

# 基于超支配自适应收敛性计算的超多目标进化算法研究

王梦真\*, 唐春慧, 刘云青, 朱 凯, 朱 琪

阜阳幼儿师范高等专科学校健康和管理学院, 安徽 阜阳

收稿日期: 2025年3月18日; 录用日期: 2025年4月17日; 发布日期: 2025年4月25日

## 摘 要

针对超多目标优化问题中传统帕累托支配失效及参考点选择对收敛性度量的关键影响, 提出一种基于超支配自适应收敛性计算的超多目标进化算法(HACCEA)。该算法通过超支配方法有效区分非支配解与支配解, 结合自适应收敛性计算机制动态调整参考点, 以精确估计帕累托前沿(PF)形状, 进而优化收敛性度量。环境选择阶段采用行列式点过程平衡收敛性与多样性。在IGD指标上, 相较于4个先进算法, HACCEA优势显著, 鲁棒性强。

## 关键词

超多目标优化, 进化算法, 超支配, 自适应收敛性, PF形状

# Research on Many-Objective Evolutionary Algorithms Based on Hyper-Dominated Adaptive Convergence Calculation

Mengzhen Wang\*, Chunhui Tang, Yunqing Liu, Kai Zhu, Qi Zhu

School of Health and Management, Fuyang Preschool Teachers College, Fuyang Anhui

Received: Mar. 18<sup>th</sup>, 2025; accepted: Apr. 17<sup>th</sup>, 2025; published: Apr. 25<sup>th</sup>, 2025

## Abstract

In order to address the failure of traditional Pareto domination and the critical influence of reference point selection on the convergence in many-objective optimization problems, a many-

\*通讯作者。

文章引用: 王梦真, 唐春慧, 刘云青, 朱凯, 朱琪. 基于超支配自适应收敛性计算的超多目标进化算法研究[J]. 计算机科学与应用, 2025, 15(4): 309-317. DOI: 10.12677/csa.2025.154103

**objective evolutionary algorithm based on hyper-dominance adaptive convergence calculation (HACCEA) is proposed. The algorithm effectively distinguishes between non-dominated and dominated solutions through the hyper-dominance method, and dynamically adjusts the reference points by combining with the adaptive convergence calculation mechanism in order to accurately estimate the shape of the Pareto front (PF), and then optimize the convergence. The environmental selection phase uses a determinantal point process to balance convergence and diversity. On the IGD indicator, HACCEA shows significant advantages and excellent robustness compared to the four state-of-the-art algorithms.**

## Keywords

**Many-Objective Optimization, Evolutionary Algorithms, Hyper-Dominance, Adaptive Convergence, PF Shape**

Copyright © 2025 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY 4.0).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

## 1. 引言

在实际应用中经常会出现多个目标之间相互冲突的优化问题(例如, 家庭能源管理系统[1]), 这类问题中需要优化的目标数量往往大于三个, 即超多目标优化问题(MaOPs) [2]。超多目标进化算法(MaOEA)是解决这类问题的有效途径, 但是由于目标数量的增加, 解在目标空间的维度升高, 传统的帕累托支配方法失效, 大多数解都变成了非支配解, 导致算法的选择压力丧失, 解的收敛性较差。同时, 目标空间维度的升高导致解的分布变得稀疏, 种群中的多样性难以保持, 这也使得算法难以平衡收敛性和多样性 [3]。

因此, 提出一种基于超支配自适应收敛性计算的超多目标进化算法(HACCEA), 通过超支配方法有效区分非支配解与支配解, 结合自适应收敛性计算机制动态调整参考点, 以精确估计 PF 形状(凹、凸或线性), 进而优化收敛性度量。环境选择阶段采用行列式点过程构造兼顾收敛性与多样性的核矩阵, 无需参数调整即可实现收敛性和多样性的平衡。

## 2. 相关工作

超多目标优化问题(MaOPs)是指目标数量超过三个的多目标优化问题, 定义如下:

$$\min \mathbf{F}(\mathbf{x}) = (f_1(\mathbf{x}), f_2(\mathbf{x}), \dots, f_m(\mathbf{x})), \mathbf{x} \in \mathbf{D} \quad (1)$$

其中:  $m(m > 3)$  为目标数,  $\mathbf{x} = (x_1, x_2, x_3, \dots, x_n)$  是  $n$  维决策变量,  $\mathbf{D}$  是由  $\mathbf{x}$  组成的决策空间,  $\mathbf{F}(\mathbf{x})$  是由目标函数组成的  $m$  维的目标向量。MOPs 包含多个相互冲突的目标, 这意味着并非所有的最优解都能实现所有目标的最优值; 因此, 得到的解不是单个最优解, 而是多个达到最佳权衡的非支配解, 称为帕累托最优解集, 这些解被映射到目标空间, 形成帕累托最优前沿(PF)。

