$ 时，CMA-ES 会在函数的脊线上找到解，但大多数运行次数在给定预算内都无法达到最优点。这表明，F9 的特有难点——算法必须持续调整其搜索方向 [7], ——阻碍了所使用的对角 CMA-ES 的收敛。然而，随着 α 增加，F21 的结构逐渐加入其中，从而增加了算法接近最优值的方式。对于 $\alpha = 1$ 而言，F21 的多峰性完全占据主导，导致部分运行次数陷入局部最优值，进而降低算法的性能。这表明以这种方式组合这两个函数，会生成一个函数，使二者原有的困难以相互抵消的方式结合起来，而 CMA-ES 则利用了这一点。

7 结论与未来工作

BBOB 问题的仿射组合为研究优化算法的行为提供了一种新方法。我们展示了如何将任意函数与球体模型组合，以确定增加全局结构对一组算法性能的影响。此外，将具有不同高层特征的函数组合起来，使我们能够观察不同优化挑战之间的转变。尽管本研究并不全面，但它凸显了利用这些新函数组合来理解优化算法行为的潜在益处。

然而，在分析选项方面获得这些益处的同时，也带来了若干需要考虑的挑战。我们确定了以下几个方面：

尺度。 正如提出这些组合时所指出的 [4], ，两个问题之间的尺度差异可能相当显著。

尽管我们试图通过采用对数尺度的权重来减轻这种影响，但从实验结果可以明显看出，尺度仍在很大程度上影响着我们对性能的解读。如何在组合两个函数的景观时保持函数值范围一致，仍是一个有待解决的问题。

实例。 BBOB 套件基于这样一种理念构建：每个函数都可以通过多种方式生成实例。这是通过多种变换实现的，其中最常见的是将最优点移至定义域中的其他位置。我们展示的结果表明，选择这些最优位置的方式会对优化算法的性能产生重大影响。由于最优点在定义域中并非均匀分布，某些函数会呈现不同类型的偏差，算法可以利用这些偏差。因此，在使用函数组合时，如何公平地考虑不同的实例生成机制，与在一组 BBOB 实例上观察到的性能能否良好泛化这一问题密切相关。

即便存在这些挑战，仿射函数组合仍有许多潜在用途。它们的一个重要用途是训练算法选择模型 [11],，因为它们能够显著增加训练数据的规模和多样性，而这对于检验泛化能力至关重要。

这些函数组合的基准测试数据还可以通过与启发其创建的探索性景观分析联系起来，进一步加以利用。由于这些组合能够平滑地填充景观特征空间，因此可以将其与算法性能结合起来，更细致地了解景观与不同算法之间的相互作用方式 [9, 22]。

致谢

我们的工作获得了 ANR-22-ERCS-0003-01 项目 VARIATION 和 CNRS INS2I 项目 IOHprofiler 的经费支持。本研究使用了莱顿大学提供的 ALICE 计算资源。

