

文章编号:1671-6833(2025)06-0032-08

基于适应度地形分析的优化算法调度方法

朱晓东,任春晓,刘晓兰,陈科,余春明

(郑州大学 电气与信息工程学院,河南 郑州 450001)

摘要: 由于不同的优化问题具有不同的适应度地形,而一种优化算法通常只在某一种适应度地形上有更好的效果,因此,提出了一种基于适应度地形分析的优化算法调度方法(FL-AMAS)。首先,通过提取优化目标函数的局部峰簇数特征来描述优化问题的地形特征,根据地形特征选择相应具有优势的算法,利用对算法的调度发挥不同算法的最大优势;其次,根据优化问题对探索性与开发性的平衡要求,选择了具有高开发能力的哈里斯鹰优化算法(HHO)和具有高探索能力的差分进化算法(DE)作为调度使用的算法,根据不同的适应度地形特征来选择更适合的算法。实验结果表明:在基准测试集上,相较于单独使用HHO,FL-AMAS在收敛性能上提升了75%;与DE算法相比,FL-AMAS收敛性能提升了40%。将FL-AMAS与6种先进算法进行比较,在75%的基准测试集上,FL-AMAS的收敛精度均优于这些算法。通过调度其他类型优化算法的结果进行对比,也验证了所提调度方法的有效性和扩展性。

关键词: 优化算法调度; 适应度地形; 特征提取; 局部峰值点; 哈里斯鹰优化算法; 差分进化算法

中图分类号: TP391;TP301.4;TP18 **文献标志码:** A **doi:**10.13705/j.issn.1671-6833.2025.03.017

现实生活中的许多优化问题都可以用优化模型来表达^[1],在工程领域和日常生活应用中,元启发式算法已被证明是一种有效且实用的算法^[2],相较于梯度下降、分支界限等优化方法,元启发式算法可以避免陷入局部最优并能够更有效地得到全局最优解。

适应度地形分析是一种用于发掘问题本身信息的方法,它反映了问题的特征属性^[3]。通过准确表达这些特征,有助于选择或开发更适合此类问题特征的优化算法。这些优化算法不能在每个优化问题上都优于其他单一的算法,每种算法需要考虑的主要问题是算法中探索和开发的平衡,更多的探索意味着算法需要更多资源寻找最优解,尤其是对于复杂、多峰的优化问题可以避免陷入局部最优;而更多的开发则是让算法可以更好地利用已知局部信息,提高局部搜索的精度和效率,有助于算法快速收敛于局部最优解^[4]。

对于不同的问题特征,优化算法需要有不同的侧重点^[5]。近年来,部分学者通过利用问题的适应度地形信息,自适应地调整算法策略与参数,以期提

升优化算法的性能^[6];同时,另一部分研究者致力于发掘算法本身的内在特性,并基于这些特性预测从算法组合中选择最佳算法进行下一步迭代^[7-8]。然而,这些研究工作往往未能充分结合问题的地形特征与算法特性,导致其在实际应用中存在显著的局限性。

为了能够结合优化问题特征选择更合适的优化算法,本文提出了一种基于适应度地形特征的多种优化算法调度方法(meta-heuristic algorithm scheduling based on fitness landscape analysis, FL-AMAS), 通过从复杂的优化问题中提取问题特征,将问题特征划分为地形复杂问题和地形简单问题。对地形复杂问题采用探索能力强的算法,对地形简单问题采用开发能力强的算法。

在众多优化算法中,差分进化(differential evolution, DE)算法^[9]具有较好的探索能力,而哈里斯鹰优化(harris hawks optimization, HHO)算法^[10]则具有较好的开发能力。汪慎文等^[11]通过分析差分进化算法的控制参数及策略,强调该算法在处理复杂问题时具有更强的探索性能。Heidari等^[10]提出

收稿日期:2025-01-26;修订日期:2025-03-02

基金项目:国家自然科学基金青年项目(62206255);中国博士后科学基金面上项目(2022M712878)

作者简介:朱晓东(1970—),男,河南郑州人,郑州大学副教授,博士,主要从事智能控制及智能信息处理方面的研究, E-mail:Zhu_xd@zzu.edu.cn。

引用本文:朱晓东,任春晓,刘晓兰,等. 基于适应度地形分析的优化算法调度方法[J]. 郑州大学学报(工学版),2025,46(6):32-39. (ZHU X D, REN C X, LIU X L, et al. Optimization algorithm scheduling method based on fitness landscape analysis [J]. Journal of Zhengzhou University (Engineering Science), 2025, 46(6): 32-39.)

