基于方形邻域和随机数组的哈里斯鹰优化算法

刘小龙, 梁彤缨

基于方形邻域和随机数组的哈里斯鹰优化算法

刘小龙, 梁彤缨†

(华南理工大学 工商管理学院，广州 510640)

摘 要: 针对哈里斯鹰优化算法探索能力和开发能力不平衡的问题,通过设置一种多子群方形邻域拓扑结构引导各子群内的个体可以纵横双向随机觅食. 为了避免局部最优,通过设置固定置换概率,加强各个子群个体的信息交流,使子群内个体依照随机数组与其他子群的相应个体进行置换. 在子群内部,基于历史进化信息进行HHO算法中的算子选择,以更好地利用现有问题领域的信息. 利用可变维度基准函数与各种智能优化算法及其改进方法进行跨文献对比,结果表明改进方法在收敛精度、寻优能力上明显高于原始算法和对比文献,且具有较好的鲁棒性,适合推广至实际的优化问题之中.

关键词: 哈里斯鹰优化算法；多子群结构；方形邻域；随机数组；跨种群信息交换；全局最优

中图分类号: TP301.6　　**文献标志码:** A

DOI: 10.13195/j.kzyjc.2021.0478

引用格式: 刘小龙,梁彤缨. 基于方形邻域和随机数组的哈里斯鹰优化算法[J]. 控制与决策, 2022, 37(10): 2467-2476.

Harris hawk optimization algorithm based on square neighborhood and random array

LIU Xiao-long, LIANG Tong-ying†

(School of Business Administration，South China University of Technology，Guangzhou 510640，China)

Abstract: Aiming at the problem of the imbalance between the exploration and development capabilities of the Harris hawk optimization algorithm (HHO), a multi-subgroup square neighborhood topology is set up to guide individuals in each subgroup to forage randomly in both directions. In order to avoid local optima, a fixed replacement probability is set to strengthen the information exchange of each subgroup individual, so that the individuals in the subgroup can be replaced with the corresponding individuals of other subgroups according to the random array. Within the subgroup, the operator selection in the HHO algorithm is performed based on historical evolution information to make better use of the information in the existing problem domain. Cross-document comparisons are made using variable-dimensional benchmark functions with various intelligent optimization algorithms and their improved methods. The results show that the improved method is significantly higher than the original algorithm and comparative literature in terms of convergence accuracy and optimization capability, and has better robustness, which is suitable for generalization to actual optimization problems.

Keywords: Harris hawks optimization；multi-subgroup structure；square neighborhood；random array；information exchange in cross-group；global optimal

0 引 言

20世纪50年代以来,源于生物或自然界系统灵感的“元启发式”方法得到了极大应用,这类方法主要针对不连续、多峰值、非线性、不可微等特征的实际工程问题,无法像传统方法那样求得精确解,近似解是解决这类问题的较好方案[1].“元启发式”一词最早是由Fred Glover提出的,良好的“元启发式”应平衡两个主要特征:全局探索和局部开发. 探索性可以保留解的多样性并找到最有可能的解区域,否则算法会过早收敛到局部极值. 开发性使算法可以深入挖掘在探索阶段发现的有价值解区域,否则可能导致算法不收敛[2]. 因此,算法要达到全局最优,就必须完美地平衡这两种能力.“元启发式”优化算法在智能计算中越来越流行,并已经广泛应用于大量的实际工程问题.

哈里斯鹰优化(Harris hawks optimization, HHO)

收稿日期: 2021-03-22；**录用日期:** 2021-07-05.
基金项目: 国家社会科学基金面上项目(19BJL068).
责任编委: 梁樑.
†通讯作者. E-mail: bmtliang@scut.edu.cn.

是Heidari等[3]于2019年提出的基于群体的算法之一,提出伊始便得到了极大关注,在图像识别[4]、神经网络[5]、约束工程优化[6]、支持向量机优化[7]、三杆桁架设计[8]、供水网络的优化[9]、光伏系统建模[10]和降雨径流建模[11]等工程应用领域得到广泛应用. HHO算法的灵感来自于目前最聪明的鸟类——哈里斯鹰(栗翅鹰),通过对它狩猎兔子时的行为模拟,开发出相应的随机数学模型. HHO模型公式源于灰狼优化算法(grey wolf optimizer, GWO)[12],但融入了有关逃逸、群体中心、随机种群划分、布谷鸟搜索(CS)[13]等相关理念. HHO针对大多数连续非约束问题都具有较好的求解能力,与现有3类算法如遗传算法(genetic algorithm, GA)和差分进化(differential evolution, DE)、粒子群算法(particle swarm optimization, PSO)和布谷鸟搜索(cuckoo search, CS)、教学优化(teaching-learning-based optimization, TLBO)和花朵授粉算法(flower pollination algorithm, FPA)等相比,具有较为出色的寻优性能,但探索能力和开发能力的平衡仍是无法规避的问题之一. 文献[3]数据表明, HHO针对Rosenbrock函数的寻优精度不高(e-02等级);针对Schwefel函数的鲁棒性较差(e+02等级),难以获得全局极值;针对Quartic、Penalized1、Penalized2等函数的收敛精度(e-04、e-06、e-04等级)还有很大的提升空间.

面对HHO表现出的上述问题,现有文献进行了相关改进研究. 文献[4]指出,当HHO的逃逸能量为零时,哈里斯鹰的猎物能量完全耗尽,此时算法探索能力受限,该阶段的随机行为毫无效果,据此提出了差分进化的自适应改进算法(DEAHHO),数值实验表明对低维度问题有一定改善,但对于高维度改善不明显,且Rosenbrock和Schwefel函数的寻优精度和鲁棒性不平衡问题依然存在, Step函数的寻优精度改善不明显. 文献[6]发现,针对工程约束优化问题, HHO非常容易陷入局部最优,因此将正弦余弦算法(SCA)与HHO算法相融合,针对CEC2017、CEC2018和11个多学科工程设计优化问题进行了实验验证,发现融合方法比各自的标准算法要好得多. 数值实验表明,该方法相对于文献[4]有一定优势, Rosenbrock函数有一定改善,但这是通过融合方法增加迭代次数作为代价换来的. 文献[8]认为,基本HHO算法在协同搜索时不能进行充分的信息共享,导致算法探索和开发难以平衡,进而产生过早收敛和陷入局部最优的境况. 为此,文献[8]设计具有混沌干扰的非线性逃逸能量因子来进行能力平衡,同时提出一种基于信息交换的改进方法. 数值实验表明,该算法在收敛速度、求解精度和鲁棒性方面均有一定提高. 但该方法仅针对Sphere、Rosenbrock、Schwefel和Kowalik四个30维的基准函数进行了比较研究,具有较高的寻优精度和鲁棒性,而面向1 000维问题时其寻优性能下降明显, Sphere、Rosenbrock和Schwefel函数的精度和鲁棒性不均衡问题明显. 文献[14]针对HHO陷入局部极值的可能性,提出将混沌初始化机制、多种群策略、差分进化思想进行融合,以增强算法的全局探索和局部开发能力,在CEC2017测试函数上的实验结果表明,该方法相对遗传算法(GA)、粒子群算法(PSO)等具有较大的优势. 针对探索和开发的不平衡问题,文献[15]提出了非线性能量参数、差异化变异、贪婪选择、对立学习等,对于缓解过早收敛的停滞问题有一定的效果. 实验结果表明,针对大多数函数优化问题,该方法在精度和鲁棒性方面都有很大提升,但针对Schwefel、Penalized1、Penalized2等函数的求解精度和鲁棒性反而下降. 文献[16]提出了一种能量周期性递减与牛顿局部增强的改进算法,但其能量周期性递减是相当于减少了探索次数和增加了开发迭代次数,这样会增强部分基准函数的寻优精度,且文献[16]的对比函数仅采用4种函数,其精度改进虽然较为明显,但是相应改进举措都是针对开发而非探索能力. 另外,文献[16]的6次牛顿局部迭代增强措施,变相增加了迭代次数和提高了计算代价.

