大 连 理 工 大 学 本 科 外 文 翻 译

进化算法的另类基准集构建

On constructing alternative benchmark suite for evolutionary algorithms

学 部（院）：软件学院
专　　业：数字媒体技术
学 生 姓 名：张金哲
学　　号：201692117
指 导 教 师：任志磊
完 成 日 期：2020 年 2 月 25 日

进化算法的另类基准集构建

Yang Lou , Shiu Yin Yuen
香港大学电子工程系

摘要：

基准测试为一个进化算法在应用之前提供了性能度量。本文提出了一种构建基准测试集的系统方法。本文使用了一组已建立的算法，对于每一个算法，进化都会产生一个唯一容易的问题实例。结果实例由一个新的基准测试套件组成。每一个问题实例只对一个算法有利（唯一简单）。设计了一种基于统计测试结果的层次适应度分配方法，用于为算法生成唯一的简单（或困难）问题实例。实验结果表明，每种算法对其唯一有利问题的鲁棒性最好。测试结果是可重复的。算法性能在套件中的分布是无偏的（或均匀的），它模拟了真实世界中均匀分布的问题的任何子集。结果套件提供了 1）进化算法的替代基准套件；2）访问新算法的新方法；3）对于进化算法选择和组合有意义的训练和测试问题。

关键词：进化算法；基准实例生成；统计检验；层次适应度；算法性能度量。

1. 绪论

基准测试套件应该对正在测试的进化算法（EA）给出客观、公平的性能度量，以便能够谨慎、合理地推断其在其他未测试问题上的优化能力。否则，EA 的性能可能被夸大或低估，从而导致误用。除了测量，基准通常用于训练进化算法组合（EAP）[29,40,44-46]。由于理论研究比较困难，因此通过对许多问题实例的测试，对算法的优缺点进行了实证研究[31,33,35]。一个好的基准套件应该能够代表现实世界中的问题。然而，实际上不可能表示现实世界问题的集合，因此，覆盖整个现实世界问题域的特定目标部分的基准套件变得具有实际意义。例如，在[30]中提出了一个多模态基准套件来表示工程优化问题中的多模态。在本文中，我们提出了一个演化的方法来组成一个基准测试套件，这是一个替代现有套件的方法。所提出的基准测试套件是基于性能而不是基于问题的特性来组成的。使用一组已建立的 EA，并为每个 EA 演化出一个唯一有利的问题。因此，每个 EA 在测试套件中的一个且唯一的问题上分别比任何其他 EA 执行得唯一（并且显著地）好。与现有的测试套件相比，这种测试套件不支持任何采用的算法。该套件以所采用的算法为参考，便于测量新算法的性能。

常见的基准问题有四类：1）每年提出的竞赛测试套件，如 IEEE 进化计算大会（CEC）基准[22,30,38]；黑箱优化基准（BBOB）[48]；2）在有影响力的文章中使用的著名基准问

题，如[43]；3） 可调基准生成器，例如高斯（MSG）景观生成器的最大集[9]、多模景观生成器[31]、NK[3]和 NM[28]生成器；以及 4）现实世界的优化问题[4]。其中，可调发电机是灵活的，能够涵盖广泛的问题。然而，参数调整[19]是一个阻碍可调生成器广泛使用的问题。请注意，这个问题也存在于其他基准类别中，这表现为难以选择有意义的问题实例。此外，每个可调谐发生器都有其自身的局限性，例如高斯发生器旨在构建超钟形峰或谷，但不擅长构建高原和平原盆地等景观[9]。

无免费午餐（NFL）定理[41]表明，在所有问题的平均值上，没有任何算法优于任何其他算法。因此，在基准测试套件中胜出的 EA 并不意味着它在所有问题上都表现良好。