的哈里斯鹰优化算法在基准问题测试中表现优秀,具有更好的开发能力与更快的收敛速度,但在处理复杂问题时容易陷入局部最优解的情况。考虑到DE和HHO各自的特性,将它们作为互补类算法,根据问题的适应度地形特征进行动态调度,以最大化发挥各自算法的优势。本文的主要贡献可总结如下:①提出了一种自适应优化算法调度方法,通过分析待解决问题的地形特征来选择更合适的优化算法,当问题的解空间相对简单时,可调度开发性较强的HHO算法;当问题的解空间结构复杂时,则调度探索性较强的DE算法进行求解。②提出了一种新的能够表征问题特征的适应度地形,即局部最优峰值簇NOPC。NOPC=1表示问题空间结构相对简单;NOPC>1表示问题空间结构较为复杂。利用NOPC为优化算法的调度提供依据。

1 相关工作

1.1 优化问题描述

对于无约束单目标数值优化问题在不失一般性的前提下,可以用式(1)进行描述:

$$\min_{x \in \mathbf{R}^n} f(x). \quad (1)$$

式中: $f(x)$ 为在搜索空间 \mathbf{R}^n 中定义的目标函数; $x=(x_1, x_2, \dots, x_D)$ 为目标函数的解向量,求解该数值优化问题就是找到一个解向量使 $f(x)$ 取最小值, D 为优化问题的维度,每一维度的上下边界分别为 x^{\min} 和 x^{\max} 。

1.2 适应度地形

适应度地形最初是由Wright等^[12]提出,用来描述问题空间的复杂程度或特征,通常是由搜索空间中个体的函数适应值来描述。

近期,适应度地形分析已经被用于指导算法策略的选择和确定问题的解决方法。王艳丽等^[13]提出了一种新的自反馈差分进化算法,通过提取各代种群的局部适应度地形特征,并结合地形中单峰和多峰的概率分布,选择最优的变异策略以自适应地解决某类型的优化问题。李亚欣等^[14]在差分进化算法选择设计中利用地形特征,提出了具有适应度距离相关性的多算子DE框架,在进化过程中将更多的权重动态分配给性能最佳的算子从而产生更高质量的解决方案。

上述研究均采用适应度地形来动态选择合适的变异算子或修改权重以提高算法性能,都属于算法内部的调整。如果能够根据地形特征选择更具有优势的算法,对于提高问题解决方案更有帮助。本文

通过NOPC这一指标来描述要解决问题的地形特征,并动态调度更适合的算法进行求解。

1.3 差分进化算法

DE是一种基于种群的优化算法,具有搜索能力强、优化结果稳健等特点。标准的DE结构是由突变算子、交叉概率和选择算子组成^[15]。本文选择DE的主要原因在于其较强的探索性能,DE的探索性能主要受算法中变异策略的影响^[16]。在DE算法中,变异操作通过对个体进行差分变异的方式生成新的候选解。具体而言,变异操作选取种群中的3个不同个体,然后通过线性组合计算差分向量,并将该差分向量与当前个体相加得到变异个体。变异操作引入了随机性和多样性,从而扩大了搜索空间,增强了算法的探索性能。变异策略中最常用的是DE/rand/1,如式(2)所示:

$$v_{i,G} = x_{r1,G} + F \cdot (x_{r2,G} - x_{r3,G}). \quad (2)$$

式中: $x_{r1,G}$ 、 $x_{r2,G}$ 、 $x_{r3,G}$ 为从当前总体中随机选择的不同个体; F 为变异中的缩放因子。

1.4 哈里斯鹰优化算法

Heidari等^[10]受到哈里斯鹰觅食行为的启发,提出了哈里斯鹰优化算法。该算法基于鹰在扑食过程中不同阶段采用的策略,将扑食阶段划分为“寻觅猎物”和“捕捉猎物”两个阶段。本文选用HHO算法主要是利用其显著的开发性能,而该性能主要受算法中捕捉阶段的影响。在HHO算法中,捕捉阶段的优化涉及逃逸能量 E 和猎物逃脱概率随机因子 r ,以及软包围、硬包围、软包围与渐进快速俯冲、硬包围与渐进快速俯冲等4个策略的结合应用。这些策略使得HHO算法具有更强的开发性能,能够迅速而有效地搜索解空间以找到最优解。

1.5 随机游走算法

要获取问题的地形特征,需要对搜索空间进行采样^[17]。本文采用了随机游走算法,由随机游走生成的样本通常可以覆盖整个搜索空间具有代表性的区域。相比于分层采样和基于梯度的采样,随机游走算法具有更广泛、更高效的覆盖性。

随机游走是从一个顶点出发,按照一定的概率随机移动到一个邻居节点,并将该节点作为新的当前节点,如此循环执行若干步,得到一条游走路径:

$$X_n = (x_0, x_1, \dots, x_n). \quad (3)$$

2 算法调度方法

算法调度方法主要包括特征提取和算法调度。特征提取阶段主要是获取待解决优化问题的地形特征,以便选择更合适的算法。地形特征利用局部最

优峰簇数来描述。

2.1 特征提取

局部最优峰值簇(*NOPC*)反映了待解决问题的地形特征。首先,对问题的搜索空间进行随机游走采样,评估采样点的适应度值,提取排序最前列的一部分最优点。其次,通过聚类分析这些点是否位于同一峰上,从而确定峰簇数。

在随机游走采样时,随着维度的增加,搜索空间也会增大,采样的游走步数也将随之增加。考虑到时间预算,本文采用的游走步数为 $500D$,其中 D 为问题维度,步长设定为单维阈值的10%。对于这些采样点,采用DBSCAN聚类方法对前2%的个体进行分类(该比例由下一节的实验确定)。由于整个搜索过程具有一定的随机性,因此对每个测试函数进行30次独立实验,然后从这30次实验结果中选择出现频率最高的两个特征值,利用轮盘赌方法从中选择一个作为最终的特征值,这个最终的特征值定义为*NOPC*。实验结果表明,对于简单优化问题,地形特征较为平滑,通过聚集局部最优点呈现的*NOPC*为1,表现为单峰结构;对于复杂优化问题,地形较为粗糙,呈现的*NOPC*则大于1,表现为多峰结构。