除HHO外,不同学者针对典型群智能算法如灰狼优化、差分进化、和声搜索、鲸鱼优化等也进行了诸多研究. 顾清华等[17]提出一种求解高维复杂函数的遗传-灰狼混合算法(HGGWA),与粒子群等3种基本算法以及9种灰狼优化改进算法的比较表明,该算法在收敛精度方面得到极大改进. 吴文海等[18]提出了基于随机邻域策略和广义反向学习的自适应差分进化算法(GOBL-RNADE),与各种改进差分进化数值实验对比发现,该方法也大大改善了差分进化算法的性能. 刘丽杰等[19]提出一种基于动态行为选择的和声搜索算法(DBSHS),在处理高维函数的过程中展现出比其他改进方法更好的适用性. 肖子雅等[20]提出了一种精英反向黄金正弦鲸鱼算法(EGolden-SWOA),相对于现有其他鲸鱼算法的改进方法,如W-SA-WOA[21]、CWOA[22]和IMWOA[23]等,表现较为优异. 上述典型群智能算法的改进为本文算法性能对比提供了数值基础.

基于上述文献的分析,本文尝试用一种多种群个体交换的拓扑结构改进方法平衡HHO算法的探索和

开发能力,可以在不增加计算复杂度的情况下,有效求得大部分函数的全局最优解. 主要工作如下:

1) 提出一种方形邻域拓扑结构的多子群划分思路,以区别于基本HHO算法的随机子群划分. 将现有种群形成一种方形结构,每一个个体依概率受到来自横向或者纵向方向的最优值影响,纵横双向没有直接的信息互动,形成信息孤岛. 这一策略设置的主要目的是为了规避局部最优,在探索与开发之间取得均衡,进而获得问题的全局极值.

2) 建立一种子群之间的信息沟通交流机制,算法迭代从纵向子群展开,对子群所有个体,依固定概率与其他子群的随机个体进行互换,以完成信息交流. 具体为在给定概率条件(如rand > 0.3)下,纵向子群选定的个体与其他子群的随机个体进行个体置换. 设置这一策略的目的是促进不同子群的信息流动,避免算法陷入局部最优.

3) 基于历史进化信息进行HHO算法中的算子选择,以区别现有基本HHO算法中基于随机概率$P>0.5$来随机划分两个子群的思路,从而更好地利用现有问题的信息. 设置这一策略的原因是规避基本HHO算法中,同一策略行为失效后的再次使用,以强化对基本HHO算法策略的引导,即策略有效则采用硬包围和软包围策略,否则采用快速俯冲式的两种策略行为. 具体为种群个体存储历史进化信息,如果个体历史进化的适应度函数值变优,则按照软硬包围的策略1和策略2进行迭代;如果历史进化的适应度函数值变差,则按照快速俯冲式觅食方式的策略3和策略4进行迭代.

通过选取文献[3, 4, 6, 8, 12, 14]的基准函数进行跨文献比较研究发现,本文算法在大部分测试函数上获得了全局极值,在Rosenbrock、Schwefel函数的探索能力和Penalized1、Penalized2等函数的寻优精度上有显著提高,且保持着较好的寻优鲁棒性.

1　哈里斯鹰优化算法

哈里斯鹰(Parabuteo unicinctus)是美国亚利桑那州南部等地区的一种著名猛禽,它们通过群体行为的协同进行觅食,哈里斯鹰捕获猎物的主要策略是"突袭",即几只鹰尝试从不同方向对猎物如掩体外的野兔进行协同攻击. 这种突击虽然可以迅速完成,但也有逃逸行为发生. "突袭"行为也包括哈里斯鹰多次、短时、迅速地在猎物附近的行为隐藏,哈里斯鹰的捕食风格取决于当时的环境情况和猎物的逃逸形式. 通过团队协作,哈里斯鹰可以困惑逃逸的猎物,使它们无法恢复防御能力,进而高效地捕获疲倦的野兔. Heidari等[3]基于上述行为和灰狼优化(GWO)的算子结构,提出了HHO模型方法. 即将哈里斯鹰的捕猎分为探索和开发两种行为,通过逃跑能量$|E|\geqslant 1$判断算法是否执行探索行为,通过$|E|\geqslant 0.5$来判断是处于软包围还是硬包围,通过包围中的随机数$\geqslant 0.5$判断是否执行快速俯冲行为. 其中,逃逸行为借助猎物的跳跃系数J度量,当逃逸失效,还允许发生LF(D)的随机游走行为.

1.1　探索行为

当$|E|\geqslant 1$时,哈里斯鹰随机栖息在搜索空间中,并伺机观察猎物. 在等概率0.5的栖息策略下,哈里斯鹰根据随机猎物位置和猎物位置与群体中心位置的向量差确定栖息位置,有

$$X(t+1)=\begin{cases}X_{\rm rand}(t)-r_1|X_{\rm rand}(t)-2r_2X(t)|,\\ \quad q\geqslant 0.5;\\ (X_{\rm rabbit}(t)-X_m(t))-r_3({\rm lb}+r_4({\rm ub}-{\rm lb})),\\ \quad q<0.5.\end{cases}\tag{1}$$

其中: $X(t+1)$和$X(t)$分别为第$t+1$次和第t次迭代时哈里斯鹰的位置, $X_{\rm rand}(t)$为第t次迭代时随机哈里斯鹰的位置, r和q为[0,1]区间内均匀分布的随机数, $X_{\rm rabbit}(t)$和$X_m(t)$为猎物兔子位置(当前迭代最优值)和哈里斯鹰群中心位置, lb和ub为取值范围的上下界. 式(1)和(2)描述了当猎物逃逸能量$|E|>1$时HHO算法的探索行为,其中E随迭代递减,有

$$E=2E_0(1-t/T).\tag{2}$$

其中: T为最大迭代次数; t为当前迭代步数; E_0为初始化能量值,取$[-1,1]$间均匀分布的随机数.

