arXiv:2601.10263v1 [cs.NE] 15 Jan 2026

一种结合交叉搜索与麻雀搜索以处理劣质个体的进化算法集成

Mingxuan Du[1], Tingzhang Luo[1], Ziyang Wang[2], 和 Chengjun Li*[1]

[1] 中国地质大学（武汉）[2] 吉林大学

摘要。 在人工智能领域，实参数单目标优化是一项重要研究方向。差分进化（DE）和协方差矩阵自适应进化策略（CMA-ES）在实参数单目标优化方面均表现出良好的性能。不过，也存在其他类型的进化算法可用于此目的。近年来，研究人员开始关注长期搜索。EA4eig——一种由三种差分进化变体和 CMA-ES 组成的集成算法——在长期搜索方面表现出色。本文将近期提出的两种进化算法——交叉搜索和麻雀搜索——作为次级进化算法引入 EA4eig，用于处理劣质个体，由此得到 EA4eigCS。在我们的集成算法中，次级进化算法旨在改变种群分布，以打破停滞。实验结果表明，EA4eigCS 优于 EA4eig，与最先进算法相比也具有竞争力。代码和补充材料见：https://anonymous.4open.science/r/EA4eigCS-2A43。

关键词： 集成 · 次级进化算法 · 交叉搜索 · 麻雀搜索。

1 引言

在解空间中搜索最佳决策向量，以使目标函数最小化（或最大化）的实参数单目标优化，是人工智能领域的重要研究方向。近年来，研究人员开始研究长期搜索，其函数评估次数的最大值（$MaxFES$）作为算法的终止准则，会随着解空间维数（D）的增加呈指数级增长。毕竟，实参数单目标优化的难度会随着 D 的增加呈指数级增长。

迄今为止，各类进化算法已用于实参数单目标优化。其中，差分进化

* 通讯作者。

（DE） [16] 和协方差矩阵自适应进化策略（CMA-ES） [7,6] 均表现出良好的性能。

在差分进化的每一代中，变异会基于目标向量生成变异向量（即个体）。随后，基于变异向量和目标向量对，通过交叉生成试验向量。在选择阶段，对于每一对试验向量和目标向量，选择适应度更高的向量，并使其成为下一代的目标向量。

在CMA-ES的每一代中，适应度更高的个体决定个体的中心。随后，通过从高斯分布中进行采样，基于该中心生成后代。在所有亲代和后代中，选择适应度更高的个体进入下一代。

目前已有许多差分进化与CMA-ES的集成算法。例如，对于长期搜索，EA4eig [4] 是由CMA-ES和三个差分进化变体组成的集成算法——CoBiDE [17], 、IDEbd [5], 和jSO [1]。在该集成算法中，每一代都会根据轮盘赌选择四个组成算法中的一个。选择第 h个算法—— p_h ——的概率计算如下:

$$p_h = \frac{n_h + n_0}{\sum_{j=1}^{H}(n_j + n_0)}(h = \{1, 2, 3, 4\}), \tag{1}$$

其中， n_h 表示当前代中第 h个算法的成功次数， n_0 是一个常数。在该集成中，协方差矩阵学习的应用范围从 CoBiDE 扩展到了所有组成算法。EA4eig采用线性种群规模缩减（LPSR）。该算法是IEEE进化计算大会（CEC）2022年实参数单目标优化竞赛的优胜者；这是最新一届面向长期搜索的CEC竞赛。

与大多数实参数单目标优化集成算法相比，EA4eig包含更多组成算法。例如，同样为长期搜索设计的APGSK-IMODE [10] 和APGSK-IMODE-FL [20] 都只有两个组成算法。事实上，EA4eig的四种组成算法早在多年前便已提出，而且完全不是为长期搜索设计的。在EA4eig框架下，每种组成算法的搜索都可能在一代之后被中断。因此，其收敛速度较慢，难以适应长期搜索。

除差分进化和 CMA-ES 外，还有其他类型的实参数单目标优化进化算法。其中，交叉搜索 [9] 和麻雀搜索 [18] 是最近提出的。尽管这两类算法仍处于研究初期，但确实具有一些有趣的特点。