*NOPC*特征提取中的聚类方法采用了DBSCAN算法,该算法是基于密度的聚类方法^[18],其中包含两个重要的参数:邻域半径 R 和核心点邻域内至少包含的样本个数 $minPts$ 。本文 $minPts$ 的值设定为3。

2.2 DBSCAN的半径自适应选择

DBSCAN中的邻域半径对分类结果有重要影响。尤其随着问题维度的增加,搜索空间随之增大,若保持恒定的邻域半径,将无法对提取出的最优点进行有效分类,所以DBSCAN算法中参数 R (邻域半径)的选择非常重要。

本文通过不同测试函数的实验确定*NOPC*特征所对应的维度-半径对,并根据已知的数据对拟合出维度与半径的关系曲线,使得DBSCAN可以根据问题的维度自适应选择合适的邻域半径进行分类。通过实验得到低维空间中几个维度-半径对的具体数值,分别为 $(2, 0.10)$, $(4, 0.40)$, $(6, 0.45)$, $(8, 0.55)$ 。针对这些维度-半径对,分别进行线性函数拟合、指数函数拟合、对数函数拟合、幂函数拟合和多项式函数拟合实验,拟合结果如图1所示。由图1可以看出,效果最好的是线性函数拟合和幂函数拟合,再利用该拟合公式进行高维空间的实验,根据实验结果确定选用效果更好的线性拟合,如式(4)

所示:

$$R = 0.07D + 0.025. \quad (4)$$

根据拟合出的公式,对于某一维度的问题空间,可以自适应地确定合适的聚类邻域半径。例如,当维度 $D=10$ 时,相应的邻域半径 R 为0.725;当 $D=30$ 时, R 为2.125。由此得到的半径能够有效应用于提取适应度地形特征*NOPC*对问题空间进行合理分类。

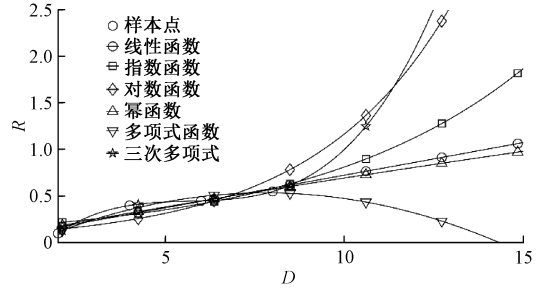


图1 自适应邻域半径预测

Figure 1 Adaptive neighborhood radius prediction

2.3 自适应算法调度方法(FL-AMAS)的步骤

本文提出的自适应算法调度方法(FL-AMAS)整体流程如图2所示。首先通过特征分析得到反映地形特征的*NOPC*值,并根据*NOPC*选择哈里斯鹰优化算法或者差分进化算法,然后进行相应算法的迭代,直到满足终止条件。

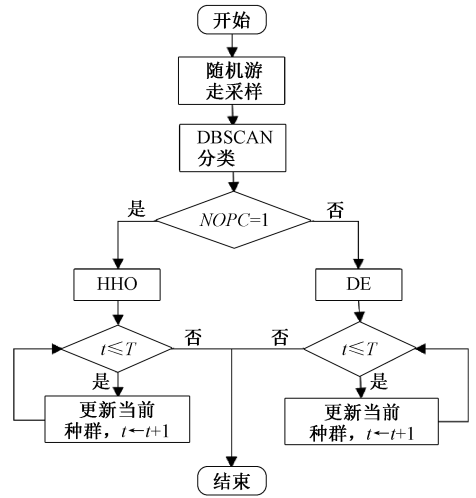


图2 自适应算法调度方法流程图

Figure 2 FL-AMAS flowchart

算法1给出了FL-AMAS的伪代码,随机游走过程中的游走步数设定为 $500D$,对这些随机游走样本的点进行评估的计算复杂度为 $O(500D)$ 。初始化种群的计算复杂度为 $O(NP)$,假设经过 T 次迭代完成整个种群的评估,其计算复杂度为 $O(T \cdot NP)$,算法中其他运算符的计算复杂度为 $O(T)$ 。以二维问题为例,随机游走步数计算复杂度为1000,而 T 次

迭代评估种群的计算复杂度 $T \cdot NP$ 为 15 000,因此 $O(500D) \ll O(T \cdot NP)$ 。综上所述,实现 FL-AMAS 总计算复杂度为 $O(500D) + O(NP) + O(T \cdot NP) + O(T) = O(T \cdot NP)$,与单独使用 DE 或 HHO 的复杂度同属一个量级,因此具有一定的竞争力。

算法 1 FL-AMAS 算法。

输入:DE 和 HHO,基准函数测试集,随机游走次数,最优点占比,种群数 NP ;