1.2　开发行为

当$|E|<1$时,表明哈里斯鹰发现了猎物兔子的方向, HHO算法执行开发行为. 依哈里斯鹰的"突袭"狩猎行为,按照能量绝对值和随机数分别与0.5比较判断,可以分为4种策略.

策略1　软包围. 当$|E|\geqslant 0.5$且随机数$\geqslant 0.5$时,哈里斯鹰执行软包围策略,有

$$X(t+1)=\\ X_{\rm rabbit}(t)-X(t)-E|J\times X_{\rm rabbit}(t)-X(t)|,\tag{3}$$

其中J为猎物兔子的跳跃能力,取$[0,2]$区间内均匀分布的随机数.

策略2　硬包围. 当$|E|<0.5$且随机数$\geqslant 0.5$时,哈里斯鹰启用硬包围策略,有

$$X(t+1)=X_{\rm rabbit}(t)-E|X_{\rm rabbit}(t)-X(t)|.\tag{4}$$

式(4)中,因猎物长期逃跑致使逃逸能力较低,无法完成跳跃J,只能待在原地等待被捕捉.

策略3 快速俯冲式软包围. 当$|E| \geqslant 0.5$且随机数< 0.5时. 因猎物仍具有较大的跳跃能力J,哈里斯鹰启用快速俯冲突袭,如果突袭失效(Y适应度没有改善),则执行随机游走Z,如果游走失效则退回原地,策略模式为

$$Y = X_{\text{rabbit}}(t) - E|J \times X_{\text{rabbit}}(t) - X(t)|, \tag{5}$$

$$Z = Y + S \times \text{LF}(D). \tag{6}$$

其中: S为随机向量, D为问题维度, LF为按文献[13]定义的Levy飞行函数.

策略4 快速俯冲式硬包围. 当$|E| < 0.5$且随机数< 0.5时,猎物兔子没有足够的逃逸能力,哈里斯鹰基于猎物位置和群体中心位置进行围捕,以减小种群与猎物间距离差,策略模式为

$$Y = X_{\text{rabbit}}(t) - E|J \times X_{\text{rabbit}}(t) - X_m(t)|. \tag{7}$$

如果突袭失效(Y适应度没有改善),则执行随机游走Z,如果游走失效则退回原地.

2 改进哈里斯鹰优化算法

2.1 哈里斯鹰优化算法的问题分析

在HHO算法中,主要依据能量值$|E| \geqslant 1$来判断算法是执行探索还是开发行为. 但根据式(3), $|E|$值虽然大概率线性递减,但与E_0值密切相关,如果$E_0 < 0.5$,则$|E|$将一致小于1,因此HHO算法在迭代不同时期的探索行为和开发行为的启动概率不同.

就探索行为而言,在迭代$t = 1$时刻,探索概率约为50 %,在$t = T/2$时刻,概率为0 %. 因此,探索概率在前述$[0, T/2]$时段是线性递减的. 可见, HHO算法的总体探索概率均值为25 %,探索执行总次数是总体迭代次数的12.5 %. 在开发行为的4种策略中, HHO算法采用随机数与0.5比较来判断是否执行快速俯冲式包围策略以及进行随机游走模式. 这里的快速俯冲式包围策略和随机游走还采用了一种贪婪比较选择,即俯冲式包围策略如果有效则保留新解,否则执行随机游走. 另外,还要判断随机游走是否有效,如果有效则保留新解,否则退回初始解. 即式(6)和(8)执行后的效果必须变好,否则执行式(7);同样式(7)执行后的效果也必须变好,否则退回原解. 为评估这种贪婪行为的效率,利用文献[3]中$F_1 \sim F_{13}$基准函数,测试策略3和策略4的策略效率问题(维度$D = 30$,最大评估次数$\text{fEvals} = 15\,000$),发现策略3的执行有效率在不同的基准函数中有差异,其针对13种测试函数的平均有效率为56.08 %;策略失效后执行一种Levy的行为,但其平均有效率仅为0.45 %. 同时发现,策略4的平均有效率为30.06 %,策略失效后的Levy行为有效率仅为0.03 %. 可见, HHO算法中的Levy策略效率相对较低.

2.2 基于方形邻域的多子群拓扑结构

HHO算法在能量值$|E| < 1$的开发行为中,利用随机数与0.5比较区分为两个随机种群,这一机制对于基准函数如Sphere、Schwefel2.22、Rastrigin等具有一定效果,但对于其他基准函数如Rosenbrock、Schwefel等,并不能获得全局最优,也没有很好地利用种群进化的历史信息. 前述策略效率分析发现,基于贪婪行为的策略3和策略4,式(6)和(8)的有效率为56.08 %和30.06 %,而式(7)的Levy飞行有效性为0.45 %和0.03 %. 因此,针对开发行为提出一种方形邻域的多种群拓扑结构来平衡算法的开发能力和探索能力,具体如下:改进算法将整体哈里斯鹰种群区分为K个子群,每个子群成员固定也为K个,这样便形成了纵向子群和横向子群的双向结构. 以纵向子群作为主体子群,横向子群作为备用子群,纵向子群间单独觅食,相互没有信息联系,形成信息孤岛. 其中,纵向子群的最优值和横向子群的最优值各有1/2机会成为子群猎物位置$X_{\text{rabbit}}(t)$. 即纵向子群成员执行开发行为的位置更新时,子群迭代按照纵向方向依次进行,但子群猎物位置却有可能是个体所在列的纵向最优或者是行的横向最优,具体迭代式为策略1和策略2中的式(4)和(5). 之所以如此设置,主要是考虑两个方面:一是为规避子群最优个体的猎物可能就是自己的情况;二是在算法收敛的后期能够跳出局部极值.