交叉搜索包含水平交叉和垂直交叉。前者在成对个体之间执行，涉及个体的所有维度。而后者则在所有个体之间进行，只考虑两个维度。

在麻雀搜索中，种群中表现最佳的个体是生产者，而其他个体则是乞讨者。这两类个体的行为各不相同。此外，当察觉到危险时，群体边缘的麻雀会迅速移向安全

区域，以获得更好的位置；而群体中间的麻雀则会随机行走，以便靠近其他麻雀。

本文的研究动机如下。鉴于 EA4eig 在实参数单目标优化的长期搜索中表现良好，进一步改进该集成算法是有意义的。由于边际效应，向 EA4eig 中加入更多竞争力相当的组成算法并不是最佳的改进方式。相反，值得尝试的是，在轮盘赌之后，向框架中加入一个或多个组成算法，用于处理特定个体。

本文将交叉搜索和麻雀搜索集成到 EA4eig 中，这是因为它们的行为与 EA4eig 中的组成算法非常不同。因此，我们提出了同时采用交叉搜索和麻雀搜索的 EA4eig——EA4eigCS。在 EA4eigCS 中，如果最佳适应度经过若干代仍未更新，则使用交叉搜索处理最差个体。此外，在所有代结束时，应用一步麻雀搜索来更新最差个体。在我们提出的集成算法中，CoBiDE、IDEbd、jSO 和 CMA-ES 被称为主要组成算法。同时，交叉搜索和麻雀搜索均为次要组成算法，因为在一代中，最多只有一部分个体由这两种方法控制。我们的 EA4eigCS 可能是用于实参数单目标优化、包含组成进化算法数量最多的集成算法。

我们基于 CEC 2020、2021年和 2022年基准测试套件开展实验。在实验中，我们将 EA4eigCS 与最先进或最新的算法进行比较。关于对比算法的详细信息，请参见第2节。结果表明，我们的算法具有竞争力。

总之，我们的主要贡献如下：

– **算法方面：** 我们基于 EA4eig 框架提出了 EA4eigCS，这是一种改进的进化算法集成，包含交叉搜索和麻雀搜索。– **方法方面：** 我们采用次级进化算法而非局部搜索来处理劣质个体，而不是处理最优个体。– **实验方面：** 在 CEC基准测试上进行的大量实验表明，EA4eigCS 的性能优于最先进算法。

2 相关工作

本节首先详细介绍交叉搜索和麻雀搜索，因为我们的 EA4eigCS 基于这两种技术。然后，我们回顾现有的长期搜索算法。

2.1 交叉搜索

如上所述，交叉搜索包括两种交叉方式。在每一代中，首先执行水平交叉。第 i 个个体在第 g 个

代$\boldsymbol{x}_{i,\ g}$与同一代中的第 i' 个个体 $\boldsymbol{x}_{i',\ g}$ 产生后代$\boldsymbol{MS_hc}_{i,\ g}$和$\boldsymbol{MS_hc}_{i',\ g}$。这里,

$$MS_hc_{j,i,g} = r_1 \cdot x_{j,i,g} + (1 - r_1) \cdot x_{j,i',g} + c_1 \cdot (x_{j,i,g} - x_{j,i',g}), \tag{2}$$

而

$$MS_hc_{j,i',g} = r_2 \cdot x_{j,i',g} + (1 - r_2) \cdot x_{j,i,g} + c_2 \cdot (x_{j,i',g} - x_{j,i,g}). \tag{3}$$

产生的 $\boldsymbol{MS_hc}_{i,g}$和 $\boldsymbol{MS_hc}_{i',g}$分别与 $\boldsymbol{x}_{i,g}$和 $\boldsymbol{x}_{i',g}$竞争。进行水平交叉后，再进行垂直交叉。所有个体在第 g 代的第 $j1$ 维将基于与第 $j2$ 维的交叉进行变异，如下所示,

$$MS_vc_{j1,i,g} = r \cdot x_{j1,i,g} + (1 - r) \cdot x_{j2,i,g}, \tag{4}$$

其中 $i = 1, 2, ..., NP$。类似地，产生的 $\boldsymbol{MS_vc}_{i,g}$（$i = 1, 2, ..., NP$）与 $\boldsymbol{x}_{i,g}$竞争。在公式 2-4 中， $r1$、 $r2$、 r 是取值在 0 和 1 之间的均匀分布随机值参数，而 $c1$ 和 $c2$ 是取值在 -1和 1 之间的均匀分布随机值参数。