帕累托支配的定义如下: 对于一个最小化优化问题, 如果  $\mathbf{x}$  的所有目标值小于  $\mathbf{y}$  的所有目标值, 即  $f_i(\mathbf{x}) \leq f_i(\mathbf{y}), \forall i = 1, 2, \dots, m$ , 且存在  $f_i(\mathbf{x}) < f_i(\mathbf{y}), \exists i = 1, 2, \dots, m$ , 则称  $\mathbf{x}$  支配  $\mathbf{y}$ , 即  $\mathbf{x} \prec \mathbf{y}$ ,  $\mathbf{x}$  和  $\mathbf{y}$  分别是非支配解和支配解[3]。

为了解决 MaOPs, 已经提出了多种超多目标进化算法(MaOEA)。这些算法大体上可以分为基于帕

累托支配的、基于分解的、基于指标的和基于多样性的，其中基于帕累托支配的算法通过修改或放松帕累托支配关系增强选择压力，如 NSGAI-SDR [4] 通过为每个候选解设置独立的小生境并自适应调整其大小来维持种群多样性，同时在每个小生境内选取具有最优收敛性的非支配解，但是这类算法需要额外的参数设置。基于多样性的算法通过设计多样性维护方法来提高解的多样性，使解能沿 PF 均匀分布，但是容易忽略解的收敛性，难以实现收敛性和多样性的平衡；基于分解的算法将复杂的多目标优化问题转化为简单的子问题，并协同优化这些子问题。基于指标的算法根据每个解的指标值选择解，能够有效解决选择压力不足的问题，如 MaOEA/IGD [5]，采用 IGD 指标作为解的选择标准，但是计算开销较大[6]。

因此，本研究提出一种基于超支配自适应收敛性计算的超多目标进化算法(HACCEA)，通过超支配方法有效区分非支配解与支配解，结合自适应收敛性计算机制动态调整参考点，以精确估计 PF 形状(凹、凸或线性)，进而优化收敛性度量，环境选择阶段采用行列式点过程平衡收敛性和多样性。

### 3. 超多目标进化算法(HACCEA)

本节主要介绍 HACCEA 的算法结构。如算法 1 所示，HACCEA 主要由三部分组成：超支配方法、自适应收敛性计算(ACC)和环境选择。首先，初始化一个包含  $N$  个个体的种群  $P$ ，执行模拟二进制交叉和多项式变异[7]来生成子代  $Q$ ，然后合并  $P$  和  $Q$ ，应用超支配方法计算个体的支配程度，根据每个个体的支配程度进行优胜劣汰，选择非支配程度较高的个体加入候选种群  $P$ ，同时将非支配程度最高的个体  $x_k$  作为关键个体来更加准确的估计 PF 的形状，进而自适应的计算个体的收敛性。在环境选择中，使用自适应计算得到的收敛性和余弦相似度来构造核矩阵  $L$ 。最后，使用行列式点过程(DPP)从  $L$  中抽样最优个体 [8]。下面我们将依次详细介绍这些部分。

**算法 1:** HACCEA 框架

**输入:** 种群规模  $N$

**输出:** 种群  $P$

- 1:  $P = \text{initialize}(N)$ ; //种群初始化
- 2: **while** maxFEs is not reached **do**
- 3:  $\text{Parent} = \text{mating strategy}(P)$ ;
- 4:  $Q = \text{variation}(\text{Parent})$ ; //交叉变异生成子代  $Q$
- 5:  $(P, x_k) = \text{hyper-dominance method}(P, Q, N)$ ; //超支配方法区分解
- 6:  $\text{con} = \text{adaptive calculate convergence}(P, x_k)$ ; //自适应计算收敛性
- 7:  $P = \text{DPP}(P, N, \text{con})$ ; //采用行列式点过程 DPP 选择优秀个体
- 8: **end while**

#### 3.1. 超支配方法

为了解决高维目标空间中无法有效区分解的问题，我们采用超支配方法来评估解的非支配程度，从而将较好的非支配解纳入到下一代种群进行迭代优化，并淘汰掉支配解。超支配方法定义如公式(2)和公式(3)所示：

