

DOI:10.19651/j.cnki.emt.2313837

引入改进迭代局部搜索的灰狼算法及应用^{*}文昌俊^{1,2} 陈凡^{1,2} 陈洋洋^{1,2} 何永豪^{1,2}

(1.湖北工业大学机械工程学院 武汉 430068; 2.湖北省现代制造质量工程重点实验室 武汉 430068)

摘要:针对标准灰狼算法(GWO)收敛速度慢,易陷入局部最优等缺点,提出一种引入改进迭代局部搜索的灰狼算法(IGWO)。首先,通过佳点集策略增强初始种群的均匀性与多样性;其次,采用双收敛因子,收敛因子基于种群位置非线性自适应更新,在种群寻优全期平衡全局勘探与局部开发能力;再次,在种群位置更新公式引入欧氏动态权重与莱维飞行策略,提升寻优精度,并帮助种群跳出局部最优值;最后,引入改进迭代局部搜索,使算法的搜索能力更加灵活,帮助算法加速收敛。通过10个基准测试函数的仿真分析及种群寻优平衡性对比,证明了IGWO具有更优的寻优精度、稳定性及收敛速度,随后将IGWO应用于工程优化问题中,相比GWO、GJO、WOA、HSSAHHO、SCHOA、NCPGWO、DSFGWO 7种算法,适应度分别优化了3.25%、27.2%、28.9%、3.15%、3.04%、0.23%、0.07%,证实了在工程应用中的可行性和有效性。

关键词:灰狼算法;佳点集;双收敛因子;莱维飞行和欧氏动态权重策略;迭代局部搜索

中图分类号: TP301.6 **文献标识码:** A **国家标准学科分类代码:** 520.1040

Improved iterative local search grey wolf algorithm and its application

Wen Changjun^{1,2} Chen Fan^{1,2} Chen Yangyang^{1,2} He Yonghao^{1,2}

(1. School of Mechanical Engineering, Hubei University of Technology, Wuhan 430068, China;

2. Hubei Key Laboratory of Modern Manufacturing Quality Engineering, Wuhan 430068, China)

Abstract: Aiming at the disadvantages of slow convergence of standard grey wolf optimizer (GWO) and easy to fall into local optimality, an improved grey wolf optimizer (IGWO) with improved iterative local search is proposed. First, the uniformity and diversity of the initial population were enhanced by the strategy of the best point set. Secondly, the dual convergence factor is used, which is nonlinear and adaptive updating based on population location, to balance the global exploration and local development ability in the whole period of population optimization. Thirdly, European dynamic weight and Levy flight strategy were introduced into the population position updating formula to improve the optimization accuracy and help the population jump out of the local optimal value. Finally, the improved iterative local search is introduced to make the search ability of the algorithm more flexible and help the algorithm accelerate convergence. Through the simulation analysis of 10 benchmark test functions and the comparison of population optimization balance, it is proved that IGWO has better optimization accuracy, stability and convergence speed. Then IGWO was applied to the engineering optimization problem. Compared with GWO, GJO, WOA, HSSAHHO, SCHOA, NCPGWO, DSFGWO 7 algorithms, the fitness was optimized by 3.25%, 27.2%, 28.9%, 3.15%, 3.04%, 2.33%, 0.07%, respectively. The feasibility and effectiveness in engineering application are proved.

Keywords: gray wolf algorithm; good point set; double convergence factor; levy flight and dynamic weight strategies; iterative local search

0 引言

群智能算法作为一种新兴的演化计算技术,受到了研究学者的广泛关注。灰狼算法(grey wolf optimizer, GWO)^[1]

作为近几年新起的群智能算法之一,因其具有参数简单、实现方便、稳定性强等特点,在解决特征选择、神经网络优化、路径规划等优化问题时表现出了良好的性能。但也存在求解精度不高,收敛速度慢,易陷入局部最优等缺陷。

收稿日期:2023-06-06

* 基金项目:国家自然科学基金(51875180)项目资助

针对以上不足,相关学者在从初始化种群、收敛因子、位置更新公式、跳出局部最优方法上提出改进策略。滕志军等^[2]针对标准灰狼算法易陷入局部最优等问题,通过采用 Tent 映射初始化种群,引入非线性控制参数等策略改进算法,能更快搜索到全局最优解,且鲁棒性更好。王海峰等^[3]针对传统复原算法在退化图像复原过程中过度依赖先验知识弊端,利用改进收敛因子与动态权重指导种群移动的方式优化 GWO,并将其用于解决 BP 神经网络因初始化权值和阈值的随机性而导致的易收敛及收敛速度慢的问题。王秋萍等^[4]针对 GWO 算法存在寻优精度不高和收敛较慢的问题,设计出余弦变化收敛因子和动态权重位置更新方法,取得了较高的寻优精度与收敛速度。张大明等^[5]为了充分利用 GWO 特有的头狼机制,提出一种双向搜索策略,通过一种停滞检测机制来帮助狼群判断当前状态,跳出局部最优。以上改进基本都是对 GWO 算法本身进行优化,虽提高了部分性能,但全局搜索能力依然较弱,算法本身机制仍然存在局限性。

为此,研究者们引入其他群智能算法的优秀搜索策略来改进 GWO 算法机制,以更好的提高算法的寻优性能。Heidari 等^[6]针对 GWO 在某些问题求解中因种群的多样性不足而易陷入局部最优等问题,提出了一种基于 α 和 β 双狼引领、引入布谷鸟策略的灰狼算法来提升算法全局寻优能力,但存在局部寻优精度不高的问题。曾碟等^[7]引入差分策略变异重构包围步长,更好地平衡了算法的全局勘探与局部搜索能力。刘紫燕等^[8]提出一种基于杂交策略的自适应灰狼,首先引入遗传杂交策略以增强灰狼初始群体的多样性,其次使用蝠鲞翻滚觅食策略并加入了自适应因子以调节寻优能力,借助这两种群智能策略较强的全局搜索能力,帮助算法动态跳出局部最优。以上引入策略虽然改善了算法本身的机制,但未设置协调机制,无法较好地平衡其他算法策略与 GWO 本身机制的性能。