输出:种群中的最优解。

- ① 初始化种群 NP 和最大迭代次数 T ;
- ② 设置随机游走步数为 $500D$,最优点占比为 2%;
- ③ 计算随机游走采样个体的适应度;
- ④ 根据最优点占比选择最优个体;
- ⑤ 通过 DBSCAN 对最优个体分类计算 $NOPC$;
- ⑥ if $NOPC = 1$ then
- ⑦ 使用 HHO 更新种群;
- ⑧ $t = 0$;
- ⑨ while $t \leq T$ do
- ⑩ 根据 HHO 策略进行更新;
- ⑪ $t = t + 1$;
- ⑫ end while
- ⑬ else
- ⑭ 使用 DE 更新种群;
- ⑮ $t = 0$;
- ⑯ while $t \leq T$ do
- ⑰ 根据 DE 策略进行更新;
- ⑱ $t = t + 1$;
- ⑲ end while
- ⑳ end if

3 实验评估和结果

3.1 基准函数和实验验证

为了评估所提出的算法调度方法的性能,本文选择了 8 种不同的优化算法作为对比,在相同的一组基准测试函数上进行实验对比。基准函数选择文献中普遍使用、具有不同适应度地形的 23 种函数^[19]。其中函数 $f_1 \sim f_7$ 具有单峰地形,其他为多模态函数。多模态函数中 $f_8 \sim f_{13}$ 是高维地形, $f_{14} \sim f_{23}$ 是固定维的低维地形。在维度可变的基准函数中,本文选择 2 维、10 维和 30 维 3 个不同的维度类型,选择了 $f_1, f_4, f_7, f_{10}, f_{13}, f_{16}, f_{19}, f_{22}$ 这 8 个测试函数为代表进行 FL-AMAS 算法的综合评估。实验中,随机游走采样步数为 $500D$,每个优化算法在相应测试函数上的最大迭代次数设为 500,并对每个测试函数重复运行 30 次取平均结果。实验环境为具有

Intel Core 2.40 GHz CPU 和 8 GB 的计算机,使用 MATLAB R2022a 软件。8 种对比算法分别为 HHO、DE、WOA^[20]、GWO^[21]、MFO^[22]、MVO^[23]、SSA^[24]、SCA^[25],表 1 列出了这些算法的参数取值(未在表中列出的算法则遵循其相关文献中的原始算法参数)。

对每种算法的性能进行统计评估。定义测试函数全局最优解的函数值误差为 $f(x) - f(x^*)$,其中 $f(x)$ 为优化算法得到的最优解, $f(x^*)$ 为函数真实的最优解。计算误差值的均值 (Mean) 和标准差 (Std) 来衡量优化算法的准确性。此外,利用 Friedman 检验^[26]得到了多个算法在所有函数上的排名。

表 1 算法参数设置

Table 1 Algorithm parameter settings

参数	取值	参数	取值
变异因子 F	0.5	随机游走数 RN	$500D$
交叉概率 CR	0.7	最优点采样数	$RN \times 2\%$
逃逸能力 E	线性 2→0	最大迭代次数 T	500
LF β	1.5	种群大小 NP	30

3.2 实验验证

3.2.1 FL-AMAS 与 HHO 和 DE 算法对比

为了验证 FL-AMAS 方法的有效性,在 8 个基准函数集上,将 FL-AMAS 与 HHO 和 DE 算法分别进行对比。表 2 给出了 3 种算法在 2, 10 和 30 的基准函数上的实验统计结果。

表 2 中的 f_{16}, f_{19}, f_{22} 为固定维测试函数,基于 $NOPC$ 值,FL-AMAS 在固定维基准函数上的表现均优于 HHO,并且在 f_{19}, f_{22} 上获得了显著的改善效果。

对于 2 维函数的实验(表 2),根据 $NOPC$ 值,FL-AMAS 较好地结合了 HHO、DE 算法的优点(对 HHO 与 DE 的选择),整体效果与 HHO 算法的性能持平,在 f_1, f_4, f_7 这 3 个函数上优于 DE,但在 f_{13} 上不如 DE。

在 10 维测试函数实验上(表 2),根据 $NOPC$ 值,FL-AMAS 在 f_{13} 上选择 DE 算法,在其余的函数上选择 HHO 算法。与 HHO 相比,在 f_{13} 上结果得到了很大的改善,其余性能持平;与 DE 相比,FL-AMAS 在 f_1, f_4, f_7, f_{10} 这 4 个函数上均优于 DE,在 f_{13} 上性能持平;在 30 维测试函数实验上(表 2),根据 $NOPC$ 值,FL-AMAS 在 f_1, f_4, f_7, f_{10} 和 f_{13} 上选择了 HHO 算法,因此性能与 HHO 持平,与 DE 相比,在 f_1, f_4, f_7 和 f_{10} 上优于 DE,但在 f_{13} 上不如 DE。综上所述,FL-AMAS 根据 $NOPC$ 可以有效地利用 HHO 和 DE 算法优点,在测试函数上得到更好的结果。

表2 不同算法在测试函数 $f_1 \sim f_{23}$ 基准集上的对比Table 2 Comparison of different algorithms on the benchmark set of test functions $f_1 \sim f_{23}$