假设种群中有  $n$  个个体，该种群可以描述为  $S = (x_1, x_2, \dots, x_n)$ ，则个体  $i$  的超支配度  $h_{d_i}$  可以表示为：

$$h_{d_i} = \min(H(x_i, x_1), H(x_i, x_2), \dots, H(x_i, x_n)) \quad (2)$$

其中：

$$H(\mathbf{x}_i, \mathbf{x}_j) = \begin{cases} \infty, & (\mathbf{x}_i - \mathbf{x}_j) \leq 0 \\ -\frac{\max(\mathbf{x}_i - \mathbf{x}_j)}{\min(\mathbf{x}_i - \mathbf{x}_j)}, & (\mathbf{x}_i - \mathbf{x}_j) > 0 \end{cases} \quad (3)$$

其中： $H(\mathbf{x}_i, \mathbf{x}_j)$  是用于反映  $\mathbf{x}_i$  相对于  $\mathbf{x}_j$  支配情况的指标，当值为无穷大时，意味着个体  $\mathbf{x}_i$  在帕累托支配意义上优于个体  $\mathbf{x}_j$ ；当值为负数时，意味着  $\mathbf{x}_i$  在帕累托支配意义上被  $\mathbf{x}_j$  支配；当值为正数时，则表明个体  $\mathbf{x}_i$  和  $\mathbf{x}_j$  相互非支配，数值越大表明  $\mathbf{x}_i$  在收敛性方面相对于  $\mathbf{x}_j$  越有优势。

超支配度  $h_{d_i}$  是通过将  $\mathbf{x}_i$  与种群中的其他个体进行超支配比较所取得的最小值，见公式(2)，它能反映出  $\mathbf{x}_i$  相对于当前种群中其他个体的支配程度，当  $h_{d_i}$  值为无穷大时，表示个体  $\mathbf{x}_i$  在帕累托意义上优于当前种群中的所有个体；当  $h_{d_i}$  值为负时，表示种群中至少有一个个体在帕累托意义上优于  $\mathbf{x}_i$ ；当  $h_{d_i}$  值为正时，表示个体  $\mathbf{x}_i$  与种群中的其他个体之间是非支配关系。

$h_{d_i}$  的值越大，表明  $\mathbf{x}_i$  相对于种群中的其他个体具有更大的优势，所以在本算法中，我们采用超支配方法来区分解，将  $h_{d_i} > 0$  的个体(即非支配解)纳入到候选种群  $\mathbf{P}$  中进行迭代优化，若非支配解的数量小于种群规模  $N$  时，则采用补充操作，从  $h_{d_i} < 0$  的个体中选择较优的个体填充候选种群  $\mathbf{P}$ 。同时，将使  $h_{d_i}$  取最大值的个体  $\mathbf{x}_k$  作为关键解在 ACC 中估计 PF 的形状。

### 3.2. 自适应收敛性计算

ACC 方法首先通过关键解  $\mathbf{x}_k$  估计 PF 的形状，然后根据 PF 的形状自适应确定参考点来计算个体的收敛性  $con(x)$ ，具体过程如下：

1) **PF 形状估计**：通过比较关键解  $\mathbf{x}_k$  与理想点  $\mathbf{Z}_{\min}$  之间的距离  $d_1$  和理想点  $\mathbf{Z}_{\min}$  到超平面之间的距离  $d_2$  来判断当前 PF 的形状，如果  $d_1 < d_2$ ，则 PF 为凹形；如果  $d_1 > d_2$ ，则 PF 为凸形；如果  $d_1 = d_2$ ，则 PF 为线性。其中理想点  $\mathbf{Z}_{\min}$  是由每个目标函数的最小值组成的  $m$  维向量，通过归一化之后，理想点变为目标原点  $\mathbf{O}$ ；最低点  $\mathbf{Z}_{\max}$  是由每个目标函数的最大值组成的  $m$  维向量，如图 1 和图 2 所示。