基于此,本文提出一种引入改进迭代局部搜索的灰狼算法(improved grey wolf optimizer, IGWO)。采用佳点集初始化种群,生成更均匀个体,采用非线性自适应更新双收敛因子,动态平衡勘探与开发能力;针对灰狼的位置更新机制,采用欧氏距离的动态权重与改良莱维飞行并行的策略,既提升 GWO 机制本身的寻优精度,又帮助算法跳出局部最优,提升寻优准确率;引入改进迭代局部搜索策略,平衡与协调两种不同的位置更新策略,提高算法的灵活度与全面性。接着采用控制变量法验证了 IGWO 每个策略发挥的作用,随后通过 10 个基准测试函数的仿真分析及种群寻优平衡性对比,证明 IGWO 拥有更好的寻优性能与强大的鲁棒性。最后将其应用于工程优化问题,与其他 7 种算法相比,IGWO 取得了最优的收敛速度与适应度值,有效地提升了工程应用效果。

1 标准灰狼算法

GWO 算法是 Mirjalil 等^[1]于 2014 年受狼群捕猎活动启发而提出的一种群智能算法,具有参数简单、实现方便、稳定性强等特点。该算法将狼群整体当做初始解种群,按照自然条件下灰狼的社会分级制度, α 狼对应最优解位置, β 狼对应次优解位置, δ 狼对应第三优解位置。前三只狼追踪猎物 and 引导狼群,其余解划分为 ω 狼,负责围捕猎物。

1) 追踪猎物

头狼追踪猎物并引领狼群,其数学模型表示为:

$$D = |C \cdot X_p(t) - X_k(t)| \quad (1)$$

$$X_k(t+1) = X_p(t) - A \cdot D \quad (2)$$

其中, t 表示当前迭代次数; $X_k(t)$ 表示第 k 只灰狼个体在第 t 次迭代的位置位置向量; $X_p(t)$ 表示第 t 次迭代猎物的位置向量; $X_k(t+1)$ 表示更新后灰狼个体位置向量; D 表示灰狼搜索步长; A 和 C 是控制向量, A 用于调节灰狼个体移动的方向, C 用于修正猎物位置,分别下式计算:

$$A = 2a \cdot rand() - a \quad (3)$$

$$C = 2 \cdot rand() \quad (4)$$

其中, $rand()$ 为 $(0, 1)$ 之间的随机数值; a 为收敛因子,随着迭代次数从 2 线性减小至 0,计算公式为式(5)。

$$a = 2 - 2 \cdot \frac{t}{t_{\max}} \quad (5)$$

式中: t 为当前迭代次数, t_{\max} 为最大迭代值。

2) 围捕猎物

ω 狼在三只头狼的带领下围捕猎物,位置更新公式如下:

$$D_\alpha = |C_1 \cdot X_\alpha(t) - X_k(t)| \quad (6)$$

$$D_\beta = |C_2 \cdot X_\beta(t) - X_k(t)| \quad (7)$$

$$D_\delta = |C_3 \cdot X_\delta(t) - X_k(t)| \quad (8)$$

$$X_1(t+1) = X_k(t) - A_1 \cdot D_\alpha \quad (9)$$

$$X_2(t+1) = X_k(t) - A_2 \cdot D_\beta \quad (10)$$

$$X_3(t+1) = X_k(t) - A_3 \cdot D_\delta \quad (11)$$

$$X_k(t+1) = \frac{1}{3} \sum_{i=1,2,3} X_i(t+1) \quad (12)$$

式中: $X_\alpha(t)$ 、 $X_\beta(t)$ 、 $X_\delta(t)$ 分别为当前 α 、 β 、 δ 狼的位置; D_α 、 D_β 、 D_δ 分别表示三只头狼的搜索步长; $X_i(t+1)$ ($i=1,2,3$) 为个体在 α 、 β 、 δ 狼引导下的新位置。

2 IGWO 算法

2.1 佳点集

GWO 算法在处理函数优化问题时,初始种群信息通常使用随机产生的数值,但这会导致种群的多样性和均匀性变差,影响算法的寻优结果。因此,本文引入佳点集^[9]理论确定初始种群的分布,其理论优越性为:使用佳点集选择的样本偏差小于随机选取的样本偏差,其原理为:设 G_S 是 S 维欧氏空间中的单位立方体,若 $r \in G_S$,则数学模型为:

$$P_n(k) = \{(\{r_1^{(n)} \cdot k\}, \{r_2^{(n)} \cdot k\}, \dots, \{r_s^{(n)} \cdot k\}), 1 \leq k \leq n\} \quad (13)$$

将佳点集映射到所求优化问题的搜索空间,公式为:
 $x_i(j) = (ub_j - lb_j) \cdot \{r_j^{(i)} \cdot k\} + lb_j \quad (14)$
 $x_i(j)$ 是灰狼个体位置, ub_j 和 lb_j 为第 j 维上下界。

混沌运动以其随机性、规律性、以及遍历性的特性,在处理函数优化问题时能够帮助算法跳出局部最优值,增强种群多样性。目前文献中大多数采用的混沌映射有 Cubic 映射、Logistic 映射、Singer 映射等。张孟健等^[10]证明 Cubic 映射比 Logistic、Singer 映射的遍历均匀性更好以及能够提高算法的寻优速度,同时能产生分布均匀的初始数值。