2.2 麻雀搜索

在麻雀搜索中，所有个体按适应度降序排列。种群中最优的个体是生产者，其余个体则是乞讨者。生产者的位置按如下方式更新,

$$x_{j,i,g+1} = \begin{cases} x_{j,i,g} \cdot exp(\frac{-i}{\alpha \cdot MaxFES}), if\ R_2 < ST, \\ x_{j,i,g} + Q \cdot L, \qquad\qquad otherwise, \end{cases} \tag{5}$$

其中 $j = 1$、2、.....、D。 $R_2 \in [0, 1]$和 $ST \in [0.5, 1.0]$）分别表示警戒值和安全阈值。 Q 是服从正态分布的随机数。 L 是一个所有元素值均为 1 的 $1 \times D$ 矩阵。对于乞讨者，位置按如下方式更新,

$$x_{j,i,g+1} = \begin{cases} Q \cdot exp(\frac{x_{j,worst,g} - x_{j,i,g}}{i^2}), & if\ i > NP/2, \\ x_{j,best,g} + |x_{j,i,g} - x_{j,best,g}| \cdot A^+ \cdot L, & otherwise, \end{cases} \tag{6}$$

其中 $A^+ = A^T（AA^T）^{-1}$， A是一个所有元素值为 1 或-1 的 $1 \times D$矩阵。然后，再次对所有个体进行排序。前一半个体是意识到危险的个体。它们的位置按如下方式更新,

$$x_{j,i,g+1} = \begin{cases} x_{j,best,g} + \beta \cdot |x_{j,i,g} - x_{j,best,g}|, if\ f(x_{j,i,g})) > f(x_{j,best,g}), \\ x_{j,i,g} + K \cdot \frac{|x_{j,i,g} - x_{j,worst,g}|}{f_i - f_w + \varepsilon}, \qquad if\ f(x_{j,i,g})) = f(x_{j,best,g}), \end{cases} \tag{7}$$

其中， β 是均值为0、方差为1的随机数正态分布， $K \in [-1, 1]$ 是一个随机数，f(·)表示适应度， ε 是用于避免零除错误的最小常数。

2.3 长期搜索的现有算法

长期搜索的研究始于2020年。最初，长期搜索算法主要来源于2020年、2021年和2022年分别举办的最近三届实参数单目标优化CEC竞赛。近年来，也可以在其他地方找到适用于长期搜索的算法。

IMODE [12], 是 CEC 2020竞赛的获胜算法，是一种采用三种变异策略的差分进化变体。这三种变异策略会在每一代中随机选取个体。APGSK-IMODE [10], 是 CEC 2021竞赛中非移位问题的获胜算法，由一种自适应的获取-共享知识（GSK） [11] 算法与 IMODE 集成而成。两个组成算法会定期共享最优个体。NL-SHADE-RSP [14],在同一竞赛中同时赢得移位问题和旋转移位问题，采用选择压力调整，并自动调整档案用于收集被淘汰个体的使用概率。jDE-21 [3], 是非旋转移位问题的获胜算法，采用重启机制以增加多样性。同时，经过修订后，算法采用了 j2020 [2] 提出的拥挤机制和变异个体选择方案。MLS-L-SHADE [8] 包含多个由同一差分进化算法控制的子种群。初始时，分别随机选取指定数量的个体作为各子种群的初始中心。随后，根据聚类确定子种群的数量。如果某个子种群在指定代数内没有取得进展，则该子种群会重新启动。EA4eig [4]已在第1节中介绍。NL-SHADE-LBC [15], 是 CEC 2022竞赛中排名第二的算法，其变异和交叉策略是在 L-SHADE-RSP [13] 和 NL-SHADE-RSP 的基础上修改而来。

在 [20], 中，APGSK-IMODE-FL 是基于 APGSK-IMODE 提出的。在所提出的集成中，已在种群中停留足够多代且适应度较差的个体会在两种组成算法之间交换。在 [19], 中，AMCDE 是基于 IMODE 提出的。该算法有两种状态。在垄断状态下，所有个体都由一种变异策略控制。在竞争状态下，三种变异策略相互竞争，争夺控制权。