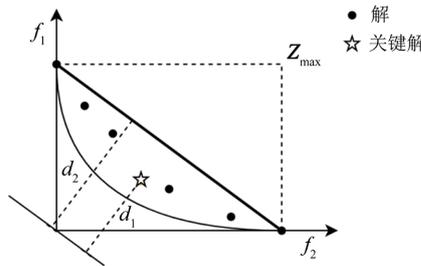


Figure 1. Concave PF  
图 1. 凹形 PF

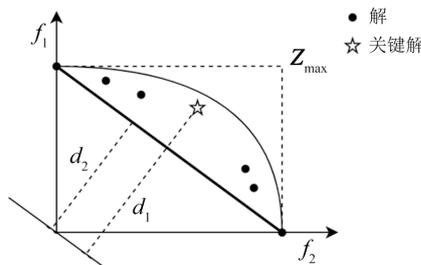


Figure 2. Convex PF  
图 2. 凸形 PF

2) 自适应计算收敛性  $con(x)$ : 根据 PF 形状自适应确定合适的参考点, 若 PF 是凹形, 则选择理想点为参考点, 通过计算个体到通过理想点  $O$  的超平面的距离来表示个体的收敛性, 如图 3 所示; 若 PF 是凸形, 则选择最低点  $Z_{max}$  为参考点, 通过计算个体到通过最低点  $Z_{max}$  的超平面的距离来表示个体的收敛性, 如图 4 所示; 若 PF 是线性, 则使用目标函数值的和来表示个体的收敛性, 即  $con(x) = \sum_{i=1}^m f_i(x)$ 。

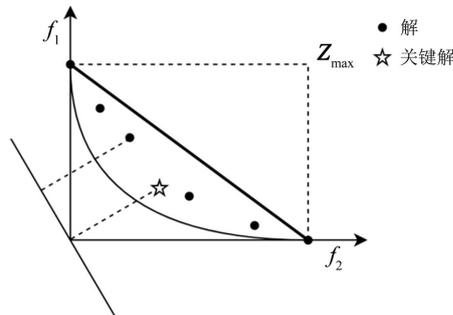


Figure 3. Convergence measures for concave PF  
图 3. 凹形 PF 的收敛性度量

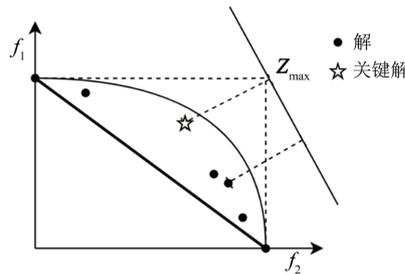


Figure 4. Convergence measures for convex PF  
图 4. 凸形 PF 的收敛性度量

### 3.3. 环境选择

在环境选择的阶段, 采用行列式点过程(DPP)来选择具有较好收敛性和多样性的解, 实现收敛性和多样性的平衡。DPP 是一种概率模型, 可将复杂的概率计算转换为简单的行列式计算[8], 它从一个大集合中选择一个子集, 适用于进化算法的环境选择。在由点过程  $P$  进行抽样从而得到随机子集  $Y$  的情况下, 对于任何满足  $A \subseteq Y$  的集合  $A$  而言, 存在如公式(4)所示的关系:

$$P_L(A=Y) = \frac{\det(L_A)}{\det(L+I)} \quad (4)$$

其中: 矩阵  $L$  为实对称半正定矩阵,  $L_A = L_{i,j} (i, j \in A)$ ,  $I$  是  $N$  阶单位矩阵,  $\det(\cdot)$  表示行列式。

DPP 通过构建核矩阵  $L$ , 在  $L$  中选择优秀解, 与传统的选择策略相比, 它的设计思路更为简单, 只需要在构造核矩阵  $L$  时兼顾收敛性和多样性, 即可实现收敛性和多样性的平衡。由于  $L$  具有实对称的特性, 故其可写为  $L = BB^T$  的形式, 则其中  $L_{i,j}$  可用公式(5)表示:

$$L_{i,j} = q_i \phi_i \phi_j q_j \quad (5)$$

其中:  $q_i$  代表第  $i$  个元素所具有的质量,  $S_{i,j} = \phi_i \phi_j$  表示第  $i$  个元素和第  $j$  个元素之间的相似度。因此, 可以使用公式(6)来构建核矩阵  $L$ :

$$L = q(x)D(x,y)q(y) \quad (6)$$

其中:  $q(x)$  表示解  $x$  的收敛性,  $D(x, y)$  表示解  $x$  和  $y$  之间的余弦相似度, 具体地,  $D(x, y) = \exp(-\cos(x, y))$ , 用于表示多样性,  $q(x) = 1/\text{con}(x)$ , 表示收敛性。核矩阵  $L$  中的值越大, 其被选中的概率就越高, 说明解的收敛性和多样性越好。

## 4. 实验测试与分析

### 4.1. 参数设置

本节将 HACCEA 与算法 MaOEA/IGD [5]、lby1EA [9]、NSGAI-SDR [4] 和 PREA [8] 在 3、5 和 15 个目标上进行了比较。我们选择基准问题 WFG1-9 [10] 测试 HACCEA 解决各种问题的能力。其中, WFG1 是一个可分离的单模态问题, 具有凹形 PF; WFG2 的 PF 是按比例断开的; WFG3 是一个线性退化问题; WFG4 用于测试算法的全局搜索能力。WFG5-9 的 PF 均为凹形, 其中 WFG5 是欺骗性的, WFG7 是可分离的。