因此,本文选取混沌映射中的 Cubic 映射与佳点集作为对比。构造种群个数为 500, 维度为 1 时的随机生成的初始种群,取值范围为 $[0, 1]$ 。图 1 为 Cubic 映射和佳点集在上述条件下的初始种群分布图。图 2 则为 Cubic 映射^[10]和佳点集的频率分布直方图。从图 1、2 可以看出佳点集呈均匀分布,且效果较 Cubic 映射误差更小,在初始化过程中具有较好的遍历性与均匀性。

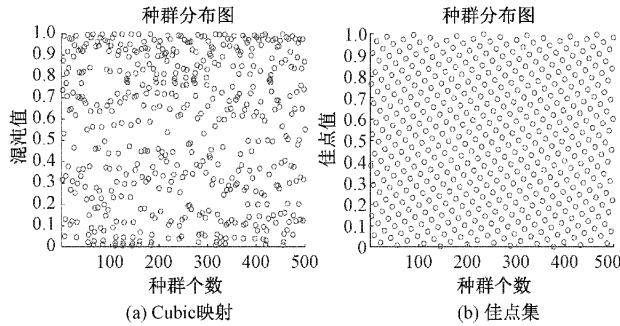


图 1 初始种群在一维空间分布

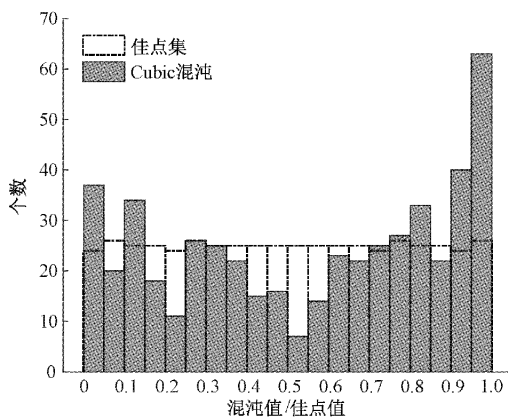


图 2 频率分布直方图

2.2 改进非线性收敛因子

GWO 的全局追踪和局部围捕能力由 $|A|$ 决定,当 $|A| \leq 1$ 时,狼群跟随头狼围捕猎物; $|A| > 1$,狼群扩散追踪猎物。同时,由式(2)可知, A 值随收敛因子 a 的变化而变化。即 GWO 算法的全局搜索和局部开发能力取决于

收敛因子 a 的变化。由式(3)可知,收敛因子 a 随迭代次数的增加而线性减小。然而,由于 GWO 的搜索过程较为繁杂, a 的线性变化不能准确地体现算法的实际寻优过程。文献[11]中提出一种收敛因子 a ,其随迭代次数非线性变化,且通过基准函数的测试结果表明,采用非线性收敛因子策略比线性收敛因子策略寻优精度更高。因此,IGWO 依据改进局部迭代搜索策略将狼群分为捕猎狼和探索狼,并提出相应的非线性双收敛因子。

捕猎狼采用一种自适应收敛因子 a_1 ;算法在前 60% 迭代寻优周期中,给予一个自 2 到 1 的区间,使种群以大步长、慢速度遍历整个搜索空间,帮助算法快速勘探全局最优位置,加速算法收敛;后 40% 迭代寻优周期中,算法开始逐渐收敛,种群开始局部搜寻最优解,此时给予一个自 1 到 0 的区间,帮助算法以小步长精细开发最优位置,且此时收敛因子开始扰动,减少种群陷入局部极值的概率,提高算法寻优精度。自适应收敛因子 a_1 数学模型如式(15)所示。

$$a_1 = \begin{cases} a_{\min} + \gamma_1 \cdot (\cos(t \cdot \pi / \gamma_2) + \gamma_3) & t/t_{\max} \leq 0.6 \\ a_{\min} + \lambda_1 \cdot \sin(\lambda_2 \cdot t \cdot \pi) + \lambda_3 & t/t_{\max} > 0.6 \end{cases} \quad (15)$$

其中, $\gamma_1, \gamma_2, \gamma_3, \lambda_1, \lambda_2, \lambda_3, a_{\min}$ 为常数系数, t 为当前迭代次数, t_{\max} 为最大迭代次数。

探索狼则采用 \sin 正弦函数控制收敛因子 a_2 控制取值,在整个迭代周期内,收敛因子 a_2 随迭代次数在 $[1, 0]$ 非线性递减,且越接近迭代后期,递减越慢,增强探索狼的全局勘探能力。

$$a_2 = (a_{\max 2} - a_{\min 2}) + \sin\left(\pi \cdot \left(\frac{t}{t_{\max}}\right)^{0.11} + \frac{\pi}{2}\right) \quad (16)$$

其中, $a_{\max 2} = 1, a_{\min 2} = 0$ 。

改进的双收敛因子与原收敛因子的曲线如图 3 所示。

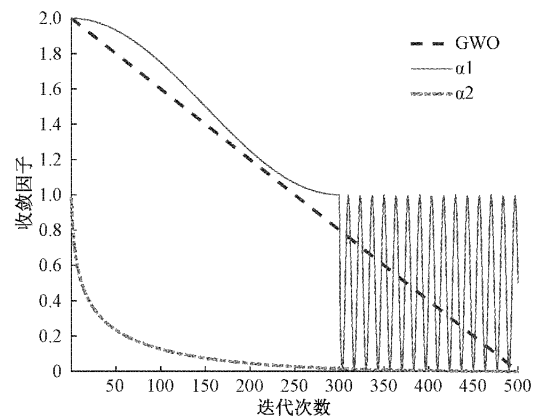


图 3 收敛因子变化曲线图

2.3 基于欧氏距离的动态权重更新策略

GWO 算法在 α, β, δ 狼的带领下进行位置更新,由式(12)可以看出标准 GWO 采用平均权重作为灰狼个体的位置更新,表示 α, β, δ 具有同等影响力,然而 α, β, δ 是具有不同特征和种群领导力。因此,本文采用一种基于欧氏距