在本文中，我们使用了几个已建立的 EAs，用集合 $S_A=\{A_1, \cdots, A_N\}$表示。对于每个算法 A_i（i=1，…，N），使用 EA 生成一个唯一简单的问题实例。对于 A_i 来说唯一简单的方法是，只有 A_i 在问题实例上表现良好，而所有其他算法 $A_j \neq A_i$ 的表现都不如 A_i。同样，对于 A_i 来说，一个独特的难题意味着只有 A_i 在这个问题上表现最差，而所有其他算法 $A_j \neq A_i$ 的表现都优于 A_i。利用本文的方法可以构造一个由 N 互补问题组成的基准集，每个问题对于算法来说都是唯一容易的，而对于其他算法来说则是困难的。此外，应用统计测试使得算法之间的性能差异显著，从而确保结果在统计上是可重复的。

本文的主要贡献包括：1）采用多比较难度测量方法生成基准。2） 提出了一种新的层次适应度分配方法，使多个比较难度能够被精确地测量和分配。3） 将统计测试引入到基准生成中，从而在统计上保证了问题的难度。该方法可用于生成算法的唯一简单问题实例和唯一困难问题实例。

论文的其余部分安排如下：第 2 节对现有的工作进行了概述。第 3 节描述了拟议工作的详细情况。在第 4 节中，进行了模拟和比较。第 5 节进一步讨论，第 6 节结束本文。

2. 现有工作调查

不鼓励为可调问题实例生成器随机选择参数（例如[3,4,28,31]），因为它们容易缺乏多样性[16]。另一方面，使用进化计算生成所需的基准实例[2,13,15,21,32]允许在问题空间中获得简单和困难的实例，因为 EA 是为搜索极值而设计的[5]。然而，为基准测试而生成有意义的问题实例的现有工作很少。

在[13]中，EA 用于生成比常用测试套件更困难的组合基准问题。在问题演化过程中，只使用一种算法，通过单次搜索（即搜索操作的次数）来度量问题的难度，以获得用户定义精度的结果。因此，问题难度的测量可能受到随机因素（例如，随机种子）的严重影响。

在[21]中，遗传规划被用来进化离散化的问题实例。采用两种算法，通过最大化两种算法的性能差异来生成每个结果实例。不考虑三个或更多 EAs 的性能比较。在这项工作中，EA 性能被认为是一个噪声函数[17]。将适应度（两个 ea 之间的性能比较）分配给实例时，结果值是五次重复运行的平均值。此外，每一个实例都被独立地分配了两次适应度，第一次是为了生存而竞争，第二次是为了选择父母来繁殖。虽然这种显式的五次运行平均法稍微降低了噪声，但结果没有统计保证（例如，两种算法之间的性能差异在统计上是显著的）。

为了为多个算法中的一个算法生成唯一容易或困难的问题实例，性能比较应该扩展到三个或更多算法。在[32]中，使用多目标适应度赋值，其中每个目标是与[21]中相似的成对比较。例如，当搜索一个唯一易于算法 A_1 但难以算法 A_2 和 A_3 的实例时，一个实例同时将两个目标最大化。第一个目标是 A_1 和 A_2 之间的性能差异，第二个目标是 A1 和 A_3 之间的性能差异。或者，在[34]中，这样的实例是通过最大化 A_1 与 A_2 和 A_3 的平均性能之间的性能差来进化的。然而，当参与算法的数量增加时，第一种方法需要进行多目标优化，难度也随之增加。对于第二种方法，平均值不能代表每个算法的实际性能。

EA 是随机方法，因此它们的结果（在给定数量的评估中获得的最佳结果，或用于搜索给定精度的结果的评估总数）不应被视为确定性值。据我们所知，除了[21]之外，以前的工作没有把算法性能看作是一个统计问题。在[21]中，五次运行的平均值可以减少随机性，但不产生统计保证。此外，以往的工作都没有充分处理多算法性能比较的问题。注意，统计测试通常用于 EA 性能比较，而很少用于基准问题的生成。

3. 基于层次适应度的进化基准生成器

本文提出了一种基于层次适应度的进化基准生成器（HFEBG），其目标是：1）以层次化的方式有效地处理多个 EA 比较；2）使用统计测试处理 EA 得到的结果，从而给出统计保证。