2.3 基于随机数组的个体互换机制

本文基于方形邻域纵横双向随机选择最优值的迭代方法中,虽然个体可以向两个方向随机学习,但是子群之间没有信息交互. 为了获取全局极值,需要推动不同子群间的信息互动. 现有文献的信息交流都是通过子群个体的加权来实现,本文提出一种固定概率的个体互换机制. 假设固定概率P为0.3,对每个纵向子群的个体进行轮询判断,如果$\text{rand} < 0.3$,则针对K行K列矩阵采用随机数组的方式产生两个随机整数确定替换个体所在的行和列. 随机数组的第1个数为行,第2个数为列,要求随机数组的行和列不与现有个体的行或列相同,否则重新产生,即置换的两个点所在的行列延长线内部形成一个矩阵. 譬如针对M_{53}进行个体置换,产生随机数组$[3, k] (k > 5)$,由于M_{53}和M_{3k}所在的行和列都不同,且其纵横双向

M_{33}、M_{53}、M_{5k}、M_{3k}构成了一个矩阵,所以可以进行个体置换. 如果不能构成矩阵,则再次产生随机数组.

由上述机制可以看出,这种信息交流方法的本质是个体互换,而非对子群个体进行位置迁移,相当于扩大部分子群的多样性以提高探索能力,缩小部分子群的个体差距以增加收敛精度. 数值实验表明,上述设置对于很多基准函数都很快获得了全局最优.

2.4 基于历史信息的策略算子选择

HHO算法中,能量值E按照均匀分布从±2收缩到0,即迭代$t = 0$时刻,能量$|E| \geqslant 1$的概率为50%; 当迭代$t \geqslant T/2$时,能量$|E| \geqslant 1$的概率为0. 在前$T/2$次迭代中,探索和开发行为是同时执行的,没有先探索后开发的截然区分. 探索行为是为了规避局部极值,开发行为中的游走行为是为了跳出局部极值,是一种针对现有策略行为失效后的选择.

为了更有效地利用现有的4种开发策略行为,提出一种基于历史信息的策略算子选择方式,即针对所有个体存储历史进化的适应度信息. 如果个体的适应度值改善,则进一步开发,采用HHO算法中的策略1和策略2; 如果个体的适应度信息变差,则相当于需要跳出局部极值,采用HHO算法中的策略3和策略4. 这一改变相当于将开发行为中的$|E|$值判断和随机数与0.5判断进行了更改. 原算法根据随机数与0.5判断随机划分了两个子群,分别执行策略1、策略2的软硬包围和策略3、策略4的快速俯冲式软硬包围; 而本文方法依据适应值的改善情况决定是执行软硬包围还是快速俯冲式软硬包围. 设置这一策略的目的是为了规避同一策略行为失效后的再次使用行为,以加强对HHO算法中的不同策略进行引导.

2.5 计算复杂度分析

标准HHO方法的计算复杂度主要取决于3个过程: 初始化、适应性评估和哈里斯鹰的位置更新. 假设种群数为N,则初始化过程的计算复杂度为$O(N)$. 由于要寻找最佳位置并更新所有的位置矢量,种群更新的计算复杂度为$O(T \times N) + O(T \times N \times D)$. 其中: T为最大迭代次数,D为问题维度. 因此,HHO的计算复杂度为$O(N \times (T + TD + 1))$. 本文改进HHO算法,仅更改算法的逻辑拓扑结构,增加了子群间的信息交换活动,但并不调用适应度计算. 可见,改进算法的计算复杂度与标准HHO算法一样,也是$O(N \times (T + TD + 1))$.

2.6 改进算法的逻辑流程

改进算法的实现流程描述如下.

step 1: 参数初始化. 定义问题维度D、子群数K、子群个体数k,初始能量值E,最大迭代步数T.

step 2: 初始化群体位置,进行多子群划分,计算各子群适应度,标注纵横双向最佳适应度,储存个体上次进化的历史信息.

step 3: 迭代开始,能量值和历史信息判断.

step 4: 如果$|E| \geqslant 1$,则哈里斯鹰随机选择式(1)或(2)开始探索行为. 如果$|E| < 1$,则哈里斯鹰根据能量值和历史适应度改善情况,选择开发行为策略.

策略5 软包围. 当$|E| \geqslant 0.5$且历史适应度改善时,子群个体采用式(3)基于纵或横子群猎物更新位置.

策略6 硬包围. 当$|E| < 0.5$且历史适应度改善时,子群个体采用式(4)基于纵或横子群猎物更新位置.

策略7 快速俯冲式软包围. $|E| \geqslant 0.5$且历史适应度无改善时,子群个体采用式(6)基于纵或横子群猎物更新位置.

策略8 快速俯冲式硬包围. $|E| < 0.5$且历史适应度无改善时,子群个体采用式(7)基于纵或横子群猎物更新位置.

step 5: 纵向子群个体的信息交换. 对纵向子群的个体进行概率P判断,通过随机数组选定行列不交叉的其他子群个体进行置换.

step 6: 更新能量值、子群位置和适应度等.

step 7: 判断循环结束条件,满足则结束,输出最优适应度和位置,不满足则转至step 3.

3 数值实验

参考文献[3, 4, 6, 8, 15]的做法,利用基准函数进行对比,测试函数包括单峰(UM)和多峰(MM)问题,内有大量局部极小点,在不同文献中也得到了推广和经常性应用. 与现有HHO改进算法进行比较研究,这些被比较算法的参数设置为原始文献的建议值. 所有算法均在Matlab 2016a上实现,使用Windows 7 64位专业版和64 GB计算机内存. 改进算法的子群数K为7,子群内个体数也为7,低维度函数的最大评估次数fEvals设置为15 000,子群内个体间交换固定概率为0.3,能量值按照文献[3]设置依概率从2直线下降为0,记录30次独立运行的最优化结果. 参照上述文献,选取平均值、标准差、最优值、成功次数和成功比率等指标衡量相关算法性能.

选取文献[3]中13个可变维度测试函数进行对比研究,其中$F_1 \sim F_7$为单峰函数,$F_8 \sim F_{13}$为多峰函数. 单峰函数$F_1 \sim F_4$主要测试算法的寻优精度,$F_5 \sim F_7$拥有较多的局部极值点,其局部极值较难

跳出. 多峰函数 F_8、F_{12} 和 F_{13} 拥有较多的局部极值点,许多算法很难跳出局部极值.

3.1 改进算法的参数性能影响分析

为便于比较,设置测试函数的维度 $D = 30$,算法最大评估次数 $\mathrm{fEvals} = 15\,000$,其他参数同HHO. 将所提出的基于方形邻域和随机数组的哈里斯鹰优化算法命名为MHHO, MHHO的整体种群数量由子群数 K 决定 $(K \times K)$,开发行为的子群划分采用历史信息而不是随机数0.5. 针对子群内个体的信息交换,采用固定概率 P 的方法进行置换,为了规避局部极值采用一种纵横双向子群最优的随机猎物选择办法.

由于上述参数性能对算法有一定影响,需要进一步实验分析. 考虑到HHO算法中种群数为30,故本文设置子群数 $K = 6$,种群内个体数 $k = 36$,子群个体的被置换概率 P 设置为0.3,采用基于历史进化信息评估的开发行为选择策略,算法的最大评估次数fEvals即函数的调用次数与文献[3]相同取15 000,算法测试30次,利用均值指标进行算法性能的比较.