离的权重更新策略,使用式(9)、(10)、(11)中 α 、 β 、 δ 的距离占比作为权重系数对灰狼位置机制进行更新,具体公式如式(17)所示。

$$w_i = \frac{X_i(t+1)}{X_1(t+1) + X_2(t+1) + X_3(t+1)} \quad (17)$$

其中, w_i ($i=1,2,3$)为基于欧氏距离的比例权重, $X_i(t+1)$ ($i=1,2,3$)则表示为 α 、 β 、 δ 头狼引导下的位置更新公式。结合式(17),则基于欧氏距离的动态权重更新策略的围捕灰狼位置公式为:

$$X'_k(t+1) = \frac{1}{3} \sum_{i=1,2,3} w_i X_i(t+1) \quad (18)$$

式中:通过比例权重 w 的迭代变化,动态调整 α 、 β 、 δ 对灰狼种群个体的影响,既能增强 β 、 δ 的影响力,又能避免个体过于聚集于 α 狼。能更好地平衡算法的学习搜索能力。

2.4 引入 Levy 飞行更新策略

Levy 飞行是由法国数学家 Paul Pierrel Levy 提出的一种马尔可夫过程,Levy 飞行属于随机游走的一种形式,其步长的概率分布为重尾分布。Levy 飞行的特点为高频的小步移动,过程中间或出现大跨步,使得运动运动轨迹类似于随机扰动,继而改变种群的运动体系。虽然它的运动方向是随机的,但其运动步长却是按幂次律分布的。将改进局部迭代搜索策略与 Levy 飞行策略相结合,可以扩大搜索范围,并且增加了种群多样性,避免陷入局部最优。

在布谷鸟算法中,Yang 等^[12]采用 Mantegna 算法来模拟 Levy 飞行路径的步长 L_v 计算公式:

$$L_v = \frac{u}{|v|^{\frac{1}{\beta}}} \quad (19)$$

式中: β 的取值范围在 $[0, 2]$ 之间,通常取 1.5; u 、 v 是服从式(20)的正态分布随机数:

$$u \sim N(0, \sigma_u^2), v \sim N(0, \sigma_v^2) \quad (20)$$

其中, σ_u 、 σ_v 是符合正态分布的标准差,其取值定义如下:

$$\sigma_u = \left\{ \frac{\Gamma(1+\beta) \sin(\beta\pi/2)}{\Gamma((1+\beta)/2) 2^{(\beta-1)/2} \beta} \right\}^{1/\beta} \sigma_v = 1 \quad (21)$$

结合上式,则引入 Levy 飞行策略与新的收敛因子的探索灰狼位置公式为:

$$X_k''(t+1) = a_2 L_v + X_k(t) \quad (22)$$

其中, $X_k''(t+1)$ 为采用 Levy 飞行策略更新后的个体位置。

受文献[12]启发,使用 Mantegna 算法绘制了一个在平面上连续 1 000 步的 Levy 飞行轨迹与随机运动轨迹进行对比,如图 4 所示。图 5 则为 Levy 飞行连续 1 000 步的步长频率。从图 4、5 可以看出 Levy 飞行的移动范围更广,搜索性更强,更容易跳出局部最优。

2.5 改进的迭代局部搜索策略

局部搜索(local search, LS)算法是解决最优化问题的一种启发式算法。局部搜索从一个初始解出发,然后搜索

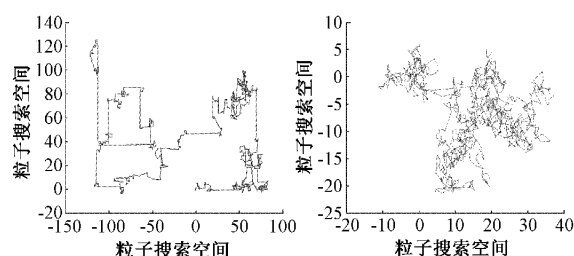


图4 Levy 飞行与随机运动轨迹对比图

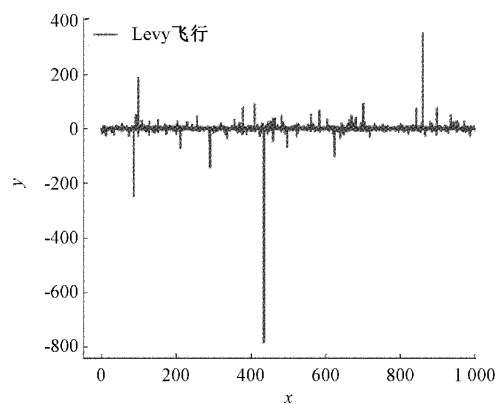


图5 Levy 飞行的步长频率图

解的邻域,如有更优的解则移动至该解并继续执行搜索,否则就停止算法获得局部最优解。迭代局部搜索(iterative local search, ILS)^[13-14]是一种探索性的方法,它对通过局部搜索得到的局部最优解进行扰动,然后重新搜索局部解。

改进的迭代局部搜索首先通过欧氏距离的动态权重策略更新捕猎狼位置,随后将捕猎狼位置作为初始解,在其附近执行局部搜索,然后通过 Levy 飞行更新探索狼的位置来干扰初始解,最后再次搜索更新后的位置。它充分利用了当前个体的位置信息和全局最优解的位置信息,使搜索更加灵活。第 1 种改进为探索狼采用 Levy 飞行策略进行干扰的机制,使个体跳到当前全局最优位置附近,跳出局部最优位置;第 2 次修改采用 ILS 方法,即对初始解和扰动后的中间解进行局部搜索,得到稳定的优化结果,提升局部搜索的精度;第 3 次修改利用贪婪策略对初始解和中间解局部搜索后的解 1 和 2 进行比较(即对捕猎狼和探索狼进行适应度比较),选取较优值作为最终解,获得最优的全局搜索结果,保证解的质量。改进后的迭代局部搜索既充分利用了当前个体的位置信息,又充分引入了全局最优解的位置信息,寻优更灵活,这是一种可行的全局优化方案。