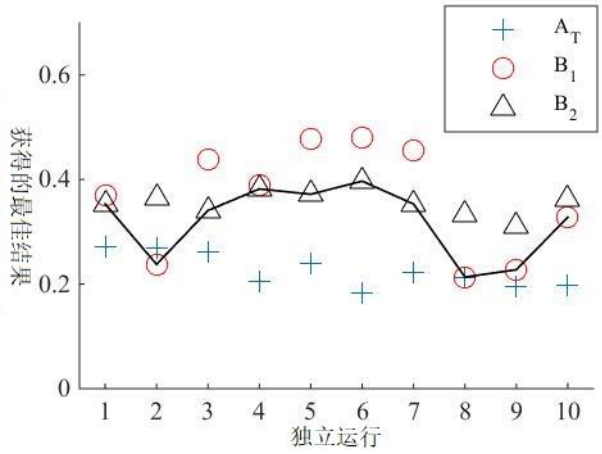
3.1. 统计检验与层次适应度

在不丧失一般性的前提下，我们将本文中提到的所有优化问题都看作是最小化问题。我们使用 N 个建立的 EA，用 $\mathbf{S_A}=\{A_1, A_2, \cdots, A_N\}$表示（粗体表示向量或矩阵）。对于每个算法 A_i（当选择作为目标算法时表示为 A_T），进化出唯一容易的问题实例，使得 A_T 优于 $\mathbf{B_A}=\mathbf{S_A}-\mathbf{A_T}=\{B_1, B_2, \cdots, B_{N-1}\}$中所有算法。$\mathbf{B_A}$ 可以被认为是一组被 A_i 打败的算法。

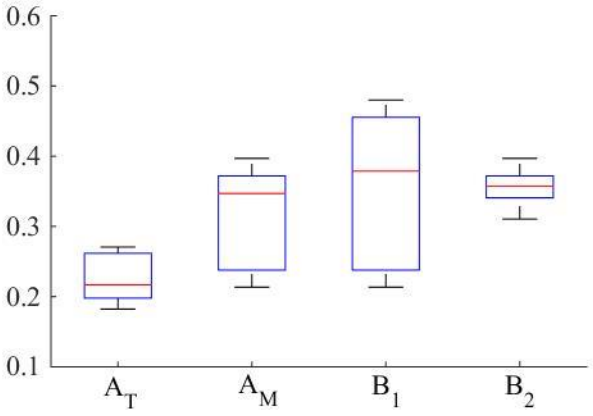
$\mathbf{B_A}$ 中的每个 EA 都被认为同等重要。NBIPOP-aCMAES[24]说，直观地说，人们可能认为，比典型的类 EA 遗传算法更值得注意的是击败最近提出的 EA。我们的假设基于以下三个事实：1）上面讨论的 NFL 定理[42]；2）[12]中的理论表明一个算法容易出问题，另一个算法可能困难，反之亦然；3）不知道关于这些问题的先验知识。这种假设使得将适应度分配给 A_T 优于 B_A 的一部分的实例变得容易。由于假设 3），焦点变成 A_T 击败了多少 EA，而不是哪个 EA。

从高优先级到低优先级，我们按层次顺序考虑以下三个事件：1）A_T 在 $\mathbf{B_A}$ 中优于所有 EA；2）A_T 在 $\mathbf{B_A}$ 中优于某些但并非所有 EA；3）A_T 在 $\mathbf{B_A}$ 中优于任何 EA。层次适应度按相应的顺序分配。这将在下文详细阐述。

通过收集 $\mathbf{B_A}$ 在每次运行中获得的最佳结果，形成一个元启发式 A_M。将 A_T 与 A_M 进行比较显然是“不公平的”，因为 A_M 的评估次数是 A_T 的 N-1 倍。然而，元启发式使比较变得容易：如果 A_T 的性能明显优于 A_M，那么 A_T 的性能明显优于 $\mathbf{B_A}$ 中的任何 EA。显著性通过 Mann-Whitney U 检验进行检验[32,33]，这是一种非参数统计检验，常用于检验两个 EA 所得结果之间是否存在显著差异[23]。显著性水平 AAAAA 是一个预定义的参数。给定 p 值 ρ，即 U 检验的输出，如果 ρ≤α，则确认存在显著性差异；否则（ρ>α）不存在显著性差异。