维度	函数	Mean			Std		
		FL-AMAS	HHO	DE	FL-AMAS	HHO	DE
2 维	f_1	9.49E-112	9.49E-112	6.85E-87	3.67E-111	3.67E-111	2.53E-86
	f_4	3.08E-49	3.08E-49	8.79E-41	1.19E-48	1.19E-48	4.26E-04
	f_7	1.44E-04	1.44E-04	9.50E-04	9.09E-05	9.09E-05	4.26E-04
	f_{10}	4.44E-16	4.44E-16	4.44E-16	5.10E-32	5.10E-32	0
	f_{13}	1.07E-06	1.07E-06	1.35E-32	1.73E-06	1.73E-06	2.83E-48
	f_{16}	4.65E-08	4.69E-08	4.65E-08	5.73E-17	7.54E-10	5.73E-17
	f_{19}	-2.78E-03	3.67E-03	-2.78E-03	0	1.41E-02	0
	f_{22}	-4.03E-01	4.92E+00	-4.03E-01	6.25E-16	7.75E-03	6.25E-16
10 维	f_1	2.02E-93	2.02E-93	4.42E-12	6.13E-93	6.13E-93	2.70E-12
	f_4	6.35E-50	6.35E-50	1.55E-03	2.37E-49	2.37E-49	6.27E-04
	f_7	2.05E-04	2.05E-04	1.78E-02	2.43E-04	2.43E-04	5.54E-03
	f_{10}	4.44E-16	4.44E-16	1.08E-06	5.10E-32	5.10E-32	6.19E-07
	f_{13}	3.96E-12	5.54E-05	3.96E-12	4.89E-12	1.15E-04	4.89E-12
30 维	f_1	5.84E-99	5.84E-99	1.81E-08	1.36E-98	1.36E-98	6.68E-07
	f_4	2.01E-49	2.01E-49	2.71E+00	6.76E-49	6.76E-49	1.86E+00
	f_7	1.31E-04	1.31E-04	2.16E-02	8.04E-05	8.04E-05	6.63E-03
	f_{10}	4.48E-16	4.48E-16	-1.07E+13	3.10E-32	3.10E-32	5.74E+11
	f_{13}	3.51E-05	3.51E-05	1.35E-32	6.11E-05	6.11E-05	2.83E-48

3.2.2 FL-AMAS 与其他算法对比

为了进一步验证 FL-AMAS 的有效性,将 FL-AMAS 与其他类型的算法进行了比较,对比结果如表 3 和表 4 所示。在 2 维函数(含固定维测试集)基准上,FL-AMAS 在统计学结果上优于 MFO、MVO、SCA、SSA、GWO、WOA,其中与 GWO 算法各有千秋,FL-AMAS 在 f_7 和 f_{16} 上优于 GWO,在 f_4 、 f_{13} 和 f_{19} 上

劣于 GWO;在 10 维函数实验中,FL-AMAS 在 f_1 、 f_4 、 f_7 和 f_{13} 函数上都优于这些先进的算法,而在 f_{10} 上不如 SCA、GWO、WOA,这可能是挑选机制的随机性所造成的;在 30 维函数实验中,FL-AMAS 在 f_1 、 f_4 、 f_7 、 f_{10} 和 f_{13} 函数上都优于对比算法。与 10 维函数实验相比,30 维在 f_{13} 上也找到了最优值,由此验证了上述说法的合理性。

表3 不同算法在测试函数 $f_1 \sim f_{23}$ 基准集上的 MeanTable 3 Mean of different algorithms on the benchmark set of test functions $f_1 \sim f_{23}$

维度	函数	Mean						
		FL-AMAS	MFO	MVO	SCA	SSA	GWO	WOA
2 维	f_1	9.49E-112	7.40E-103	3.09E-05	1.76E-68	8.00E-13	1.09E-199	6.78E-108
	f_4	3.08E-49	4.50E-50	4.01E-03	4.97E-34	7.01E-07	8.12E-92	1.11E-16
	f_7	1.44E-04	2.79E-04	2.37E-04	1.90E-04	8.40E-04	1.55E-04	2.41E-04
	f_{10}	4.44E-16	4.44E-16	4.12E-03	4.44E-16	8.67E-07	4.44E-16	1.63E-15
	f_{13}	1.07E-07	1.35E-32	7.58E-06	2.04E-04	6.08E-13	4.79E-08	2.77E-07
	f_{16}	4.65E-08	4.65E-08	3.49E-07	6.05E-05	4.65E-08	7.08E-08	4.69E-08
	f_{19}	-2.78E-03	-2.78E-03	-2.78E-03	5.00E-03	-2.78E-03	-1.71E-03	1.03E-02
	f_{22}	-4.03E-01	1.64E+00	1.48E+00	6.87E+00	1.07E-01	-4.01E-01	1.27E+00
10 维	f_1	2.02E-93	2.55E-13	1.84E-02	1.57E-12	2.23E-07	1.69E-56	8.41E-79
	f_4	6.35E-50	1.10E+00	9.24E-02	3.46E-04	2.07E-05	6.22E-18	7.84E+00
	f_7	2.05E-04	9.70E-03	2.70E-03	2.26E-03	1.46E-02	6.34E-04	1.99E-03
	f_{10}	4.44E-16	2.03E-07	2.60E-01	9.95E-02	8.17E-01	7.31E-15	4.00E-15
	f_{13}	5.54E-05	3.66E-03	5.79E-03	2.96E-01	1.46E-03	6.21E-03	1.25E-02
30 维	f_1	5.84E-99	2.55E-13	1.30E+00	5.11E-11	1.86E-07	1.14E-27	9.98E-78
	f_4	2.01E-49	1.10E+00	1.92E+00	2.33E-03	1.04E+01	6.98E-07	4.43E+01
	f_7	1.31E-04	9.70E-03	3.20E-02	5.50E-03	1.96E-01	1.94E-03	2.29E-03
	f_{10}	4.48E-16	2.03E-07	1.88E+00	3.50E-07	2.78E+00	1.02E-13	4.47E-15
	f_{13}	3.51E-05	3.66E-03	1.70E-01	3.52E-01	1.11E+01	6.51E-01	4.65E-01