本实验所采用的平台是 PlatEMO [11], 最大函数评价次数 maxFEs 设置为 100,000 次, 根据不同的目标数量  $m$  设置不同的种群规模  $N$  和决策数量  $D$ , 如表 1 所示。

Table 1. Parameter settings

表 1. 参数设置

目标数量 $m$	决策变量 $D$	种群规模 $N$
3	12	91
5	14	105
15	24	135

### 4.2. 结果分析

在本小节中, 我们根据每种算法在 WFG1-9 (27 个测试实例) 上 20 次独立运行的统计结果, 对 HACCEA 进行分析和评估。使用的性能指标是 IGD [5], 它通过测量真实 PF 与算法获得的 PF 之间的接近程度来评估收敛性和多样性, IGD 值越小说明算法的性能越好。同时, 采用 Wilcoxon 秩和检验来对比不同算法的表现, 显著性水平设置为 0.05, 符号 “+”、“-” 和 “=” 分别表示测试结果显著优于 HACCEA、显著劣于 HACCEA、与 HACCEA 无显著差异。表 2 显示了 HACCEA 与 4 个先进算法在 27 个基准问题上的性能比较结果, 可见在 IGD 指标上, HACCEA 显著优于其他 4 个先进算法, 表现出很强的竞争力。

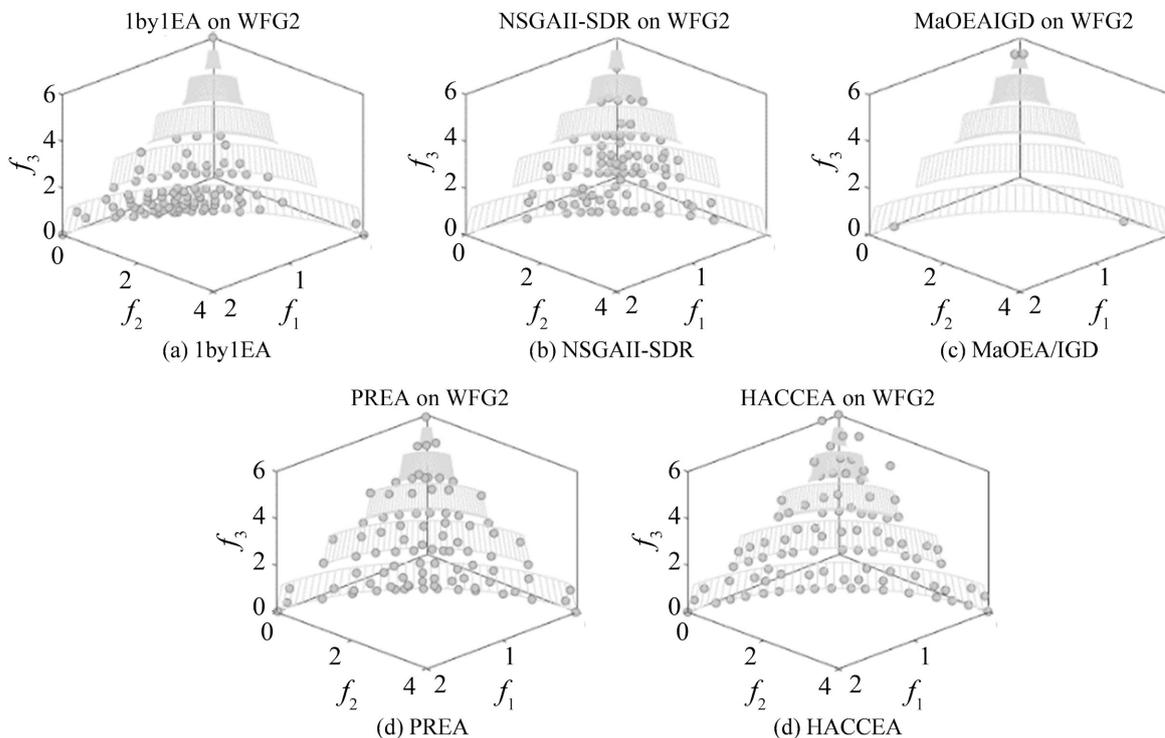
从表 2 中可以明显看出 HACCEA 的表现优于 4 个对比算法, 所有实例中优于对比算法所占的比例分别为 20/27、27/27、25/27 和 12/27。HACCEA 算法在 WFG1、WFG2、WFG4 和 WFG5 问题上展现出了优秀的性能, 其 IGD 值在所有算法中最低, 说明算法能够有效逼近真实的帕累托前沿(PF)。