（a） 结果对比



（b） 箱线图对比

图 1 A_T 同时击败 B_1 和 B_2 的例子：
（a）三种算法得到的结果，以及为 A_M 选择的多段线链接结果；
（b）箱线图对比显示 B_1 和 B_2 都没有比 A_M 更好的性能

Kruskal-Wallis H-检验[20]提供了与 U 检验类似的非参数统计检验，但是是多样本的。不同的是，U 检验直接返回 p 值，H-test 返回 ANOVA（方差分析）表，这更难以直接比较。通过构造 A_M，将多样本统计检验简化为两样本检验。注意，U-检验（A_T，$\mathbf{B_A}$）$\leqslant\alpha$ 并不确保 H-检验（A_T，B_1，…，B_{N-1}）$\leqslant\alpha$。但是，根据经验，如果 U 检验（A_T，$\mathbf{B_A}$）$\ll\alpha$，则可以假设 H 检验（A_T，B_1，…，B_{N-1}）$\leqslant\alpha$。

图 1 显示了一个例子，其中三个 EA 计算一个问题 10 次运行，每个 EA 得到的结果绘制在图 1（a）中。A_M 在每次运行时收集 $\mathbf{B_A}=\{B_1, B_2\}$的最佳结果，$A_M$ 的结果与多段线链接。在图 1（b）中，结果用方框图进行比较。很明显，B_1 和 B_2 都不比 A_M 好，因此如果 A_T 比 A_M 好，那么 A_T 也比 B_1 和 B_2 好。因此，只需将 A_T 和 A_M 之间的性能差异最大化，A_T 和 $\mathbf{B_A}$ 之间的性能差异就会最大化。与现有方法相比，本文提出的方法 1）提供了统计保证；2）比文献[23]中的多目标方法简单；3）使用了比文献[34]中 $\mathbf{B_A}$ 的平均性能更强的条件来区分性能差异。

问题生成器用于生成问题实例。在时间步骤 t，生成的问题实例 $I^{(t)}$ 的层次适应度计算如下：

让 $\boldsymbol{S_A}=\{A_T, \mathbf{B_A}\}$中的每个算法在 $I^{(t)}$ 上独立运行 R 次。A_T 的结果用 $y_T=\{y_T^1, y_T^2, \cdots, y_T^R\}$ 表示，$\mathbf{B_A}$ 的结果用

$$Y_B = \begin{bmatrix} y_{B_1}^1 & \cdots & y_{B_1}^R \\ \vdots & \ddots & \vdots \\ y_{B_{N-1}}^1 & \cdots & y_{B_{N-1}}^R \end{bmatrix}$$

表示。其中每行 $\mathbf{y_{Bj}}=\{y_{Bj}^1, y_{Bj}^2, \cdots, y_{Bj}^R\}$，（j = 1, ⋯,N-1）是由 B_A 中的算法 B_j 获得的结果向量。A_M 的结果是从矩阵 $\mathbf{Y_B}$ 中收集的：

$$y_M = \{ y_M^1, y_M^2, \cdots, y_M^R\}$$

其中 $y_M^r = \min(y_{B1}^r, y_{B2}^r, \cdots, y_{BN-1}^r)$，（r=1,2,⋯,R）是算法 $B_1, B_2, \cdots, B_{N-1}$,在第 r 次运行时获得的最佳结果。$y_M^r$ 是 $\mathbf{Y_B}$ 的第 r 列的最小数据。计算平均值：$\bar{y}_T = \sum_{r=1}^R y_T^r$，$\bar{y}_M = \sum_{r=1}^R y_M^r$ 以及$\bar{y}_{B_j} = \sum_{r=1}^R y_{B_j}^r$，（j=1，⋯，N-1）。我们令$\bar{y}_{B,max} = \max(\bar{y}_{B_1}, \cdots, \bar{y}_{B_{N-1}})$，用以表示 B_A 中执行最差的算法的平均值。