表 4 不同算法在测试函数 $f_1 \sim f_{23}$ 基准集上的 Std

Table 4 Std of different algorithms on the benchmark set of test functions $f_1 \sim f_{23}$

维度	函数	Std						
		FL-AMAS	MFO	MVO	SCA	SSA	GWO	WOA
2 维	f_1	3.67E-111	2.80E-102	2.31E-05	4.56E-68	6.55E-13	0	2.62E-107
	f_4	1.19E-48	1.71E-49	2.16E-03	1.68E-33	5.10E-07	3.14E-91	3.27E-16
	f_7	9.09E-05	1.50E-04	2.50E-04	2.00E-04	1.06E-03	1.53E-04	3.00E-04
	f_{10}	5.10E-32	0	3.13E-03	0	5.85E-07	0	2.19E-15
	f_{13}	1.73E-07	2.83E-48	1.02E-05	2.90E-04	7.35E-13	5.14E-08	5.31E-07
	f_{16}	5.73E-17	0.00E+00	2.24E-07	4.74E-05	3.64E-14	2.09E-08	7.49E-10
	f_{19}	0	0	9.68E-07	3.39E-03	3.64E-11	2.19E-03	3.01E-02
	f_{22}	6.25E-16	3.02E+00	3.27E+00	3.11E+00	1.98E+00	9.08E-04	2.90E+00
10 维	f_1	6.13E-93	5.79E-13	9.49E-03	3.59E-12	2.55E-07	6.41E-56	2.29E-78
	f_4	2.37E-49	1.69E+00	4.85E-02	6.56E-04	4.87E-06	1.07E-17	1.05E+01
	f_7	2.43E-04	5.22E-03	1.32E-03	1.33E-03	1.19E-02	3.39E-04	2.55E-03
	f_{10}	5.10E-32	1.41E-07	5.66E-01	3.85E-01	9.59E-01	9.17E-16	2.69E-15
30 维	f_{13}	1.15E-04	5.36E-03	6.04E-03	9.70E-02	3.87E-03	2.40E-02	1.28E-02
	f_1	1.36E-98	5.79E-13	3.77E-01	1.96E-10	2.57E-07	1.68E-27	2.14E-77
	f_4	6.76E-49	1.69E+00	6.57E-01	7.30E-03	3.32E+00	8.48E-07	3.03E+01
	f_7	8.04E-05	5.22E-03	1.05E-02	5.61E-03	6.96E-02	6.61E-04	2.35E-03
	f_{10}	3.10E-32	1.47E-07	5.58E-01	1.10E-06	6.24E-01	1.18E-14	3.25E-15
	f_{13}	6.11E-05	5.36E-03	7.33E-02	1.05E-01	1.41E+01	2.60E-01	2.67E-01

3.2.3 利用 Friedman 检验计算算法的平均排名

为了比较这些算法的性能,图 3 给出了不同算法通过 Friedman 检验的平均排名。由图 3 可以看出,在 2 维的 23 个测试函数(包含固定维多峰)上,FL-AMAS 平均排名第一;在 10 维测试函数上,FL-AMAS 与 HHO 算法的排名相似,其次是 GWO、WOA、DE、SCA、SSA、MFO;在 30 维测试函数上,虽然 FL-AMAS 在这些函数上的平均排名与 GWO 相近,但 FL-AMAS 明显优于 MFO、MVO、SCA、SSA、WOA,总体结果验证了 FL-AMAS 算法的有效性,性能优于大多数对比算法。

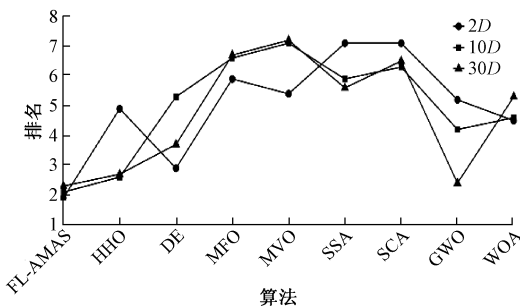


图 3 不同算法在不同维度的排名

Figure 3 Ranking of different algorithmic in different dimensions

3.3 提取最优点的个数对 FL-AMAS 的影响

在 FL-AMAS 算法中,提取的最优点个数是 $NOPC$ 特征中重要的一环;提取的个数太多,不仅增

加计算时间,还会造成本属于两个峰簇的个体合并成一类,导致聚类结果的偏差;提取的个数太少,可能会导致一个函数本来是多峰的,但错误分类为单峰,导致不能选择更合适的算法。为此,将提取的最优点数设置为采样点总数的 0.5%, 1%, 2%, 3%, 4%, 5%, 10%, 以测试最优值点数对 FL-AMAS 的影响。