在上述算法中，IMODE、APGSK-IMODE、EA4eig、APGSK-IMODE-FL 和 AMCDE 都是集成方法。IMODE 和 AMCDE 都是采用变异策略集成的差分进化算法。APGSK-IMODE 和 APGSK-IMODE-FL 都是差分进化与 GSK 的集成。尽管 EA4eig 也属于差分进化与另一种进化算法的集成，但与 APGSK-IMODE 和 APGSK-IMODE-FL 相比，它集成了数量更多的组成算法。

3 方法论

鉴于 EA4eig 已取得良好性能，我们计划进一步改进该算法。对于连续优化问题，例如实参数单目标优化，停滞是表明求解过程无法再取得进展的主要现象

算法 1 EA4eigC 的伪代码 S

输入:
NP_{max}，种群规模的初始值;
NP_{min}，种群规模的最终值;
$MaxFES$;
T_{gen}，最佳适应度未发生变化的代数阈值,
R_c，经交叉搜索处理的个体比例;
R_s，经麻雀搜索考察的个体比例;
参数:
NP, FES, 以及 t
初始化种群的初始代 P_0
评估 NP_{max} 个体在 P_0 中
$NP = NP_{max}$, $FES = NP$, 以及 $t = 0$
当 $FES <= MaxFES$ **时，执行**
    根据公式 1 得出的概率，从四种算法中选择一种以控制种群
    $FES = FES + NP$
    如果最佳适应度未更新**则**
        t=t+1
    else
        t=0
    结束如果
    如果 $t = T_{gen}$ **then**
        按适应度降序排列个体
        对最差的 $NP \cdot R_c$ 个体进行交叉搜索
        $FES = FES + NP \cdot R_c$
        $t = 0$
    结束 if
    对 $NP \cdot R_S$ 最差的个体执行麻雀搜索
    $FES = FES + NP$
    $NP = \frac{FES}{MaxFES} \cdot (NP_{min} - NP_{max}) + NP_{max}$
结束 while
报告解

。尽管 EA4eig 性能良好，集成算法仍常常陷入停滞，因此需要加以改进。然而，考虑到边际效应，在轮盘赌中加入原有四种算法之外的更多组成算法，可能并非高效之选。相反，应针对种群中的特定个体采用相应方法来增强算法。若通过所采用的方法改变一个或多个个体，种群分布就会发生变化，从而有望避免停滞。

在许多用于实参数单目标优化的算法中，局部搜索与主要算子一起作用于特定个体，以改变种群的分布。在本文中，我们倾向于让 EA4eig 中不同于四种主要算法的进化算法作用于特定个体，以进一步改变种群的分布。本文将加入集成中的此类进化算法称为次级组成算法。尽管次级组成算法在一代中可能只控制少数个体，但种群中的所有个体都可以参与其变异和交叉。因此，与局部搜索相比，次级组成算法在生成后代时考虑了更多的遗传信息。

本文选取交叉搜索和麻雀搜索作为次级组成算法。在我们的集成中，这两种次级组成

算法仅作用于劣质个体。原因如下：只有当次级组成算法产生的后代适应度优于其亲代并取代后者时，这两种方法才能通过改变种群分布对进化产生影响。这两种方法可能难以改进当前的最佳个体。毕竟，与差分进化变体和 CMA-ES 相比，交叉搜索和麻雀搜索仍有待发展。不过，改进劣质个体的可能性远高于刷新最佳个体的可能性。更新最差个体仍会导致种群分布发生变化。在随后执行主要组成算法时，最佳适应度可能会基于变化后的种群再次得到提升。下文将详细介绍如何集成交叉搜索和麻雀搜索。

首先，若经过给定代数 T_{gen}后，种群中的最佳适应度仍未更新，则在执行某个主要组成算法后，对该代中最差的 $R_c \cdot NP$个个体调用交叉搜索，其中 R_c 是由交叉搜索处理的个体所占比例， NP 表示种群规模。该方案可解释如下：种群中的最佳适应度在给定代数内未更新，表明对于主要组成算法而言，提升当前最佳适应度十分困难。因此，我们调用交叉搜索来提供帮助。

此外，在每代结束时，对一部分个体应用公式 7 中的麻雀搜索步骤。具体而言，从最差的 $R_s \cdot NP$个个体中随机选取 $R_s^2 \cdot NP$个执行该步骤，其中 R_s 是麻雀搜索所考虑个体的比例。也就是说，在每一代中，都会以一定比例对最差个体中的一部分应用该步骤，以改进这些个体。