层次适应度 H 的三个组成部分 *fit*，用 *fit.h1, fit.h2, fit.h3* 表示，并定义如下：

1） 如果$\bar{y}_T \le \bar{y}_M$（A_T 优于 A_M），则 *fit.h1* 被分配 U-检验结果（p-值）在y_T和y_M之间，而 *fit* 的其他两个分量指定为空（∅）。

2） 如果$\bar{y}_T > \bar{y}_M$且∃k≠∅，这样$\bar{y}_T \le \bar{y}_{B,k}$（$A_T$ 优于 B_A 的一部分），则 *fit.h2* 被指定为被 A_T 打败的 EA 数量的相反数，而 *fit* 的其他两个分量指定为空（∅）。例如，如果有三个 EA 被打败了，那么 *fit.h2*=-3。

3） 如果$\bar{y}_T > \bar{y}_M$且 k=∅，，这样$\bar{y}_T \le \bar{y}_{B,k}$（$A_T$ 不优于任何 B_A），则 *fit.h3* 被指定为残值$\bar{y}_T - \bar{y}_{B,max}$，而 *fit* 的其他两个分量指定为空（∅）。

表 1 三个层次适应度值的示例

	$fit.h_1$	$fit.h_2$	$fit.h_3$
fit_1	0.05	∅	∅
fit_2	∅	-2	∅
fit_3	∅	-3	∅

每个问题实例只有一个 *fit* 组件非空。层次适应度值按层次进行比较：1）如果两个实例具有不同级别的非空组件，则具有较高级别非空组件的实例更好；2）如果两个实例具有相同级别的非空组件，则具有较小值的实例获胜。例如，三个层次适应值 fit_1, fit_2, fit_3 如表 1 所示。在他们中间，fit_1 是最好的，因为它有最高级别的非空组件。fit_3 比 fit_2 好，因为前者的 h_2 值较小。

3.2. 演化基准生成器

基于统计检验（U 检验）的层次适应度分配和比较保证了对性能差异的统计稳健性，从而使生成的实例从统计的角度具有意义。HFEBG 可概括如下。

输入包括：1）一个算法池，由 N 个建立的算法，$\mathbf{S_A}$={A_1，A_2，…，A_N}；2）可调基准实例生成器（如[3,9,14,28,31]）；和 3）一个用于演化问题实例的 EA（EA_{PI}）。

每个 A_i（i=1，…，N）依次被视为 A_T。A_T 优于 B_A=S_A-A_T 中的所有算法的问题实例是使用 EA_{PI}、使用层次适应度分配和比较进化而来的。

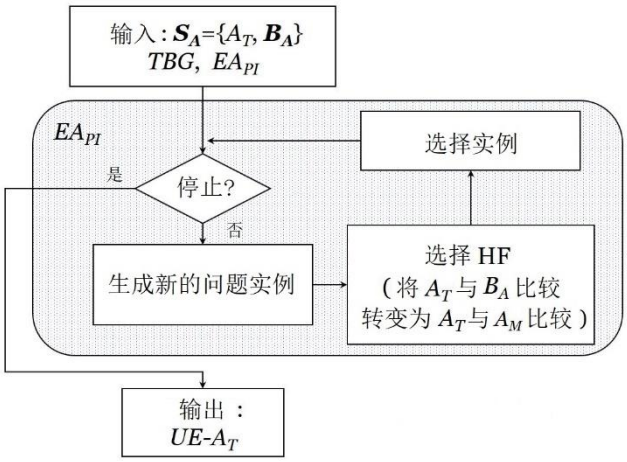


图 2　HFEBG 生成 EA 唯一有利问题的流程图：

HFEBG 的输出是一个基准套件，有 N 个问题实例。对于 EA 来说，每个问题实例都是唯一有利的问题。图 2 显示了为目标 EA 生成唯一简单问题实例的过程。

4. 实验研究

在本节中，我们使用 HFEBG 进行实验。高斯（MSG）生成器的最大集[9]被用作输入可调基准生成器（TBG 如图 2 所示）。稍加修改，使 MSG 实例 *I* 由五个参数定义，即问题维数 d；局部最优数 n；每个局部最优 σ（n×1 向量）的标准差；每个局部最优 **Q**（n×d 矩阵）的每个维度上的压缩率；以及每个局部最优与全局最优 ***r***（n×1 向量）的比率。两个参数设置为固定值：d=d_0 和 n=n_0，因此 *I* = $MSG^{(d_0,n_0)}(\sigma,Q,r)$