以 30 维为例,本文在基准测试函数上对采用不同采样点占比的算法进行了实验验证。通过对实验结果中收敛性和算法的计算复杂度进行归一化处理,最终将适应度值和时间复杂度的乘积作为综合评价指标 ($|\text{适应度值}| \times |\text{时间复杂度}|$),实验结果如图 4 所示。可以看出,综合考虑收敛精度与计算代价后,本文在 FL-AMAS 算法中选择的最优采样占比为 2%。

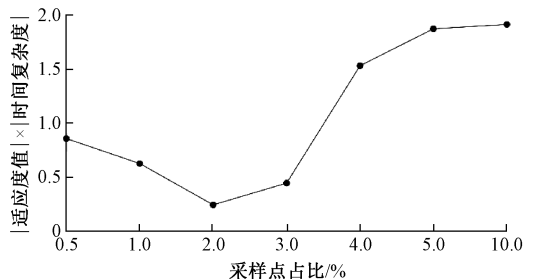


图 4 不同采样数的 $|\text{适应度值}| \times |\text{时间复杂度}|$

Figure 4 Different sample numbers of $|\text{fitness value}| \times |\text{time complexity}|$

3.4 调度方法的扩展性

本文所提出的自适应优化调度算法中所调度算法的选择也具有多样性,可以利用地形特征调度其他具有互补优势的算法。根据多数文献资料,粒子群优化算法(PSO)在解决多峰值问题也有较好的表现,而多元宇宙优化算法(MVO)则具有较好的开发性能。在使用MVO算法和PSO算法作为调度算法来验证自适应算法调度方法的扩展性时,其实验结果与本文实验结果类似:根据NOPC特征,算法调度框架结合了PSO和MVO的优点,同样产生了较好的协调作用。

4 结论

本文提出了一种自适应算法调度方法,利用有效的局部适应度地形特征自适应地为每一个优化问题调度最佳的算法进行求解。本文采用具有高开发性的哈里斯鹰优化算法和具有高探索性的差分进化算法作为调度算法。实验验证结果表明,在不增加计算量的情况下,所提算法吸纳结合每个算法的优点,使得优化结果与单个算法的结果相当或更优。此外,还将PSO和MVO作为调度算法进行了实验,实验结果也验证了所提算法调度的适用性和扩展性。

未来的工作需要在以下方面做深入研究:首先,在特征选择时,如何进一步降低适应度地形特征的计算量;其次,在算法调度后,如何根据提取的地形特征,完成所选择算法中策略或参数的自适应调整,进一步提高FL-AMAS的性能。

参考文献:

[1] 梁静,刘睿,瞿博阳,等. 进化算法在大规模优化问题中的应用综述[J]. 郑州大学学报(工学版), 2018, 39(3): 15-21.
LIANG J, LIU R, QU B Y, et al. A survey of evolutionary algorithms for large scale optimization problem[J]. Journal of Zhengzhou University (Engineering Science), 2018, 39(3): 15-21.

[2] LU P, YE L, ZHAO Y N, et al. Review of meta-heuristic algorithms for wind power prediction: methodologies, applications and challenges[J]. Applied Energy, 2021, 301: 117446.

[3] MALAN K M. A survey of advances in landscape analysis for optimisation[J]. Algorithms, 2021, 14(2): 40.

[4] LIANG J, LI K, YU K J, et al. A novel differential evolution algorithm based on local fitness landscape information for optimization problems[J]. IEICE Transactions on

Information and Systems, 2023, 106(5): 601-616.

[5] ZHENG L M, LUO S Q. Adaptive differential evolution algorithm based on fitness landscape characteristic[J]. Mathematics, 2022, 10(9): 1511.

[6] LI Y X, YU K J, LIANG J, et al. A landscape-aware particle swarm optimization for parameter identification of photovoltaic models[J]. Applied Soft Computing, 2022, 131: 109793.

[7] TANG K, LIU S C, YANG P, et al. Few-shots parallel algorithm portfolio construction via co-evolution[J]. IEEE Transactions on Evolutionary Computation, 2021, 25(3): 595-607.

[8] YUEN S Y, CHOW C K, ZHANG X, et al. Which algorithm should I choose: an evolutionary algorithm portfolio approach[J]. Applied Soft Computing, 2016, 40: 654-673.

[9] 丁青锋,尹晓宇. 差分进化算法综述[J]. 智能系统学报, 2017, 12(4): 431-442.
DING Q F, YIN X Y. Research survey of differential evolution algorithms[J]. CAAI Transactions on Intelligent Systems, 2017, 12(4): 431-442.

[10] HEIDARI A A, MIRJALILI S, FARIS H, et al. Harris hawks optimization: algorithm and applications[J]. Future Generation Computer Systems, 2019, 97: 849-872.

[11] 汪慎文,丁立新,张文生,等. 差分进化算法研究进展[J]. 武汉大学学报(理学版), 2014, 60(4): 283-292.
WANG S W, DING L X, ZHANG W S, et al. Survey of differential evolution[J]. Journal of Wuhan University (Natural Science Edition), 2014, 60(4): 283-292.