如图 6 所示,第 1 次调整表示个体扰动后,在当前全局最优解附近寻到更优解;第 2 次调整表示个体利用自身有利位置寻到更优解,两者都有助于跳出局部最优解。同时本文没有与当前全局最优解进行比较而取代,用来维持种群多样性,防止前期过早陷入聚集。

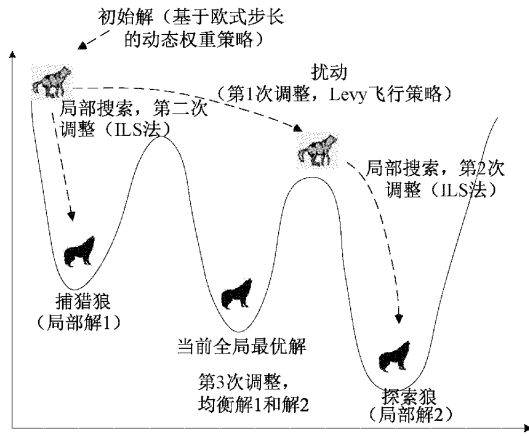


图 6 改进的迭代局部搜索原理

ILS 策略算法流程如下：

步骤 1) 借助全局最优解 X_{best} 及 Levy 飞行策略对初始解 X^* 进行扰动, 得到中间解 X^{**} 。扰动公式如下：

$$X^{**} = a_2 L_v | X^* - X_{best} | + X^* \quad (23)$$

原 Levy 飞行策略灰狼公式如下：

$$X_k''(t+1) = a_2 L_v + X_k'(t) \quad (24)$$

其中, X_{best} 为初始种群根据适应度排序出的全局最优解。 X^{**} 为 $X_k''(t)$, X^* 为 $X_k'(t)$, 为灰狼种群基于欧氏距离动态权重更新后的位置。依据式(18)得出。 X^{**} 为 $X_k''(t+1)$, 即进行 Levy 飞行之后的位置。依式(23)得出。

步骤 2) 对初始解 X^* 进行局部搜索, 公式如下：

$$X^1 = X^* \cdot rand() \quad (25)$$

步骤 3) 对中间解 X^{**} 进行局部搜索, 公式如下：

$$X^2 = X^{**} \cdot rand() \quad (26)$$

步骤 4) 计算 X^1 的适应度 $f(X^1)$

步骤 5) 计算 X^2 的适应度 $f(X^2)$

步骤 6) 比较 $f(X^1)$ 和 $f(X^2)$, 选取较优的个体进行位置更新, 即：

$$If \quad f(X^1) \leq f(X^2)$$

$$X = X^1$$

else

$$X = X^2$$

end

根据 GWO 算法和上述 3 次的修改, 将 GWO 算法中灰狼位置公式更新如下：

$$X_k'''(t+1) = \begin{cases} X_k'(t) \cdot rand(), & f(X^1) \leq f(X^2) \\ ((a_2 L_v | X_k'(t) - X_{best} | + X_k'(t)) \cdot rand(), & \text{其他} \end{cases} \quad (27)$$

2.6 IGWO 算法流程

基于前文所述的改进策略, 本文改进灰狼算法的具体流程图如图 7 所示。

本文所提的 IGWO 算法步骤如下：

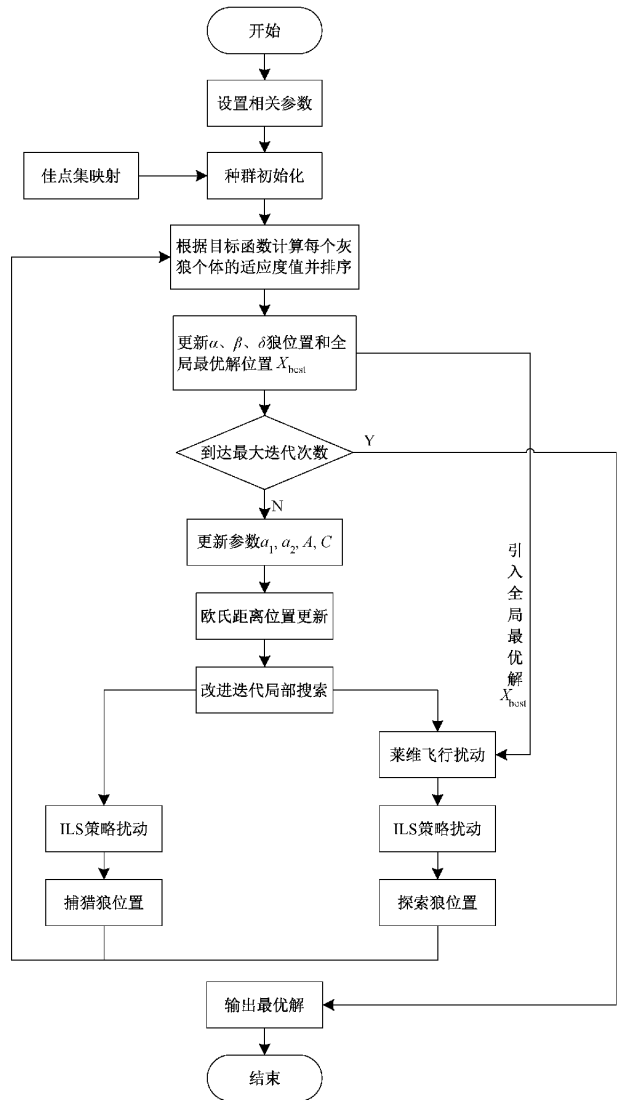


图 7 IGWO 算法流程

步骤 1) 设定种群规模 N , 最大迭代次数 t_{max} , 初始化各参数。

步骤 2) 采用佳点集理论初始化灰狼种群。

步骤 3) 计算灰狼个体适应度, 根据个体适应度确定 α 、 β 、 δ 头狼, 并取全局最优解作为 X_{best} , 然后根据式(15)~(16)计算收敛因子 a_1 、 a_2 。