我们使用差分进化（DE）[37]来进化问题实例（EA_{PI} 如图 2 所示）。DE 的参数遵循[36]中建议的设置。使用$x^* = DE(x)$，其中$DE(\cdot)$表示 DE 的操作，包括变异、交叉和选择，且x^*是 DE 找到的最佳解。DE 的解表示用$x = \{\sigma, Q, r\}$表示的 MSG 实例。因此，HFEBG 的形式如下。

$$I^* = HFEBG_{DE}(I) = HFEBG_{DE}(MSG^{(d_0,n_0)}(x)) \quad (1)$$

公式（1）显示了 MSG 和 HFEBG 的输入之间的级联关系，即 x 是 MSG 的输入，且 MSG 本身是$HFEBG_{DE}$的输入，其中下标意味着进化工具是 DE。公式（1）的结果实例I^*以唯一简单的实例为例，是一组优化参数$\{\sigma^*, Q^*, r^*\}$，发现在解决问题实例$I^* = MSG^{(d_0,n_0)}(\sigma^*, Q^*, r^*)$时，目标 EA 优于算法池中的任何其他 EA。

为了说明这一思想，我们采用了四种不同进化原理的算法，即人工蜂群（ABC）[18]、复合差分进化（CoDE）[41]、标准粒子群优化 2011（SPSO2011）[47]和 NBIPOP aCMAES[24]。NBIPOP aCMAES 是 2013 年 CEC 竞赛的获胜者[25]。因此，图 2 中所示的算法池被设置为 $\mathbf{S_A}$={ABC，CoDE，SPSO2011，NBIPOP aCMAES}。我们使用这些 EA 的引用中介绍的默认参数（即[18,24,41,47]）。我们设置d_0=10，n_0=10，以及 R=30。实例上 EA 的最大求值次数设置为$d_0 \times n_0 \times 10^3$。生成的实例总数（即 DE 的最大值）设置为 10^3。

4.1. 组成基准套件

在表 2 中，给出了使用上述方法生成的基准测试套件。套件中有四个问题实例。每个问题都是通过选择四个 EA 中的一个作为目标 A_T 而产生的。因此，每个问题对于那个 EA 来说都是唯一容易的。表 2*fit.h₁* 列中的小值表明，对于每个问题实例，A_T 和 A_M 之间的性能差异非常显著（≪0.05）。因此，A_T 与 $\mathbf{B_A}$ 中所有算法的性能差异得以最大化。

表 2 生成的基准套件。对于一个 EA 来说，
每个问题都是唯一容易的。
（前缀 UE 表示唯一容易。）

	fit.h1	**ABC**	**CoDE**	**SPSO 2011**	**NBIPOP-aCMAES**
UE-ABC	2.57×10^{-13}	1	3	2	4
UE-CoDE	8.28×10^{-13}	2	1	4	3
UE-PSO	1.94×10^{-11}	3	2	1	4
UE-CMA	1.23×10^{-7}	2	3	4	1
Average rank		2	2.25	2.75	3

我们使用相同的设置在改进的测试套件上再次测试 EA，结果如表 3 所示。对于每个问题实例，目标 EA 都排在第一位。HFEBG 的目标是生成一个对 EA 来说非常简单的实