图 5 直观反映了 HACCEA 的算法表现, 在基于 3 目标 WFG2 问题的实验中, HACCEA 算法与四个先进算法的解集分布对比表明, HACCEA 在面临复杂的 PF 问题时, 仍然能够表现出优秀的性能。WFG2 的 PF 具有非对称和非线性的特点, 且存在局部最优陷阱, 对算法的收敛性和多样性提出双重挑战。图 5 中可见, HACCEA 生成的解能较均匀地覆盖 PF 的边界和中心区域, 表明其能够较好地平衡收敛性和多样性, 而其他 4 个对比算法取得的解偏离前沿中心, 存在堆积现象, 无法均匀覆盖 PF, 即收敛性和多样性较差。

另外, 从表 2 中可见, HACCEA 在大多数问题上都优于对比算法, 但是 HACCEA 在 WFG7 上性能有所下降, 其表现不如 PREA, WFG7 的 PF 呈现线性分段特点, 而本算法的 PF 形状判断方法是将真实的帕累托前沿预设为简单连续的, 这可能是影响 HACCEA 性能的一个原因。

**Table 2.** Mean (standard deviation) of the IGD indicator values obtained by HACCEA and the four state-of-the-art algorithms on the WFG1-9 problems**表 2.** HACCEA 与 4 个先进算法在 WFG1-9 问题上取得的 IGD 指标值的平均值(标准差)