参考文献

[1] Thomas Bartz-Beielstein、Carola Doerr、Jakob Bossek、Sowmya Chandrasekaran、Tome Eftimov、Andreas Fischbach、Pascal Kerschke、Manuel López-Ibáñez、Kather- ine M. Malan、Jason H. Moore、Boris Naujoks、Patryk Orzechowski、Vanessa Volz、Markus Wagner 和 Thomas Weise。2020年。《优化基准测试：最佳实践与未决问题》。*CoRR* abs/2007.03488（2020）。arXiv:2007.03488 https://arxiv.org/abs/ 2007 03488 。Jacob de Nobel、Diederick Vermetten、Hao Wang、Carola Doerr 和 Thomas Bäck。2021年。调优：评估新理念与现有算法模块协同效益的一种手段。载于遗传与进化计算会议论文集（*GECCO'21*，伴随材料），Krzysztof Krawiec（编）。ACM，1375–1384。 https://doi.org/10.1145/3449726.3463167[3] Jacob de Nobel、Furong Ye、Diederick Vermetten、Hao Wang、Carola Doerr 和 Thomas Bäck。2021年。I OHexperimenter：迭代优化启发式算法基准测试平台。*CoRR* abs/2111.04077（2021）。arXiv:2111.04077 https: //arxiv.org/abs/ 2111 04077 。Konstantin Dietrich 和 Olaf Mersmann。2022年。通过仿射重组增加基准函数集的多样性。载于自然界并行问题求解 - *PPSN XVII* - 第*17*届国际会议，*PPSN 2022*，德国多特蒙德，*2022*年*9*月*10*日至*14*日，论文集，第一部分（计算机科学讲义，第*13398*卷），Günter Rudolph、Anna V. Kononova、Hernán E. Aguirre、Pascal Kerschke、Gabriela Ochoa 和 Tea Tusar（编）。施普林格，590–602。https://doi.org/10.1007/978-3-031-14714-2_41 [5] Nikolaus Hansen、Anne Auger、Dimo Brockhoff 和 Tea Tušar。2022年。黑盒优化基准测试中的任意时刻性能评估。*IEEE*进化计算汇刊26卷6期（2022），1293–1305。[6] Nikolaus Hansen、Anne Auger、Raymond Ros、Olaf Mersmann、Tea Tušar 和 Dimo Brockhoff。2021年。COCO：用于黑盒环境下比较连续优化器的平台。《优化方法与软件》36卷1期（2021），114–144。[7] Nikolaus Hansen、Steffen Finck、Raymond Ros 和 Anne Auger。2009年。实数参数黑盒优化基准测试 *2009*：无噪声函数定义。技术报告 RR-6829。法国国家信息与自动化研究所。https://hal.inria.fr/inria-00362633/ document[8] Nikolaus Hansen 和 Andreas Ostermeier。2001年。进化策略中的完全去随机化自适应。进化计算9卷2期（2001），159 195 10 1162 1063656017501903 98 – 。https://doi.org/ 。/ Anja [9] 2020 Jankovic 和 Carola Doerr。 。面向模块化协方差矩阵自适应进化策略变体的景观感知固定预算性能回归与算法选择。载于*2020*年遗传与进化计算会议论文集。ACM，841–849。[10] James Kennedy 和 Russell Eberhart。1995年。粒子群优化。载于国际神经网络会议论文集（*ICNN'95*），澳大利亚西澳大利亚州珀斯，*1995*年*11*月*27*日至*12*月*1*日。IEEE，1942–1948。https://doi.org/10. 1109/ICNN.1995.488968[11] Pascal Kerschke、Holger H. Hoos、Frank Neumann 和 Heike Trautmann。2019年。自动算法选择：综述与展望。进化计算 27, 1 2019 3 45 10 1162 _ _00242 ( ), – 。 [12] 2001 https://doi.org/ 。/evco a Pedro Larrañaga 和 Jose A Lozano。 。分布估计算法：进化计算的新工具。第2卷。施普林格科学与商业媒体。[13] Fu Xing Long、Diederick Vermetten、Bas van Stein 和 Anna V. Kononova。2022年。BBOB实例分析：跨问题实例的景观特性与算法性能。*CoRR* abs/2211.16318（2022）。 https://doi.org/10.48550/ arXiv.2211.16318 arXiv:2211.16318[14] Manuel López-Ibáñez、Juergen Branke 和 Luís Paquete。2021年。进化计算中的可复现性。《美国计算机协会进化学习与优化汇刊》1卷4期（2021），1–21。[15] Olaf Mersmann、Bernd Bischl、Heike Trautmann、Mike Preuss、Claus Weihs 和 Günter Rudolph。2011年。探索性景观分析。载于遗传与进化计算会议论文集（*GECCO'11*）。ACM，829–836。[16] Mario Andrés Muñoz、Michael Kirley 和 Kate Smith-Miles。2022年。分析随机性效应对探索性景观分析可靠性的影响。《自然计算》21卷2期（2022），131–154。[17] Mario A Muñoz、Yuan Sun、Michael Kirley 和 Saman K Halgamuge。2015年。黑盒连续优化问题的算法选择：方法与挑战综述。《信息科学》317（2015），224–245。[18] Michael JD Powell。1994年。一种通过线性插值对目标函数和约束函数建模的直接搜索优化方法。施普林格。[19] J érémy Rapin 和 Olivier Teytaud。2018年。Nevergrad：无梯度优化平台。https://GitHub.com/FacebookResearch/Nevergrad。[20] Quentin Renau、Johann Dréo、Carola Doerr 和 Benjamin Doerr。2021年。迈向可解释的探索性景观分析：用于对BBOB函数分类的极端特征选择。载于进化计算应用：第*24*届国际会议，*EvoApplications 2021*，作为*EvoStar 2021*的一部分举行，线上活动，*2021*年*4*月7日至*9*日，论文集*24*。施普林格，17–33。[21] Rainer Storn 和 Kenneth Price。1997年。差分进化——一种用于连续空间全局优化的简单高效启发式算法。《全球优化杂志》11卷4期（1997），341。[22] Risto Trajanov、Stefan Dimeski、Martin Popovski、Peter Korošec 和 Tome Efti- mov。2021年。可解释的景观感知优化性能预测。载于*2021*年*IEEE*计算智能研讨会系列（*SSCI*）。IEEE，01–08。[23] Diederick Vermetten、Bas van Stein、Fabio Caraffini、Leandro L. Minku 和 Anna V. Kononova。2022年。BIAS：用于对连续域中的结构偏差进行基准测试的工具箱。*IEEE*进化计算汇刊26卷6期（2022），1380–1393。https://doi.org/ 10 1109 2022 3189848 。/TEVC. 。Diederick Vermetten、[24] Furong Ye 和 Carola Doerr。2023年。可复现性文件和补充图表。代码与数据仓库：https://doi.org/10.5281/zenodo. 7629706 图表仓库：https://figshare.com/s/68587b6a82d9c6e5eccf。[25] Hao Wang、Diederick Vermetten、Furong Ye、Carola Doerr 和 Thomas Bäck。2022年。IOHanalyzer：迭代优化启发式算法的详细性能分析。*ACM*进化学习与优化汇刊2卷1期（2022），3:1–3:29。https://doi.org/ 10.1145/3510426 IOHanalyzer可在CRAN、GitHub和基于网页的图形用户界面中使用，链接见https://iohprofiler.github.io/IOHanalyzer/。