例。由于其产生的问题实例对于 EA 在另一时间对其进行测试是唯一容易的，因此它证实了由于统计显著性，实验结果可以如预期的那样重复。

表 3 显示了问题维度为 10 时的测试结果。HFEBG 不受维数的限制。表 4 显示了当问题维数分别设置d_0=20（局部最优数n_0=3）和d_0=30（n_0=2）时的测试结果。注意，局部最优n_0的数量可以任意设置，这里我们使用小值来简化。对于每个 EA，在 20 维和 30 维问题空间中生成一个 UE 实例。由于在生成每个 UE 实例时保证了统计保证，所以测试结果是一致的，即目标 EA 在优化其 UE 实例方面优于其他 EA。结果表明，当问题维数发生变化时，该方法可以得到成功的应用。

表 3 四个 EA 再次在基准套件上测试

	ABC	**CoDE**	**SPSO 2011**	**NBIPOP-aCMAES**
UE-ABC	1	3.5	2	3.5
UE-CoDE	2	1	4	3
UE-PSO	3	2	1	4
UE-CMA	2	4	3	1
Average rank	2	2.625	2.5	2.88

表 4 研究 HFEBG 的标度性质：
问题维数设为 d_0=20 和 d_0=30

	d_0=20				d_0=30			
	ABC	**CoDE**	**SPSO 2011**	**NBIPOP-aCMAES**	**ABC**	**CoDE**	**SPSO 2011**	**NBIPOP-aCMAES**
UE-ABC	1	3.5	2	3.5	1	2	3	4
UE-CoDE	2	1	4	3	3.5	1	2	3.5
UE-PSO	3	2	1	4	3	2	1	4
UE-CMA	2	4	3	1	2	3	4	1
Average rank	2	2.625	2.5	2.88	2.375	2	2.5	3.125

4.2. CEC2013 测试套件

有人批评说，提出许多新颖的电子艺界几乎没有贡献[10]，因为它们没有与竞赛的获胜者（如[1,6,7,11,24,39]）进行比较。然而，似乎没有人注意到，如果测试套件稍加修改，比较结果可能会有很大的不同，如表 5 所示。

表 5 将一个产生的问题添加到 CEC 2013 基准。

（前缀 UD 表示唯一的困难。）

	ABC	**CoDE**	**SPSO 2011**	**NBIPOP-aCMAES**
Average rank on CEC 2013	2.77	1.89	3.54	***1.80***
UD-CMA	2	1	3	4
Average rank overall	2.74	***1.86***	3.52	1.88

在表 5 中，第一行显示了 CEC 2013 基准[22]上四个算法的平均排名，共有 28 个问题（10 维，评估数为 1×10^5），第二行给出了一个生成的实例 UD-CMA（NBIPOP ACMAE 的唯一困难），对于该实例，唯一困难的实例是相反的生成唯一简单实例的方法。最后一行给出了所有 29 个问题的平均排名。可见，加入 UD-CMA 后，CoDE 成为赢家。

表 6 四个基于单次运行的搜索的简单实例（前缀 E 表示简单）

	E-ABC	**E-CoD**	**E-PSO**	**E-CMA**
result	0	0	0	0
effort	7660	28020	4964	2491

4.3. 单次运行搜索工作

与使用基于 U 检验的适应度分配（以统计方式考虑 EA 性能）不同，基于单次搜索努力（SSE）的适应度分配[13]测量单次运行性能。在下面，基于 SSE 的适应度分配用于生成对每个算法都很容易的实例。结果实例如表 6 所示。在这种情况下，问题实例的难度比较如下：如果找到全局最小值，则所需的搜索工作量越少，实例就越容易；如果没有找到全局最小值，则使用允许的最大搜索工作量，则获得的结果越小，实例就越容易。与之前一样，使用 DE，并为每个 EA 返回 10^3 评估中的最佳结果。

这些问题在四个 EAs 上进行了测试。从表 7 可以看出，由于缺乏统计检验，结果是不一致的，例如，当 E-ABC（easy for ABC）进化时，它的单次搜索工作是 7660 个 ABC 评价，以找到全局最优，而当它再次被测试时，平均花费 ABC 高达 50882.2 个评价（超过 30 个独立运行）。此外，由此产生的问题实例并不具有代表性。从表中可以看出，这四个实例对于 NBIPOP aCMAES 都是最简单的。显然，对于一个算法来说（假设）容易的实例对于另一个算法来说也可能容易。