基于这两种搜索技术的整合，我们实现了 EA4eigCS。算法 1 给出了 EA4eigCS 的伪代码。从算法 1 可以看出，与 EA4eig 相比，我们的 EA4eigCS 额外包含三个参数—— T_{gen}、 R_c 和 R_s。

4 实验研究

4.1 实验设置

基准测试。我们在两个基准测试套件——CEC 2021 和 CEC 2022 基准测试套件上验证了算法 EA4eigCS 的性能。在本研究中，我们首先在 CEC 2022 基准测试上对最先进算法进行了比较分析，以确定表现最具竞争力的算法。随后，我们进一步在 CEC 2021 基准测试上评估这些表现最佳的算法，以验证其有效性。此外，CEC 2022 数据集还用于开展消融研究，从而剖析所提出方法中各个组成部分的贡献。

对比方法。除我们的 EA4eigCS 外，还将七种算法纳入比较：IMODE [12], NL-SHADE-RSP [14], APGSK-IMODE [10], MLS-L-SHADE [8], EA4eig [4],

实验中还将 NL-SHADE-LBC [15], 和 AMCDE [19] 作为对比算法。所有对比算法均已在第2节中介绍。

实现细节。所有算法的实验参数设置见表1。为公平比较，所有算法均在 MATLAB R2021b 上运行。我们采用 Wilcoxon秩和检验和 Friedman 检验作为评估指标。

表1：所涉及算法的设置 ms

算法	参数
IMODE	$NP_{max} = D^2 \cdot 6$、 $NP_{min} = 4$、 $A_{rate} = 2.60$、 $H = D \cdot 20$、 $FES_{LS} = MaxFES \cdot 0.85$和 $p = 0.3$ [12]
NL-SHADE-RSP	$NP_{max} = 30D$、 $M_{f,\ r} = 0.2$、 $M_{CR,\ r} = 0.2$和 $n_A = 0.5$ [14]
APGSK-IMODE	$NP_2^{max} = \frac{D \cdot 30}{4}$, $NP_2^{min} = 12$, $NP_1^{max} = D \cdot 30 - NP_2^{max}$、 $NP_1^{min} = 4$和 $CS = 50$ [10]
MLS-L-SHADE	$NP_{max} = D \cdot 18$、 $NP_{min} = 4$、$A_{rate} = 2.60$和 $H = D^2 \cdot 0.36$
EA4eig	$NP_{max} = 100$ 和 $NP_{min} = 10$ [4]
NL-SHADE-LBC	$NP_{max} = D \cdot 20$, $M_{F,\ r} = 0.9$, $M_{Cr,\ r} = 0.9$, $k = 1$, $\|A\| = NP \cdot 0.1$, 和 $n_A = 0.5$ [15]
AMCDE	$NP_{max} = D^2 \cdot 6$, $NP_{min} = 4$, $A_{rate} = 2.60$, $H = D \cdot 20$, $FES_{LS} = MaxFES \cdot 0.85$, $p = 0.3$, $G_{n\ _init} = 5$, $p_{bc1} = 0.4$, $p_{bc2} = 0.4$, $p_r = 0.01$, 以及 $p_w = 0.2$ [19]
EA4eigCS	除了 EA4eig 的参数设置外, $T_{gen} = 3$, $R_c = 1/6$, $R_s = 1/2$

4.2 实验结果

首先，我们基于 CEC 2022基准测试套件（最新的 CEC 长期搜索套件），将 EA4eigCS 与所有对比方法进行比较。结果列于补充材料中的表 S1 和表 S2，并汇总于表 2。如表 2 所示，Wilcoxon秩和检验和 Friedman检验均表明 EA4eigCS 性能优越。根据 Wilcoxon秩和检验，在 $D = 10$ 时，EA4eigCS 与 EA4eig 表现相当；在其他所有情况下，EA4eigCS 的表现都更好。根据 Friedman检验，我们的集成表现最佳。

我们在图 1 中给出了七个函数上六种算法的收敛图$D = 20$ 。选择这七个函数，是因为所有算法都未能达到全局最优解。对于每个函数，图中展示了经过一定次数的函数评估（具体为 11 次评估）后，30 次独立运行所得的平均适应度值。从图中可以看出，MLS-L-SHADE 的收敛曲线在运行过程中可能变差。这一现象归因于子种群重启策略。图中表明，对于大多数函数，例如 F4、F6、F8、F9、F11 和 F12，EA4eigCS 与 EA4eig 的曲线非常相似。简而言之，这两种搜索方法的集成对收敛方式影响甚微。