Problem	M	NSGAII-SDR	IbylEA	MaOEA/IGD	PREA	HACCEA
WFG1	3	1.9968e-1 (1.11e-2)-	3.1786e-1 (2.91e-2)-	2.2756e+0 (7.88e-2)-	1.6907e-1 (6.40e-3)-	<b>1.4768e-1</b> <b>(2.04e-3)</b>
	5	7.6818e-1 (6.43e-2)-	8.0626e-1 (5.37e-2)-	3.7882e+0 (1.12e-1)-	6.1334e-1 (3.66e-2)-	<b>4.4540e-1</b> <b>(6.22e-3)</b>
	15	2.4786e+0 (2.76e-2)-	2.4227e+0 (1.78e-2)-	1.0822e+1 (9.06e+0)-	2.4414e+0 (1.89e-1)-	<b>1.9644e+0</b> <b>(5.51e-2)</b>
WFG2	3	2.0119e-1 (6.03e-3)-	2.6687e-1 (1.28e-2)-	1.5061e+0 (2.68e-1)-	1.8304e-1 (3.63e-3)-	<b>1.7238e-1</b> <b>(4.36e-3)</b>
	5	5.7647e-1 (4.36e-2)-	7.9497e-1 (8.34e-2)-	1.9529e+0 (4.58e-1)-	6.1591e-1 (2.73e-2)-	<b>4.5836e-1</b> <b>(4.68e-3)</b>
	15	2.3827e+0 (5.97e-2)-	2.4940e+0 (4.44e-2)-	3.9042e+0 (6.76e-1)-	2.6353e+0 (2.09e-1)-	<b>2.2109e+0</b> <b>(9.93e-2)</b>
WFG3	3	1.0045e-1 (7.24e-3)+	4.6582e-1 (2.18e-2)-	3.1985e+0 (5.82e-3)-	<b>8.5120e-2</b> <b>(6.60e-3)+</b>	1.2002e-1 (1.76e-2)
	5	<b>4.2680e-1</b> <b>(6.34e-2)=</b>	1.3501e+0 (1.23e-1)-	5.4339e+0 (2.09e-2)-	5.1428e-1 (6.57e-2)=	4.6157e-1 (2.19e-2)
	15	5.6225e+0 (1.55e+0)-	9.5445e+0 (3.68e-1)-	1.2426e+1 (6.24e+0)-	4.7850e+0 (8.70e-1)-	<b>3.1232e+0</b> <b>(3.46e-1)</b>
WFG4	3	2.5206e-1 (7.16e-3)-	3.9410e-1 (7.18e-2)-	3.5907e+0 (6.21e-1)-	2.3836e-1 (6.43e-3)=	<b>2.3631e-1</b> <b>(1.31e-3)</b>
	5	1.2633e+0 (2.34e-2)-	1.8543e+0 (1.26e-1)-	6.4494e+0 (7.69e-4)-	1.2169e+0 (5.66e-3)=	<b>1.2101e+0</b> <b>(1.20e-2)</b>
	15	9.2959e+0 (3.48e-1)=	1.1655e+1 (3.32e-1)-	2.7148e+1 (2.17e+0)-	<b>8.4473e+0</b> <b>(1.04e-1)+</b>	9.4749e+0 (5.06e-1)
WFG5	3	2.7042e-1 (8.25e-3)-	3.6838e-1 (5.22e-2)-	1.8890e+0 (1.60e+0)-	2.4618e-1 (2.85e-3)=	<b>2.4592e-1</b> <b>(2.29e-3)</b>
	5	1.2388e+0 (2.18e-2)-	1.7601e+0 (1.42e-1)-	5.7338e+0 (2.20e+0)-	1.1982e+0 (1.75e-2)=	<b>1.1948e+0</b> <b>(1.53e-2)</b>
	15	9.2351e+0 (9.88e-2)=	1.1335e+1 (3.16e-1)-	2.6152e+1 (5.23e+0)-	<b>8.5790e+0</b> <b>(1.30e-1)+</b>	9.1061e+0 (1.00e-1)
WFG6	3	2.7360e-1 (1.24e-2)=	5.3675e-1 (7.39e-2)-	3.7708e+0 (3.34e-1)-	<b>2.5265e-1</b> <b>(9.67e-3)=</b>	2.5936e-1 (9.74e-3)
	5	1.2796e+0 (5.04e-2)-	2.2633e+0 (9.77e-2)-	6.3164e+0 (1.07e+0)-	1.2401e+0 (1.08e-2)-	<b>1.1892e+0</b> <b>(1.11e-2)</b>
	15	9.7654e+0 (4.35e-1)=	1.2051e+1 (5.30e-1)-	2.3842e+1 (9.65e+0)=	<b>8.2822e+0</b> <b>(8.31e-2)+</b>	9.2482e+0 (1.92e-1)
WFG7	3	2.5599e-1 (8.54e-3)-	6.4563e-1 (6.62e-2)-	2.7298e+0 (7.24e-1)-	2.4058e-1 (2.94e-3)-	<b>2.3593e-1</b> <b>(1.75e-3)</b>
	5	1.2504e+0 (1.28e-2)-	2.3545e+0 (1.54e-1)-	6.0480e+0 (7.32e-1)-	1.2607e+0 (1.53e-2)-	<b>1.2071e+0</b> <b>(9.95e-3)</b>
	15	1.0015e+1 (1.55e+0)-	1.1097e+1 (7.75e-1)-	2.5861e+1 (4.02e+0)-	<b>8.2405e+0</b> <b>(4.98e-2)+</b>	9.0280e+0 (1.30e-1)
WFG8	3	3.2956e-1 (4.49e-3)-	5.7403e-1 (2.46e-2)-	3.6348e+0 (4.06e-2)-	<b>2.9312e-1</b> <b>(4.86e-3)=</b>	3.0258e-1 (6.48e-3)
	5	1.2858e+0 (9.73e-3)-	1.8959e+0 (1.28e-1)-	4.8401e+0 (1.93e+0)-	1.2431e+0 (1.63e-2)-	<b>1.1791e+0</b> <b>(1.13e-2)</b>
	15	1.0753e+1 (1.39e+0)-	1.2065e+1 (8.09e-1)-	2.5026e+1 (3.41e+0)-	<b>8.4308e+0</b> <b>(1.57e-1)+</b>	9.1181e+0 (5.36e-1)
WFG9	3	2.6068e-1 (1.03e-2)-	3.7246e-1 (6.42e-2)-	2.4763e+0 (9.17e-1)-	2.3789e-1 (3.32e-3)-	<b>2.3283e-1</b> <b>(1.07e-3)</b>
	5	1.2431e+0 (1.84e-2)-	1.7670e+0 (1.02e-1)-	5.5005e+0 (1.63e+0)-	1.1982e+0 (5.21e-3)=	<b>1.1930e+0</b> <b>(1.62e-2)</b>
	15	8.9269e+0 (2.12e-1)=	1.0227e+1 (2.16e-1)-	2.3562e+1 (8.61e+0)=	<b>8.2000e+0</b> <b>(1.03e-1)+</b>	8.8581e+0 (1.40e-1)
+/-/=		<b>1/20/6</b>	<b>0/27/0</b>	<b>0/25/2</b>	<b>7/12/8</b>	



**Figure 5.** Comparison of solution set distributions between HACCEA and four advanced algorithms on the 3-objective WFG2 problem

**图 5.** HACCEA 与 4 个先进算法在 3 目标的 WFG2 问题上的解集分布

综上, HACCEA 优势显著, 在解决不同的问题时能够保持较为稳定的性能, 这对于实际应用来说是十分重要的。但是在处理部分帕累托前沿不连续的问题时, 性能略有下降, 未来将考虑改进现有的搜索机制, 以提高算法的性能, 增强算法的通用性, 使算法能够更好地适应不同类型的优化问题。