表 7 四个 EA 在四个基于单次运行搜索进化而来的简单实例上进行了测试。（前缀 E 表示简单。）

		ABC	CoDE	SPSO 2011	NBIPOP-aCMAES
E-ABC	**result**	0	0	0	0
	effort	50882.2	31038	22265.9	7600
E-CoDE	**result**	0	0	1.30E-5	0
	effort	50582.8	29640	100000	7504
E-PSO	**result**	0	0	0	0
	effort	52789.2	29478	17963	7435
E-CMA	**result**	0	0	0	0
	effort	60031.7	30201	15275.7	7430

5. 讨论

当所采用的算法（N）数目较大时，对于每一个算法，生成了唯一容易问题 M_E 和唯一困难问题 M_H，然后所得到的问题由一个均匀分布的基准集组成，其中均匀分布是由所采用的所有算法的等效性能显式保证的

HFEBG 的小规模测试套件（如表 2 所示，其中 N=4，M_E=1 和 M_H=0）提供了访问新算法 A_{nv} 性能的另一种方法：

1） 如果 A_{nv} 在解决其唯一的简单问题上不劣于或优于已建立的 EA，则 A_{nv} 被认为与已建立的 EA 一样好。例如，如果 A_{nv} 在解决 UE-ABC 问题上没有明显优于或低于 ABC，则 A_{nv} 被认为与 ABC 一样好。请注意，CoDE、SPSO2011 和 NBIPOP-aCMAES 在解决 UE-ABC 方面不如 ABC。

2） 如果 A_{nv} 不劣于或优于两个（或更多）已建立的 EAs 解决其唯一容易的问题，例如，如果 A_{nv} 不劣于 ABC 解决 UE-ABC，也不劣于 NBIPOP-aCMAES 解决 UE-CMA，则有理由推断 A_{nv} 优于所使用的算法，因为它将算法的适用性从一个唯一的简单问题扩展到两个问题。

3） 当 A_{nv} 在其唯一容易的问题上优于已建立的 EA 时，由于其相似的性能，有可能观察到相似的算法特性。例如，如果 A_{nv} 在求解 UE-ABC 方面优于 ABC，我们可以首先分析这两种算法是否具有一些共同的特性，而这些特性在 CoDE、SPSO2011 或 NBIPOP-aCMAES 中是找不到的。

4） 如果 A_{nv} 不能很好地处理任何已建立的 EA 的有利问题，那么我们进一步考察 A_{nv} 是否是一种与已建立的算法有不同特点的新算法。此外，它独特的简单问题可以由 HFEBG 生成，可以添加 HFEBG 来丰富测试套件。

此外，由此产生的问题也可用于 EA 选择框架[26,29,46]和 EA 组合[40,44,45]。唯一简单和唯一困难的问题提供了极端的训练案例，一个算法将执行最好和最坏的。文[26]给出了一个例子，其中算法选择器的知识库被有意生成的唯一简单问题所丰富。

6. 结论

本文提出了一种新的方法来系统地构建进化算法的替代基准测试集。使用了一组已建立的算法。由于已建立的算法在应用中得到了广泛的应用，因此选择了它们。因此，评估它们的性能更有意义，并提供更多有用的信息。对于每一个算法，都会演化出一个唯一简单的问题，并且所有这些问题实例组成一个测试套件。每个算法只赢一次，而且套件中的每个实例对于一个且只有一个算法是唯一容易的。在统计上，获胜算法与其他算法的性能差异最大。基于统计检验（U 检验）的层次适应度分配显著降低了算法性能的随机性。这种新的测试集生成方法使得新提出的算法和已建立的算法之间的比较易于执行。与在固定基准测试套件中竞争第一名相比，它提供了更多的洞察力，并且比基于单次运行搜索工作的性能比较更可靠。它还为 EA 选择框架和 EA 投资组合提供有意义的培训问题实例。拟议工作的源代码见[49]。

致谢：致谢已略(见原文)

参考文献：参考文献已略(见原文)