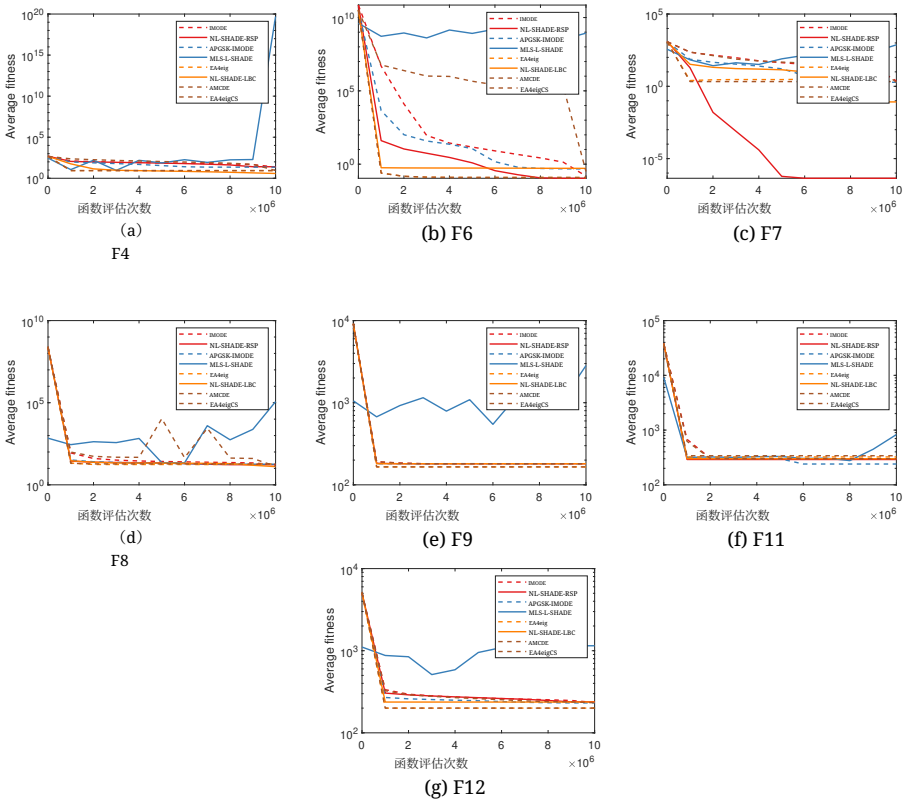
表 2：基于 CEC 2022 基准测试套件结果的 Wilcoxon秩和检验和 Friedman检验结果。“+”和“–”分别表示当前算法显著优于和统计上劣于 EA4eigCS。同时，“≈”表示差异不显著

算法	检验类型	$D = 10$	$D = 20$
IMODE [12][CEC'20]	Wilcoxon +/-/≈	2/4/6	1/6/5
	弗里德曼排名	4.63 (6)	5.08 (6)
NL-SHADE-RSP [14][CEC'21]	Wilcoxon +/-/≈	2/4/6	3/5/4
	弗里德曼排名	4.50 (5)	5.08 (6)
APGSK-IMODE [10][CEC'21]	Wilcoxon +/-/≈	2/4/6	1/9/2
	弗里德曼排名	4.21 (3)	5.46 (8)
MLS-L-SHADE [8][CEC'20]	Wilcoxon +/-/≈	3/5/2	3/4/5
	弗里德曼排名	5.29 (8)	3.46 (2)
EA4eig [4][CEC'22]	Wilcoxon +/-/≈	0/0/12	0/2/10
	弗里德曼排名	4.21 (3)	4.83 (5)
NL-SHADE-LBC [15][CEC'22]	Wilcoxon +/-/≈	0/5/5	3/4/5
	弗里德曼排名	5.21 (7)	4.71 (4)
AMCDE [19][SEC'23]	Wilcoxon +/-/≈	2/4/6	1/6/5
	弗里德曼排名	4.17 (2)	4.21 (3)
EA4eigCS（我们的方法）	Wilcoxon +/-/≈	-/-/-	-/-/-
	弗里德曼排名	**3.79 (1)**	**3.17 (1)**