首先,进行置换概率 P 实验,分别实验 $P = 0$、0.3、0.5和1四种情况,记为实验1 $\sim$ 实验4,在其他参数不变的情况下,算法针对基准函数的测试精度等级如表1所示. 由表1可见,不同的置换概率 P 对测试函数的影响不同,$P = 0.5 \sim 1$ 时的效果相对较好,此时 F_5、F_6、$F_8 \sim F_{13}$ 全部获得全局极值,$F_1 \sim F_4$ 的求解精度也相对较高,F_7 函数最优值是接近0的随机值,因此不予比较. 但总体而言,需要置换概率 $P > 0$,在此设置置换概率 $P = 0.8$.

其次,对不同的子群数在 $P = 0.8$ 下进行实验,设置 $K = 4$、8、10三种情况进行分析,记为实验5 $\sim$ 实验7,其他参数不变,算法针对基准函数的测试精度等级如表2所示. 由表2可见,不同的子群数目对测试函数的影响不同,子群数越少,$F_1 \sim F_4$ 的求解精度越高,但子群数过少如 $K = 4$,导致 F_5、F_6 和 F_8 的部分维度陷入局部极值,求解精度偏低. 因此,子群数目平衡着探索和开发能力的平衡,相对较大的子群数有利于平衡该能力,本文取子群数 $K = 6$.

最后,对能量 E 值的取值范围限制进行实验,前述7次实验的初始能量为2,现设置为1.5和1,即分别从1.5和1线性下降至0,分别记为实验8和实验9. 设置 $K = 6$, $P = 0.8$,其他参数不变,算法针对基准函数的测试精度等级如表3所示.

表1 置换概率 P 对算法性能影响的均值精度 ($D = 30$)

实验	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}
1	e-100	e-050	e-090	e-051	e-004	e-004	e-004	−12 568	0	e-016	0	e-005	e-004
2	e-193	e-107	e-194	e-096	e-005	0	e-004	−12 569	0	e-016	0	e-032	e-032
3	e-202	e-093	e-187	e-105	e-005	0	e-004	−12 569	0	e-016	0	e-032	e-032
4	e-181	e-103	e-207	e-104	0	0	e-004	−12 569	0	e-016	0	e-032	e-032

表2 子群数目 K 对算法性能影响的均值精度 ($D = 30$)

实验	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}
5	e-257	e-136	e-273	e-138	e-003	e-032	e-004	−12 568	0	e-016	0	e-032	e-032
6	e-146	e-070	e-141	e-071	0	0	e-004	−12 569	0	e-016	0	e-032	e-032
7	e-109	e-057	e-107	e-057	0	0	e-004	−12 569	0	e-016	0	e-032	e-032

表3 初始能量值对算法性能影响的均值精度 ($D = 30$)

实验	F_1	F_2	F_3	F_4	F_5	F_6	F_7	F_8	F_9	F_{10}	F_{11}	F_{12}	F_{13}
8	e-124	e-064	e-122	e-065	0.95	0	e-004	−12 569	0	e-016	0	e-032	e-032
9	e-155	e-076	e-163	e-085	0	0	e-004	−12 569	0	e-016	0	e-032	e-032

对比实验9、实验8和实验4发现,较低的初始能量 E 值可以相对提高 $F_1 \sim F_4$ 的求解精度. 但总体而言,初始能量值 E、置换概率 P 对算法的性能影响有限,改进算法针对不同参数值具有较强的鲁棒性且能够获得大多数测试函数的全局极值.

3.2 本文MHHO方法与其他HHO改进方法的对比

为了对比MHHO与其他改进HHO方法的优劣,利用文献[3]中13个可变维度基准函数进行实验研究,并与文献[3,4,6,8,12,13]的数值结果进行对比. 文献[4,6,8]对文献[3]的HHO算法进行了有效改进,HHO算法采用了布谷鸟搜索(CS)和灰狼优化(GWO),因此对比上述算法. 函数的问题维度 $D = 30$,算法的最大评估次数 $\mathrm{fEvals} = 15\,000$,统计30次独立运行的算法均值和标准差指标,计算测试数据的Friedman排名,实验结果如表4和表5所示.

表 4　改进HHO算法在基准函数上的均值指标($D=30$)

函数	MHHO	IEHHO[8]	HHOSCA[6]	DEAHHO[4]	HHO[3]	GWO[12]	CS[13]
F_1	1.387 8e-195	1.38e-143	1.86e-91	3.36e-211	3.9e-97	1.18e-27	9.06e-04
F_2	1.673 4e-106	–	2.46e-51	6.90e-106	1.5e-51	9.71e-17	1.49e-01
F_3	8.152 4e-182	–	8.88e-72	6.46e-139	1.9e-63	5.12e-05	2.10e-01
F_4	5.718 5e-102	–	8.01e-49	8.04e-103	1.0e-47	1.24e-06	9.65e-02
F_5	0	1.44e-05	1.43e-02	3.66e-02	1.3e-02	2.70e+01	2.76e+01
F_6	0	–	2.23e-04	5.63e-04	1.1e-04	8.44e-01	3.13e-03
F_7	1.211e-04	–	1.22e-04	1.36e-04	1.4e-04	1.70e-03	7.29e-02
F_8	−12 569.486 6	−12 569.5	−12 569.08	12 547	−1.25e+04	−5.97e+03	−5.19e+19
F_9	0	–	0	0	0	2.19e+00	1.51e+01
F_{10}	8.881 8e-016	–	8.881 7e-16	8.88e-16	8.88e-16	1.03e-13	3.29e-02
F_{11}	0	–	0	0	0	4.76e-03	4.29e-05
F_{12}	1.570e-032	–	1.13e-05	2.89e-05	7.35e-06	4.83e-02	5.57e-05
F_{13}	1.349e-032	–	1.13e-04	3.30e-04	1.57e-04	5.96e-01	8.19e-03
friedman	1.154	2.333	2.385	1.769	2.615	5.538	5.692

表 5　改进HHO算法在基准函数上的标准差指标($D=30$)