步骤 4) 依据式(18)更新捕猎狼的个体位置 X_k' , 并将 X_k 作为 ILS 策略的初始解。

步骤 5) 根据 ILS 策略, 依据式(18)与(25)再次更新捕猎狼位置 X_{k+1}' , 并计算其适应度值 $f(X_{k+1}')$ 。

步骤 6) 根据 ILS 与 Levy 飞行策略, 依据式(23)与(26)更新探索狼位置 X_{k+1}'' , 并计算其适应度值 $f(X_{k+1}'')$ 。

步骤 7) 比较 $f(X_{k+1}')$ 与 $f(X_{k+1}'')$ 适应度大小, 根据式(27)确定最终的灰狼位置公式 $X_k'''(t+1)$ 。

步骤 8) 更新迭代次数, 判断是否达到 t_{max} , 满足则输出最优适应度值, 未满足则返回步骤 3)。

2.7 IGWO 算法伪代码

- 1) 输入参数:种群数量 N 、最大迭代次数 t_{max} 、变量上界 ub 、变量下界 lb 、目标函数 $f(x)$
- 2) 佳点集序列初始化灰狼种群,并计算每个个体的适应度 $X_i = (1, 2, \dots, n)$,并对适应度进行排序
- 3) 确定 X_α 、 X_β 、 X_δ 狼,并选取全局最优解的位置为 X_{best}
- 4) 更新参数 a_1, a_2, A, C ,更新迭代次数 $t = 1$
- 5) While $t < t_{max}$
- 6) For $i = 1 : N$
- 7) 欧氏距离位置更新
- 8) ILS 策略扰动
- 9) 捕猎狼位置 X'_{k+1} ,适应度 $f(X'_{k+1})$
- 10) 生成莱维飞行参数 L_V
- 11) 莱维飞行引入 X_{best} 扰动更新
- 12) ILS 策略扰动
- 13) 探索狼位置 X''_{k+1} ,适应度 $f(X''_{k+1})$
- 14) If $f(X'_{k+1}) < f(X''_{k+1})$
- 15) $X''_k(t+1) = X'_{k+1}$
- 16) Else
- 17) $X''_k(t+1) = X''_{k+1}$
- 18) End if
- 19) End for
- 20) 更新参数 a_1, a_2, A, C
- 21) 计算每个个体适应度,并更新 $X_\alpha, X_\beta, X_\delta, X_{best}$
- 22) $t = t + 1$
- 23) End while
- 24) 输出:最优解 X_{best} ,最优适应度 $f(X_{best})$

3 仿真实验与结果分析

仿真实验环境为:Win10 64 位,AMD Ryzen 5 5600X CPU,频率 3.70 GHz,运行内存 16 GB,MATLAB R2021b。为保证实验公平性,所有算法实验均独立运行 30 次,每次总迭代次数 $t_{max} = 500$ 次,种群规模 N 均设为 30。

3.1 基准测试函数

为了验证 IGWO 算法的有效性,本文选取文献[15]所采用的 10 个基准测试函数用于仿真验证。如表 1 所示,其中 F1~F5 为单峰函数,F6~F10 为复杂多峰函数。

3.2 改进策略对算法寻优性能影响分析

为了验证每个策略发挥的作用和对算法性能的影响,本文将灰狼算法(GWO)、加入佳点集的灰狼算法(GWO1)、加入改进非线性收敛因子的灰狼算法(GWO2)、加入欧氏距离动态权重的灰狼算法(GWO3)、加入 Levy 飞行的灰狼算法(GWO4)、加入迭代局部搜索的灰狼算法(GWO5),本文改进的灰狼算法(IGWO)在表 1 所列的 10 个基准函数中独立运行 30 次,将各算法的最优值、最差值、均值等作为评价指标,并将每个指标的最优值以粗体字处理,同时根据

表 1 基准测试函数

序号	函数	维度	搜索范围	最优解
f_1	Sphere	30	$[-100, 100]$	0
f_2	Schwefel 2.22	30	$[-10, 10]$	0
f_3	Schwefel 1.2	30	$[-100, 100]$	0
f_4	Schwefel 2.21	30	$[-100, 100]$	0
f_5	Rosenbrock	30	$[-30, 30]$	0
f_6	Step	30	$[-100, 100]$	0
f_7	Quartic	30	$[-1.28, 1.28]$	0
f_8	Rastrigin	30	$[-5.12, 5.12]$	0
f_9	Ackley	30	$[-32, 32]$	0
f_{10}	Griewank	30	$[-600, 600]$	0

均值对算法在各个测试函数中的结果进行排名(如均值相等则比较标准差),Rank 是指每种改进策略在不同的函数中的排名情况,Ave rank 是指对排名结果取平均,Total rank 为总排名。具体结果如表 2 所示。

首先,通过理论最优值可以看出 IGWO 在函数 F1、F2、F3、F4、F8、F10 均能找到理论最优值,且在其余函数的寻优中仍取得了较优的精度,体现了综合改进后 IGWO 的强大性能。整体来说,可以明显看出 5 种改进策略均对原始算法有着不同程度的提高。在运行时间方面,7 种算法中 IGWO 所耗时长相对较短。但会出现单一策略改进的 GWO 耗时更短,由于在 GWO 中加入多种改进策略之后,算法搜索区域更广,找到的解更多,导致算法的寻优时间变长,但均在合理范围之内。