表 3：基于 CEC 2021基准测试套件结果的 Wilcoxon秩和检验与 Friedman检验结果

算法	检验类型	2021年基准测试	
		$D = 10$	$D = 20$
NL-SHADE-RSP [14][CEC'21]	Wilcoxon +/-/≈	4/1/5	5/0/5
	弗里德曼排名	**2.15 (1)**	**1.95 (1)**
EA4eig [4][CEC'22]	Wilcoxon +/-/≈	0/0/10	0/0/10
	弗里德曼排名	3.70 (5)	3.25 (3)
NL-SHADE-LBC [15][CEC'22]	Wilcoxon +/-/≈	0/4/6	1/5/4
	弗里德曼排名	3.65 (3)	3.65 (5)
AMCDE [19][SEC'23]	Wilcoxon +/-/≈	1/4/5	2/4/4
	弗里德曼排名	3.65 (3)	3.40 (4)
EA4eigCS（我们的方法）	Wilcoxon +/-/≈	-/-/-	-/-/-
	弗里德曼排名	3.05 (2)	2.75 (2)

随后，为了进一步验证 EA4eigCS 的优越性，我们基于 CEC 2021 基准测试，将我们的集成算法与先前比较中表现最佳的同类算法进行比较。结果列于补充材料的表 S3 和表 S4，并汇总于表 3。根据表 3，在 $D = 10$和 $D = 20$ 两方面，就 Wilcoxon秩和检验而言，EA4eigCS 与 EA4eig 相当，逊于 NL-SHADE-RSP，但优于其他同类算法。就 Wilcoxon秩和检验而言，我们的集成算法不敌 NL-SHADE-RSP，但胜过其他算法。



(a) F4 (b) F6 (c) F7

(d) F8 (e) F9 (f) F11

(g) F12

图1： $D = 20$时，CEC 2022基准测试套件中七个函数上八种算法的收敛图

4.3 消融研究

我们进行了消融测试，以评估所提出方案的有效性。我们将麻雀搜索简称为 SSA，将交叉搜索简称为 CSO。“- 劣质 -”一词用于表示 CSO 和 SSA 均仅应用于劣质个体。四个算法版本在 CEC 2022基准测试套件上、维度为 D=20 时的结果见表S5，并在表4中通过 Wilcoxon秩和检验和 Friedman检验的结果进行了汇总。可以看出，CSO 和 SSA 均能带来改进。此外，当两种方法集成后，解的性能得到进一步提升。值得注意的是，当 CSO 和 SSA 有选择地应用于预先定义的劣质个体，而非所有个体时，算法的性能会变得更强。简而言之，所提出的每个步骤都至关重要。

表4：消融研究总结

SSA	CSO	劣解	Wilcoxon +/-/≈	弗里德曼排名
			0/2/10	4.04 (5)
✓			0/1/11	2.54 (3)
	✓		0/2/10	3.54 (4)
✓	✓		0/0/12	2.46 (2)
✓	✓	✓	-/-/-	2.42 (1)

5 结论

尽管实参数单目标优化已经研究了数十年，但近年来人们才开始关注长期搜索。已有许多算法被提出用于长期搜索。例如，EA4eig 是由三种差分进化变体和 CMA-ES 组成的集成算法，性能非常出色。

本文将交叉搜索和麻雀搜索集成到 EA4eig 中，作为次级进化算法来处理劣质个体，从而得到 EA4eigCS。实验结果表明，在实参数单目标优化的长期搜索方面，与最先进算法相比，我们的 EA4eigCS 具有竞争力。尽管 EA4eig 和 EA4eigCS 的收敛方式相似，但我们提出的集成性能优于前者。