[12] WRIGHT S. The roles of mutation, inbreeding, crossbreeding, and selection in evolution[J]. Proceedings of the Sixth International Congress on Genetics, 1932, 1: 356-366.

[13] 王艳丽,梁静,薛冰,等. 基于进化计算的特征选择方法研究概述[J]. 郑州大学学报(工学版), 2020, 41(1): 49-57.
WANG Y L, LIANG J, XUE B, et al. Research on evolutionary computation for feature selection[J]. Journal of Zhengzhou University (Engineering Science), 2020, 41(1): 49-57.

[14] 李亚欣,梁静,岳彩通,等. 基于适应度地形分析的进化计算研究综述[J]. 陕西师范大学学报(自然科学版), 2021, 49(5): 39-53.
LI Y X, LIANG J, YUE C T, et al. A survey of evolutionary computing based on fitness landscape analysis[J]. Journal of Shaanxi Normal University (Natural Science Edition), 2021, 49(5): 39-53.

[15] LI Y X, LIANG J, YU K J, et al. Keenness for charac-

- terizing continuous optimization problems and predicting differential evolution algorithm performance[J]. *Complex & Intelligent Systems*, 2023, 9(5): 5251-5266.
- [16] 廖雄鹰,李俊,罗阳坤,基于自适应变异算子的差分进化算法[J]. *计算机工程与应用*, 2018, 54(06): 128-134.
- LIAO X Y, LI J, LUO Y K, Differential evolution algorithm based on adaptive mutation operator[J]. *Computer Engineering and Applications*, 2018, 54(6): 128-134.
- [17] LANG R D, ENGELBRECHT A P. Decision space coverage of random walks[C]//2020 IEEE Congress on Evolutionary Computation (CEC). Piscataway: IEEE, 2020: 1-8.
- [18] ZUO Y Y, HU Z Q, YUAN S J, et al. Identification of convective and stratiform clouds based on the improved DBSCAN clustering algorithm[J]. *Advances in Atmospheric Sciences*, 2022, 39(12): 2203-2212.
- [19] MEIDANI K, MIRJALILI S, BARATI FARIMANI A. Online metaheuristic algorithm selection[J]. *Expert Systems with Applications*, 2022, 201: 117058.
- [20] MIRJALILI S, LEWIS A. The whale optimization algorithm[J]. *Advances in Engineering Software*, 2016, 95: 51-67.
- [21] MIRJALILI S, MIRJALILI S M, LEWIS A. Grey wolf optimizer[J]. *Advances in Engineering Software*, 2014, 69: 46-61.
- [22] MIRJALILI S. Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm[J]. *Knowledge-Based Systems*, 2015, 89: 228-249.
- [23] MIRJALILI S, MIRJALILI S M, HATAMLOU A. Multi-Verse Optimizer: a nature-inspired algorithm for global optimization[J]. *Neural Computing and Applications*, 2016, 27(2): 495-513.
- [24] MIRJALILI S, GANDOMI A H, MIRJALILI S Z, et al. Salp Swarm Algorithm: a bio-inspired optimizer for engineering design problems[J]. *Advances in Engineering Software*, 2017, 114: 163-191.
- [25] MIRJALILI S. SCA: a Sine Cosine Algorithm for solving optimization problems[J]. *Knowledge-Based Systems*, 2016, 96: 120-133.
- [26] ALCALÁ-FDEZ J, SÁNCHEZ L, GARCÍA S, et al. KEEL: a software tool to assess evolutionary algorithms for data mining problems[J]. *Soft Computing*, 2009, 13(3): 307-318.

Optimization Algorithm Scheduling Method Based on Fitness Landscape Analysis

ZHU Xiaodong, REN Chunxiao, LIU Xiaolan, CHEN Ke, YU Chunming

(School of Electrical and Information Engineering, Zhengzhou University, Zhengzhou 450001, China)

Abstract: Optimization algorithms often perform optimally on specific types of fitness terrains due to the varying nature of optimization problems. To address this limitation, in this study an optimization algorithm scheduling method grounded in fitness terrain analysis was introduced. This method characterizes the terrain features of an optimization problem by extracting the local peak cluster number features of the optimization objective function. Based on these terrain features, the method selected the most suitable algorithm to maximize the advantages of different algorithms through effective scheduling. In particular, this study considered the balance between exploration and exploitation in optimization problems by selecting the harris hawks optimization algorithm (HHO), known for its high development capability, and the differential evolution algorithm (DE), recognized for its strong exploration ability, as the scheduling algorithms. The choice of algorithm was tailored to the specific adaptability characteristics of the terrain. Experimental results show that the convergence performance of FL-AMAS was improved by 75% compared with that of HHO alone, and by 40% compared with that of DE algorithm. Further, FL-AMAS was compared with six advanced algorithms, and FL-AMAS outperformed these algorithms in convergence accuracy on 75% of the benchmark set. The effectiveness and scalability of the proposed scheduling method were further validated through comparisons with other types of scheduling optimization algorithms.

Keywords: optimization algorithm scheduling; fitness landscape analysis; feature extraction; local peak points; harris hawks optimization algorithm; differential evolution algorithm