函数	MHHO	IEHHO[8]	HHOSCA[6]	DEAHHO[4]	HHO[3]	GWO[12]	CS[13]
F_1	0	6.44e-143	9.49e-91	0	1.7e-96	1.47e-27	4.55e-04
F_2	9.160 7e-106	–	1.11e-50	4.86e-106	6.9e-51	5.60e-17	2.79e-02
F_3	0	–	4.86e-71	3.81e-138	1.0e-62	2.03e-04	5.69e-02
F_4	2.408 8e-101	–	2.82e-48	5.68e-102	5.0e-047	1.94e-06	1.94e-02
F_5	0	2.00e-05	2.02e-02	5.25e-02	1.8e-02	7.78e-01	4.51e-01
F_6	0	–	3.38e-04	9.29e-04	1.5e-04	3.18e-01	1.30e-03
F_7	1.069e-04	–	1.10e-04	1.57e-04	1.0e-04	1.06e-03	2.21e-02
F_8	1.850 1e-012	1.93e-04	0.766 9	84.5581	1.47e+02	7.10e+02	1.76e+20
F_9	0	–	0	0	0	3.69e+00	1.25e+00
F_{10}	0	–	0	0	0	1.70e-14	7.93e-03
F_{11}	0	–	0	0	0	8.57e-03	2.00e-05
F_{12}	5.567 4e-048	–	1.50e-05	4.78e-05	1.19e-06	2.12e-02	4.96e-05
F_{13}	5.567 4e-048	–	1.66e-04	4.86e-04	2.15e-04	2.23e-01	6.74e-03

由表4可见, MHHO在13个基准函数中, 除了$F_1 \sim F_4$的精度稍低外(基本处于同一档次, 如果增加迭代次数则本文算法的精度更高), $F_5 \sim F_6$和$F_8 \sim F_{13}$等全部获得了全局极值(F_7最优为接近0的随机值). 可见, MHHO相对3种HHO改进算法以及布谷鸟搜索(CS)、灰狼优化(GWO)等算法具有较为明显的优势, 其Friedman的rank排名最低为1.154, 优于现有HHO改进算法和基本HHO算法. 针对文献所述较难优化的Rosenbrock函数F_5和Schwefwel函数F_8, MHHO完全收敛. 研究发现, MHHO采用了多子群结构导致子群内的个体有限, MHHO对$F_1 \sim F_4$函数不占优势, 但测试发现如果迭代次数增加5倍, MHHO针对这4个函数可以稳定地获得全局极值, 其均值和标准差为0.

由表5可见, MHHO在13个基准函数中, 除F_4的标准差稍大外(与文献[4]处于同一档次), 针对其他测试函数的标准差全部最小且较为稳定.

3.3　MHHO方法与典型群智能优化算法的改进方法对比

为了对比MHHO方法与其他典型群智能算法如灰狼优化、差分进化、和声搜索、鲸鱼优化等改进方法的优劣, 继续利用文献[3]中的13个可变维度函数进行实验研究, 并与文献[17-20]的4种典型改进算法进行对比, 相关数值结果如表6所示. 由表6可见, 针对测试寻优精度的$F_1 \sim F_4$函数, MHHO方法在现有条件下的寻优精度仅高于文献[17]的HGGWA, 低于文献[18-20]的3种算法, 但提高最大评估次数fEvals值发现, MHHO方法可以全部获得精确解. 测试函数F_7的最优解是接近0的随机值, 现有几种优化算法的优化效果基本一样. 针对测试函数$F_9 \sim F_{11}$, MHHO方法在30次测试中均获得了全局极值, 标准差为0, 较为稳定, 故MHHO不劣于对比算法. 针对较难优化的F_5、F_6、F_8、F_{12}和F_{13}五种函数, MHHO方法稳定地获得了全局极值, 标准差是最小的. 综上, 本文方法在低维度函数上不劣于上述改进算法.

表 6 本文MHHO方法与典型群智能优化方法改进算法的性能对比($D = 30$)

函数	HGGWA[17]		GOBL-RNADE[18]		DBSHS[19]		EGolden-SWOA[20]		MHHO	
	均值	标准差	均值	标准差	均值	标准差	均值	标准差	均值	标准差
F_1	9.56e-43	9.46e-118	1.63e-240	0	0	0	0	0	1.387e-195	0
F_2	2.07e-64	3.49e-64	1.80e-126	6.91e-126	1.00e-178	0	6.69e-202	0	1.673e-106	9.160e-106
F_3	2.63e-30	3.88e-30	7.15e-276	0	9.76e-239	0	0	0	8.152e-182	0
F_4	2.54e-35	4.85e-35	6.26e-115	2.50e-114	7.02e-144	1.50e-143	3.58e-191	0	5.718e-102	2.408e-101
F_5	2.52e+01	3.35e-01	5.94e-02	1.14e-01	2.89e+01	8.11e-02	3.75e-09	7.45e-09	0	0
F_6	3.19e-04	4.16e-04	0	0	5.24e-00	7.07e-01	6.86e-10	1.29e-09	0	0
F_7	7.86e-05	8.25e-05	7.70e-02	2.65e-02	1.45e-05	8.15e-04	3.25e-05	2.55e-05	1.211e-04	1.069e-04
F_8	−3.12e+03	3.50e+02	1.34e-02	0	-	-	−5.58e-101	1.67e + 102	−12 569.486	1.850e-012
F_9	0	0	0	0	3.03e-13	2.86e-13	0	0	0	0
F_{10}	7.99e-15	2.33e-15	0	0	8.88e-16	0	8.88e-16	0	8.881e-016	0
F_{11}	0	0	0	0	0	0	0	0	0	0
F_{12}	4.57e-04	5.77e-04	1.69e-32	3.99E-33	9.07e-32	0	8.53e-14	2. 61e-10	1.570e-032	5.567e-048
F_{13}	1.15e-02	2.17e-02	2.28e-32	3.45e-32	2.16e-32	0	2.63e-11	6.65e-10	1.349e-032	5.567e-048

3.4 MHHO方法与其他优化算法大规模问题对比

为了进一步研究MHHO方法与其他群智能优化改进方法在大规模问题上的性能差异,参考文献[21]的函数设置,选择7种公共函数进行对比研究,问题维度D设置为1 000. 对比算法包括改进粒子群(CLPSO)[24]、协方差进化(CMAES)[24]、改进的自适应差分进化(SaDE)[24]、组合差分进化(CoDE)[24]、改进动态和声搜索算法(DIHS)[24]、改进鲸鱼优化算法(IWOA)[24]和改进飞蛾扑火优化算法(IMFO)[25]等,参数设置和部分测试数据来自文献[24-25]. 本文算法最大评估次数fEvals值从15 000调整为5×10^6,测试结果见表7.

由表7可见,本文MHHO方法在上述测试函数中具有一定的优势,除F_{12}的性能低于IWOA外,其他测试函数都获得了全局极值. 经过对比发现,文献[24]中F_{12}和F_{14}函数的表达式有误(F_{12}中第1个X_d应为X_i, F_{14}求和公式中的d应为$d-1$),其理论最优值为0,理论最优位置为全1. 通过Matlab将上述全1位置代入公式,得F_{12}最优值为1.349 783 804 395 672e-31,并不是0. 因此,本文算法也获得了全局极值.