未来，我们可以进一步研究由大量组成进化算法构成的集成，用于实参数单目标优化。我们还可以考虑更多类型的进化算法，将其作为次级进化算法，以构建性能更好的集成。

参考文献

1. Brest, J., Maučec, M.S., Bošković, B.: 单目标实参数优化：jSO算法。载于：2017年IEEE进化计算大会（CEC）。第1311–1318页。IEEE（2017） 2. Brest, J., Maučec, M.S., Bošković, B.: 用于单目标边界约束优化的差分进化算法：j2020算法。载于：2020年IEEE进化计算大会（CEC）。第1–8页。IEEE（2020年） 3. Brest, J., Maučec, M.S., Bošković, B.: 用于单目标边界约束优化的种群规模缩减自适应差分进化算法：j21算法。载于：2021年IEEE进化计算大会（CEC）。第817–824页。IEEE（2021年） 4. Bujok, P., Kolenovsky, P.: 特征向量交叉在应用于CEC 2022单目标数值优化的进化算法协同模型中的应用。载于：2022年IEEE进化计算大会（CEC）。第1–8页。IEEE（2022年） 5. Bujok, P., Tvrdík, J.: 具有种群规模自适应的增强型个体依赖差分进化。载于：2017年IEEE进化计算大会（CEC）。第1358–1365页。IEEE（2017）

6. Hansen, N., Müller, S.D., Koumoutsakos, P.: 降低带协方差矩阵自适应（CMA-ES）的去随机化进化策略的时间复杂度。《进化计算》**11**(1), 1–18 (2003) 7. Hansen, N., Ostermeier, A.: 在进化策略中自适应任意正态变异分布：协方差矩阵自适应。载于：1996年IEEE进化计算大会（CEC）。第312–317页。IEEE（1996） 8. Jou, Y.C., Wang, S.Y., Yeh, J.F., Chiang, T.C.: 用于单目标有界约束优化的多种群改进型L-SHADE。载于：2020年IEEE进化计算大会（CEC）。第1–8页。IEEE（2020年） 9. Meng, A.b., Chen, Y.c., Yin, H., Chen, S.z.: 交叉优化算法及其应用。《知识型系统》**67**, 218–229 (2014) 10. Mohamed, A.W., Hadi, A.A., Agrawal, P., Sallam, K.M., Mohamed, A.K.: 用于求解CEC 2021基准问题的获取-共享知识算法与自适应参数及iMODE算法的混合。载于：2021年IEEE进化计算大会（CEC）。第841–848页。IEEE（2021年） 11. Mohamed, A.W., Hadi, A.A., Mohamed, A.K.: 用于求解优化问题的获取-共享知识算法：一种新型自然启发算法。《国际机器学习与控制论杂志》**11**(7), 1501–1529 (2020年) 12. Sallam, K.M., Elsayed, S.M., Chakrabortty, R.K., Ryan, M.J.: 用于求解无约束问题的改进型多算子差分进化算法。载于：2020年IEEE进化计算大会（CEC）。第1–8页。IEEE（2020年） 13. Stanovov, V., Akhmedova, S., Semenkin, E.: 采用基于排名的选择压力策略、用于求解CEC 2017基准问题的LSHADE算法。载于：2018年IEEE进化计算大会（CEC）。第1–8页。IEEE（2018） 14. Stanovov, V., Akhmedova, S., Semenkin, E.: 采用自适应档案和选择压力、用于CEC 2021数值优化的NL-SHADE-RSP算法。载于：2021年IEEE进化计算大会（CEC）。第809–816页。IEEE（2021年） 15. Stanovov, V., Akhmedova, S., Semenkin, E.: 采用线性参数自适应偏差变化、用于CEC 2022数值优化的NL-SHADE-LBC算法。载于：2022年IEEE进化计算大会（CEC）。第01–08页。IEEE（2022年） 16. Storn, R., Price, K.: 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。《全球优化杂志》**11**(4), 341–359 (1997) 17. Wang, Y., Li, H.X., Huang, T., Li, L.: 基于协方差矩阵学习和双峰分布参数设定的差分进化。《应用软计算》**18**, 232–247 (2014) 18. Xue, J., Shen, B.: 一种新型群体智能优化方法：麻雀搜索算法。《系统科学与控制工程》**8**(1), 22–34 (2020) 19. Ye, C., Li, C., Li, Y., Sun, Y., Yang, W., Bai, M., Zhu, X., Hu, J., Chi, T., Zhu, H., et al.: 变异策略在稳定垄断与短暂竞争之间交替的差分进化。《群体与进化计算》**83**, 101403 (2023) 20. Zhu, X., Ye, C., He, L., Zhu, H., Chi, T., Hu, J.: 差分进化与获取-共享知识算法的集成，基于适应度和生命周期选择个体并进行交换。《软计算》第1–16页（2023）