其次,观察 10 个基准函数的均值以及总排名可以看出,迭代局部搜索策略 GWO5 在五中策略中占据主导地位,说明在迭代中提升局部质量,搜索邻域具有重要意义;其次为欧氏距离权重策略 GWO3,可以动态地提升 β 狼和 δ 狼的权重占比,提升寻优精度,但如存在多个局部最优点, β 狼和 δ 狼易陷入局部最优,继而影响 α 狼的问题;再次为 Levy 飞行策略 GWO4,可以较大范围寻优及多方向前进,可以较好的摆脱局部最优点,在单峰函数 F5 与多峰函数 F6 中,表现出的寻优性能仅次于 IGWO,具有较强的跳出局部极值的寻优能力,但存在范围与方向过大,寻优精度降低的问题;最后分别为 GWO2、GWO1,在全函数测试中对比 GWO 均有寻优精度提升,但影响较小。因为初始种群策略和收敛因子策略影响算法的前期寻优速度和寻优结果的稳定性,对后期精度提升不大。为了直观的表现 5 种改进策略的均值排名,绘制了 5 种改进策略与 GWO 均值排名雷达图,如图 8 所示,其中圆点曲线为 GWO5,其所围成的面积最小,表明 GWO5 整体性能最好,其余依次为 GWO3、GWO4、GWO2、GWO1、GWO。

所以 IGWO 以迭代局部搜索策略 GWO5 为基础,对相邻域分别采用欧氏距离权重策略 GWO3 与莱维飞行策略 GWO4,兼顾了寻优精度与跳出局部最优的能力。并且

表 2 5 种改进策略实验对比

函数	结果	GWO	GWO1	GWO2	GWO3	GWO4	GWO5	IGWO
f_1	最优值	7.37×10^{-29}	2.58×10^{-31}	1.03×10^{-32}	0	5.36×10^{-58}	0	0
	最差值	7.36×10^{-27}	2.35×10^{-29}	3.62×10^{-30}	0	2.17×10^{-50}	0	0
	均值	1.48×10^{-27}	5.28×10^{-30}	8.48×10^{-31}	0	5.61×10^{-52}	0	0
	标准差	2.13×10^{-27}	8.55×10^{-30}	1.15×10^{-30}	0	2.43×10^{-51}	0	0
	运行时间	0.130	0.113	0.109	0.093	0.098	0.076	0.081
	Rank	7	6	5	1	4	1	1
f_2	最优值	3.30×10^{-17}	1.10×10^{-19}	5.25×10^{-22}	1.50×10^{-215}	3.17×10^{-31}	2.66×10^{-230}	0
	最差值	9.42×10^{-17}	8.42×10^{-19}	4.53×10^{-21}	7.88×10^{-215}	4.14×10^{-28}	1.37×10^{-223}	0
	均值	5.94×10^{-17}	3.53×10^{-19}	2.84×10^{-21}	4.99×10^{-215}	6.95×10^{-29}	1.46×10^{-224}	0
	标准差	1.88×10^{-17}	2.25×10^{-19}	1.28×10^{-21}	0	1.32×10^{-28}	0	0
	运行时间	0.152	0.143	0.115	0.108	0.102	0.085	0.097
	Rank	7	6	5	3	4	2	1
f_3	最优值	3.20×10^{-7}	5.97×10^{-9}	1.48×10^{-11}	0	8.31×10^{-12}	0	0
	最差值	2.94×10^{-4}	2.70×10^{-6}	8.44×10^{-9}	0	3.56×10^{-8}	0	0
	均值	3.13×10^{-5}	5.30×10^{-7}	4.74×10^{-9}	0	1.04×10^{-9}	0	0
	标准差	9.23×10^{-5}	8.28×10^{-7}	2.60×10^{-9}	0	1.09×10^{-8}	0	0
	运行时间	0.136	0.121	0.110	0.092	0.094	0.074	0.088
	Rank	7	6	5	1	4	1	1
f_4	最优值	9.61×10^{-8}	4.07×10^{-9}	5.10×10^{-9}	1.29×10^{-211}	5.48×10^{-14}	3.79×10^{-231}	0
	最差值	1.20×10^{-6}	6.92×10^{-7}	1.08×10^{-7}	2.10×10^{-210}	1.64×10^{-9}	1.65×10^{-223}	0
	均值	4.82×10^{-7}	8.98×10^{-8}	2.78×10^{-8}	7.83×10^{-211}	4.39×10^{-10}	1.48×10^{-224}	0
	标准差	3.79×10^{-6}	6.66×10^{-7}	3.84×10^{-8}	0	5.52×10^{-10}	0	0
	运行时间	0.132	0.124	0.112	0.095	0.091	0.072	0.079
	Rank	7	6	5	3	4	2	1
f_5	最优值	2.59×10^1	2.62×10^1	2.55×10^1	2.67×10^1	2.43×10^1	2.61×10^1	2.41×10^1
	最差值	2.88×10^1	2.80×10^1	2.86×10^1	2.73×10^1	2.60×10^1	2.80×10^1	2.59×10^1
	均值	2.73×10^1	2.71×10^1	2.68×10^1	2.70×10^1	2.49×10^1	2.64×10^1	2.32×10^1
	标准差	9.36×10^{-1}	6.93×10^{-1}	8.87×10^{-1}	4.89×10^{-1}	5.35×10^{-1}	7.64×10^{-1}	1.35×10^{-1}
	运行时间	0.124	0.116	0.109	0.087	0.071	0.075	0.078
	Rank	7	6	4	5	2	3	1
f_6	最优值	5.46×10^{-1}	2.48×10^{-1}	4.98×10^{-1}	4.90	1.35×10^{-6}	4.93	2.35×10^{-7}
	最差值	1.24	1.15×10^1	1.26	5.91	3.90×10^{-5}	6.95	2.97×10^{-4}
	均值	8.95×10^{-1}	5.67×10^{-1}	8.12×10^{-1}	5.36	2.55×10^{-4}	6.21	8.25×10^{-5}
	标准差	2.57×10^{-1}	3.13×10^{-1}	2.99×10^{-1}	3.51×10^{-1}	7.13×10^{-5}	7.02×10^{-1}	1.37×10^{-5}
	运行时间	0.137	0.097	0.095	0.112	0.084	0.109	0.092
	Rank	5	3	4	6	2	7	1
f_7	最优值	6.77×10^{-4}	5.33×10^{-4}	3.61×10^{-4}	7.78×10^{-6}	1.20×10^{-3}	1.60×10^{-5}	3.68×10^{-6}
	最差值	6.20×10^{-3}	2.60×10^{-3}	3.00×10^{-3}	1.58×10^{-4}	4.70×10^{-3}	1.44×10^{-4}	7.16×10^{-5}
	均值	2.10×10^{-3}	1.40×10^{-3}	1.50×10^{-3}	5.90×10^{-5}	2.80×10^{-3}	5.98×10^{-5}	2.90×10^{-5}
	标准差	1.60×10^{-3}	6.58×10^{-4}	8.16×10^{-4}	6.09×10^{-5}	1.10×10^{-3}	3.63×10^{-5}	2.54×10^{-5}
	运行时间	0.201	0.126	0.118	0.140	0.185	0.107	0.126
	Rank	6	4	5	2	7	3	1