另外,文献[25]中IMFO算法针对基准函数F_3、F_4、F_6、F_8、F_{13}的寻优成功率偏低,本文利用改进算法MHHO与其进行对比研究. 公平起见,最大评估次数一样(fEvals = 50 000),成功率阈值设为1e-10,统计30次测试的相关指标数据见表8.

表 7 大规模问题上的算法寻优均值指标数据($D = 1\,000$)

函数	CLPSO(2006)	CMAES(2007)	SaDE(2009)	CoDE(2011)	DIHS(2015)	IWOA(2017)	IMFO(2020)	MHHO
F_1	3.14e-21	7.06e-01	2.22e-01	6.06e-02	1.98e-23	0	0	0
F_2	7.73e-02	1.78e+0	1.57e-02	3.16e-03	5.24e-12	0	0	0
F_5	7.45e+02	3.26e+03	3.05e+03	2.21e+03	1.21e+03	9.90e+02	8.28e-08	0
F_8	4.96e+03	3.60e+02	8.22e+02	1.06e+03	2.34e+02	0	0	0
F_9	1.95e+01	2.13e-01	1.21e+01	9.19e+00	4.02e-13	8.88e-16	8.89e-16	8.881 8e-016
F_{10}	2.78e-15	4.94e-01	9.25e-01	1.23e+00	2.50e-15	0	0	0
F_{12}	1.90e+03	1.19e+01	1.15e+02	9.94e+01	9.97e-26	0	1.54e-06	1.349 8e-031

表 8 大规模问题的数值对比实验($D = 1\,000$)

函数	IMFO			MHHO		
	均值	标准差	成功率/%	均值	标准差	成功率/%
F_3	1.50e+05	7.10e+05	36.67	0	0	100
F_4	3.88e-12	2.07e-11	100	8.204e-191	0	100
F_6	0.048 4	0.120 4	33.33	0	0	100
F_8	−418 998	0.259 2	93.33	−418 982.887 3	1.776 1e-010	100
F_{13}	0.017 7	0.057 6	36.67	1.349 8e-032	5.567 4e-048	100

由表8可见,在有限次最大评估次数条件下,MHHO方法在上述5个大规模基准函数上的测试均值均一致优于IMFO,除F_4外(精度也很高)都获得了全局极值,成功率为100%. 从标准差指标衡量,MHHO方法非常稳定,对参数变化的鲁棒性也较好,不同维度的性能实验验证了这种一致性.

IHHO算法[16]比较了Elliptic、 Schwefel2.22、Alpine和Salomon等($F_1' \sim F_4'$)四种函数,其实验设置

为:维度$D = 1\,000$,种群$N = 50$,迭代次数$T = 1\,500$次. 该算法在每个回合大约采用6次牛顿局部加强,相当于最大评估次数fEvals为$56 \times 1\,500 = 84\,000$. 考虑到文献[16]选择的函数不需要平衡探索和开发, MHHO方法的子群数设置为4, fEvals值相同,数值实验结果见表9.

表 9 MHHO与IHHO算法的性能对比($D = 1\,000$)

函数	MHHO				IHHO[16]			
	均值	标准差	最差值	最优值	均值	标准差	最差值	最优值
F_1'	0	0	0	0	0	0	0	0
F_2'	1.429 6e-288	0	4.288 7e-287	0	1.362e-213	0	2.284e-212	3.390e-238
F_3'	2.462 5e-288	0	7.387 5e-287	0	1.124e-211	0	3.372e-210	1.038e-236
F_4'	0	0	0	0	0	0	0	0

由表9可见,针对不需要平衡探索和开发的4个基准函数, MHHO方法求解精度相对更高, 30次迭代的最好值均获得了全局最优,算法性能稳定. 因此,本文算法不劣于文献[16].

针对不同算法在大规模问题上($D = 1\,000$)的收敛性情况,限于篇幅对比了WOA和HHO算法在求解F_4和F_{12}的收敛情况,其中F_4统计后450次迭代(最优值下降过快). 研究发现,测试函数F_4的最优值为0,在有限的迭代步下, MHHO方法的收敛速度快于WOA和HHO算法,且寻优精度在整个迭代周期均持续提高,而WOA算法在第50次迭代后便停滞了, HHO算法大约在370次左右也开始停滞. 测试函数F_{12}的最优解为1.57e-032, MHHO方法在第80次迭代后获得了全局极值,后续稳定收敛于该值,而WOA算法和HHO算法在第100次迭代后均限入局部极值,无法进一步提高收敛精度.

4 结 论

哈里斯鹰优化算法(HHO)是受自然界中群鹰围捕野兔逃跑时的狩猎行为而提出的新型群智能优化方法,该方法基于灰狼优化和布谷鸟搜索算子,形成了多重围猎策略. 不同学者从非线性能量、最优值变异、算子融合等视角提出了诸多改进,但仍然存在探索和开发的不平衡问题. 本文提出了一种方形邻域拓扑结构的多子群划分思路,该策略下的子群个体依概率受到来自横向或纵向的最优值影响,纵横双向没有直接的信息互动,进而在探索与开发之间取得了均衡. 基于固定概率P建立了一种子群之间的个体信息交换机制,即通过随机数组确定被交换个体的行与列,且要求两个个体的行与列均不重合,这一措施促进了不同子群的个体流动,避免了算法陷入局部最优. 提出了基于历史进化信息进行HHO算法中的算子选择思路,区别于现有基本HHO算法中利用随机概率来划分子群的思路,从而能够更好地利用现有问题的信息.

利用中低维度和较高维度的基准函数进行实验测试,并进行跨文献的广泛对比研究发现,改进方法的开发和探索能力均衡性较好,能够有效获得全局极值附近区域的位置. 对于面向精度问题设置的测试函数,通过减少子群数增加有效迭代次数,显著增强了部分问题的寻优精度. 跨文献数值实验比较表明,所提出算法具有较为强劲的全局寻优性能和算法鲁棒性. 这种基于方形邻域拓扑结构和随机数组的子群个体信息交换思路和方法,可以为其他群智能优化方法提供参考和借鉴.

参考文献(References)

[1] 黄清宝, 李俊兴, 宋春宁, 等. 基于余弦控制因子和多项式变异的鲸鱼优化算法[J]. 控制与决策, 2020, 35(3): 559-568.
(Huang Q B, Li J X, Song C N, et al. Whale optimization algorithm based on cosine control factor and polynomial mutation[J]. Control and Decision, 2020, 35(3): 559-568.)

[2] Askari Q, Younas I, Saeed M. Political optimizer: A novel socio-inspired meta-heuristic for global optimization[J]. Knowledge-Based Systems, 2020, 195: 105709.