## 5. 结论

本文深入研究了超多目标进化算法中如何更准确的度量解的收敛性, 以及收敛性和多样性的平衡问题, 提出一种基于超支配自适应收敛性计算的超多目标进化算法(HACCEA), 通过超支配方法区支配解和非支配解, 从中选取非支配程度最高的个体估计 PF 的形状, 进而自适应选择合适的参考点计算解的收敛性, 并利用行列式点过程平衡收敛性与多样性。实验表明, HACCEA 在 WFG1~9 基准问题上显著优于对比算法, 且稳定性强。但是, HACCEA 在不连续 PF 问题上仍有改进的空间, 未来将进一步优化算法性能, 并将其应用于实际问题。

## 6. 本文创新点归纳

创新点 1: 超支配方法计算解的非支配程度, 区分解的优势程度。

创新点 2: 估计 PF 的形状, 根据形状自适应确定参考点, 进而准确度量解的收敛性。

创新点 3: 采用行列式点过程平衡收敛性和多样性, 无需复杂的参数设置。

## 基金项目

阜阳幼儿师范高等专科学校校级重点科研项目(ZK2024002、ZK2024001、SK2024013); 安徽省质量工程一般项目(YJS20210463); 安徽省大数据服务十大新兴产业特色专业课题(2023sdxx255)。

## 参考文献

- [1] Tostado-Véliz, M., Kamel, S., Aymen, F. and Jurado, F. (2022) A Novel Hybrid Lexicographic-IGDT Methodology for Robust Multi-Objective Solution of Home Energy Management Systems. *Energy*, **253**, Article ID: 124146. <https://doi.org/10.1016/j.energy.2022.124146>
- [2] Zhang, M., Wang, L., Li, W., Hu, B., Li, D. and Wu, Q. (2021) Many-Objective Evolutionary Algorithm with Adaptive Reference Vector. *Information Sciences*, **563**, 70-90. <https://doi.org/10.1016/j.ins.2021.01.015>
- [3] 熊志坚. 面向超多目标优化问题的进化算法研究[D]: [博士学位论文]. 秦皇岛: 燕山大学, 2022.
- [4] Tian, Y., Cheng, R., Zhang, X., Su, Y. and Jin, Y. (2019) A Strengthened Dominance Relation Considering Convergence and Diversity for Evolutionary Many-Objective Optimization. *IEEE Transactions on Evolutionary Computation*, **23**, 331-345. <https://doi.org/10.1109/tevc.2018.2866854>
- [5] Sun, Y., Yen, G.G. and Yi, Z. (2019) IGD Indicator-Based Evolutionary Algorithm for Many-Objective Optimization Problems. *IEEE Transactions on Evolutionary Computation*, **23**, 173-187. <https://doi.org/10.1109/tevc.2018.2791283>
- [6] Wang, M., Ge, F., Chen, D. and Liu, H. (2022) A Many-Objective Evolutionary Algorithm with Adaptive Convergence Calculation. *Applied Intelligence*, **53**, 17260-17291. <https://doi.org/10.1007/s10489-022-04296-4>
- [7] 王梦真. 面向超多目标优化问题的进化算法研究[D]: [博士学位论文]. 淮北: 淮北师范大学, 2023.
- [8] Zhang, P., Li, J., Li, T. and Chen, H. (2021) A New Many-Objective Evolutionary Algorithm Based on Determinantal Point Processes. *IEEE Transactions on Evolutionary Computation*, **25**, 334-345. <https://doi.org/10.1109/tevc.2020.3035825>
- [9] Liu, Y., Gong, D., Sun, J. and Jin, Y. (2017) A Many-Objective Evolutionary Algorithm Using a One-by-One Selection Strategy. *IEEE Transactions on Cybernetics*, **47**, 2689-2702. <https://doi.org/10.1109/tycb.2016.2638902>
- [10] Huband, S., Hingston, P., Barone, L. and While, L. (2006) A Review of Multiobjective Test Problems and a Scalable Test Problem Toolkit. *IEEE Transactions on Evolutionary Computation*, **10**, 477-506. <https://doi.org/10.1109/tevc.2005.861417>
- [11] Tian, Y., Cheng, R., Zhang, X. and Jin, Y. (2017) Platemo: A MATLAB Platform for Evolutionary Multi-Objective Optimization [Educational Forum]. *IEEE Computational Intelligence Magazine*, **12**, 73-87. <https://doi.org/10.1109/mci.2017.2742868>