[3] Heidari A A, Mirjalili S, Faris H, et al. Harris hawks optimization: Algorithm and applications[J]. Future Generation Computer Systems, 2019, 97: 849-872.

[4] Wunnava A, Kumar Naik M, Panda R, et al. A differential evolutionary adaptive harris hawks optimization for two dimensional practical Masi entropy-based multilevel image thresholding[J]. Journal of King Saud University — Computer and Information Sciences, DOI: 10.1016/j.jksuci.2020.05.001.

[5] Essa F A, Abd Elaziz M, Elsheikh A H. An enhanced productivity prediction model of active solar still using artificial neural network and hyarris hawks optimizer[J]. Applied Thermal Engineering, 2020, 170: 115020.

[6] Kamboj V K, Nandi A, Bhadoria A, et al. An intensify harris hawks optimizer for numerical and engineering optimization problems[J]. Applied Soft Computing, 2020, 89: 106018.

[7] Houssein E H, Hosney M E, Oliva D, et al. A novel hybrid harris hawks optimization and support vector machines

for drug design and discovery[J]. Computers & Chemical Engineering, 2020, 133: 106656.

[8] Qu C W, He W, Peng X N, et al. Harris hawks optimization with information exchange[J]. Applied Mathematical Modelling, 2020, 84: 52-75.

[9] Khalifeh S, Akbarifard S, Khalifeh V, et al. Optimization of water distribution of network systems using the hyarris hawks optimization algorithm [J]. Methods X, 2020, 7: 100948.

[10] Chen H L, Jiao S, Wang M J, et al. Parameters identification of photovoltaic cells and modules using diversification-enriched hyarris hawks optimization with chaotic drifts[J]. Journal of Cleaner Production, 2020, 244: 118778.

[11] Tikhamarine Y, Souag-Gamane D, Ahmed A N, et al. Rainfall-runoff modelling using improved machine learning methods: Harris hawks optimizer vs. particle swarm optimization[J]. Journal of Hydrology, 2020, 589: 125133.

[12] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer[J]. Advances in Engineering Software, 2014, 69: 46-61.

[13] Yang X S, Deb S. Engineering optimisation by cuckoo search[J]. International Journal of Mathematical Modelling and Numerical Optimisation, 2010, 1(4): 330.

[14] Chen H, Heidari A A, Chen H L, et al. Multi-population differential evolution-assisted hyarris hawks optimization: Framework and case studies[J]. Future Generation Computer Systems, 2020, 111: 175-198.

[15] Gupta S, Deep K, Heidari A A, et al. Opposition-based learning hyarris hawks optimization with advanced transition rules: Principles and analysis[J]. Expert Systems With Applications, 2020, 158: 113510.

[16] 赵世杰, 高雷阜, 于冬梅, 等. 融合能量周期性递减与牛顿局部增强的改进HHO算法[J]. 控制与决策, 2021, 36(3): 629-636.
(Zhao S J, Gao L F, Yu D M, et al. Improved hyarris hawks optimization coupling energy cycle decline mechanism and Newton local enhancement strategy[J]. Control and Decision, 2021, 36(3): 629-636.)

[17] 顾清华, 李学现, 卢才武, 等. 求解高维复杂函数的遗传-灰狼混合算法[J]. 控制与决策, 2020, 35(5): 1191-1198.
(Gu Q H, Li X X, Lu C W, et al. Hybrid genetic grey wolf algorithm for high dimensional complex function optimization[J]. Control and Decision, 2020, 35(5): 1191-1198.)

[18] 吴文海, 郭晓峰, 周思羽, 等. 基于随机邻域策略和广义反向学习的自适应差分进化算法[J]. 系统工程与电子技术, 2021, 43(7): 1928-1942.
(Wu W H, Guo X F, Zhou S Y, et al. Self-adaptive differential evolution algorithm with random neighborhood-based strategy and generalized opposition-based learning[J]. Systems Engineering and Electronics, 2021, 43(7): 1928-1942.)

[19] 刘丽杰, 刘继承, 张强. 基于动态行为选择的和声搜索算法[J]. 控制与决策, 2021, 36(3): 577-588.
(Liu L J, Liu J C, Zhang Q. Harmony search algorithm based on dynamic behavior selection[J]. Control and Decision, 2021, 36(3): 577-588.)

[20] 肖子雅, 刘升. 精英反向黄金正弦鲸鱼算法及其工程优化研究[J]. 电子学报, 2019, 47(10): 2177-2186.
(Xiao Z Y, Liu S. Study on elite opposition-based golden-sine whale optimization algorithm and its application of project optimization[J]. Acta Electronica Sinica, 2019, 47(10): 2177-2186.)

[21] 褚鼎立, 陈红, 王旭光. 基于自适应权重和模拟退火的鲸鱼优化算法[J]. 电子学报, 2019, 47(5): 992-999.
(Chu D L, Chen H, Wang X G. Whale optimization algorithm based on adaptive weight and simulated annealing[J]. Acta Electronica Sinica, 2019, 47(5): 992-999.)

[22] 王坚浩, 张亮, 史超, 等. 基于混沌搜索策略的鲸鱼优化算法[J]. 控制与决策, 2019, 34(9): 1893-1900.
(Wang J H, Zhang L, Shi C, et al. Whale optimization algorithm based on chaotic search strategy[J]. Control and Decision, 2019, 34(9): 1893-1900.)

[23] 吴泽忠, 宋菲. 基于改进螺旋更新位置模型的鲸鱼优化算法[J]. 系统工程理论与实践, 2019, 39(11): 2928-2944.
(Wu Z Z, Song F. Whale optimization algorithm based on improved spiral update position model[J]. Systems Engineering — Theory & Practice, 2019, 39(11): 2928-2944.)

[24] 龙文, 蔡绍洪, 焦建军, 等. 求解大规模优化问题的改进鲸鱼优化算法[J]. 系统工程理论与实践, 2017, 37(11): 2983-2994.
(Long W, Cai S H, Jiao J J, et al. Improved whale optimization algorithm for large scale optimization problems[J]. Systems Engineering — Theory & Practice, 2017, 37(11): 2983-2994.)

[25] 刘小龙. 基于统计指导的飞蛾扑火算法求解大规模优化问题[J]. 控制与决策, 2020, 35(4): 901-908.
(Liu X L. Moth-flame algorithm based on statistical guidance for large-scale optimization problems[J]. Control and Decision, 2020, 35(4): 901-908.)

作者简介

刘小龙(1977−), 男, 讲师, 博士, 从事仿生优化与计算智能等研究, E-mail: xlliu@scut.edu.cn;

梁彤缨(1961−), 男, 教授, 博士生导师, 从事资本市场与公司财务、创新政策组合优化等研究, E-mail: bmtliang@scut.edu.cn.

(责任编辑: 郑晓蕾)