续表 2

函数	结果	GWO	GWO1	GWO2	GWO3	GWO4	GWO5	IGWO
f_8	最优值	5.68×10^{-14}	0	0	0	0	0	0
	最差值	1.15×10^1	4.09	5.06×10^1	0	5.13	0	0
	均值	4.08	7.89×10^{-1}	9.05×10^{-1}	0	1.41	0	0
	标准差	4.01	1.40	1.74	0	2.30	0	0
	运行时间	0.139	0.112	0.099	0.108	0.095	0.087	0.092
	Rank	7	4	5	1	6	1	1
f_9	最优值	7.54×10^{-14}	7.54×10^{-14}	4.35×10^{-14}	4.44×10^{-15}	7.99×10^{-15}	8.88×10^{-16}	8.88×10^{-16}
	最差值	1.46×10^{-13}	1.10×10^{-13}	6.83×10^{-14}	4.44×10^{-15}	1.51×10^{-14}	8.88×10^{-16}	8.88×10^{-16}
	均值	1.02×10^{-13}	8.43×10^{-14}	5.38×10^{-14}	4.44×10^{-15}	9.41×10^{-15}	8.88×10^{-16}	8.88×10^{-16}
	标准差	2.43×10^{-14}	1.15×10^{-14}	8.28×10^{-15}	0	2.99×10^{-15}	0	0
	运行时间	0.138	0.109	0.097	0.103	0.091	0.086	0.090
	Rank	7	6	5	3	4	1	1
f_{10}	最优值	0	0	0	0	0	0	0
	最差值	2.57×10^{-2}	2.44×10^{-2}	2.04×10^{-2}	0	9.90×10^{-3}	0	0
	均值	4.20×10^{-3}	3.70×10^{-3}	3.10×10^{-3}	0	9.85×10^{-4}	0	0
	标准差	9.10×10^{-3}	8.30×10^{-3}	7.00×10^{-3}	0	3.10×10^{-3}	0	0
	运行时间	0.155	0.107	0.093	0.105	0.121	0.079	0.095
	Rank	7	6	5	1	4	1	1
Ave rank	6.7	5.3	4.8	2.6	4.1	2.2	1	
Total rank	7	6	5	3	4	2	1	

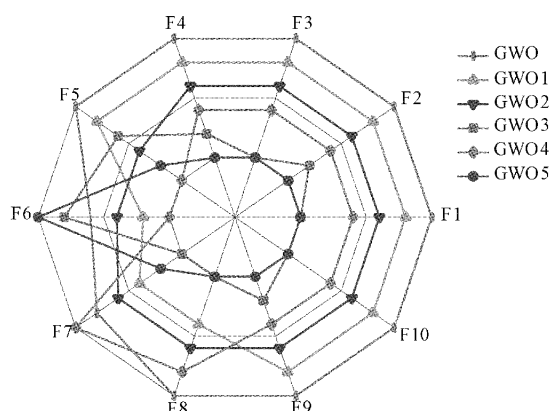


图 8 5 种改进策略均值排名雷达图

对初始种群和收敛因子采用 GWO1、GWO2 的策略,全方位的提升原 GWO 的寻优性能。

最后,观察标准差可以看出,IGWO 的鲁棒性表现最好。总体而言,观察排名 5 种单一改进策略相比原 GWO 都有所提高,但又次于综合改进的 IGWO,说明每个策略都发挥了作用,体现各阶段改进策略间的协作互补性,使得寻优性能和鲁棒性得到全面的提升。

3.3 IGWO 收敛性分析

根据表 2 数据绘制 IGWO 与单一改进策略的收敛曲线。如图 9(a)~(j)所示,反映出算法的收敛速度、收敛精度、稳定性以及算法逃离局部最优值的能力。

对于函数 F1、F2、F3、F4, GWO、GWO1、GWO2、GWO4 收敛速度与精度都较低;GWO3 与 GWO5 得益于欧氏距离权重的动态 α 、 β 、 δ 调整与迭代局部搜索强大的全局搜索能力,取得了较好的收敛精度与收敛速度,但仍无法收敛到理论最优值,而 IGWO 在结合 GWO3 与 GWO5 的策略后,不仅收敛精度明显提升,能寻优到 0,且在 500 代以内就能完成收敛,收敛速度显著加强;对于函数 F5,虽收敛速度与其他并无区别,但提升了些许收敛精度;对于大多数算法较难寻优的多峰函数 F6、F7、F9, IGWO 仍取得了较原 GWO 及其他策略算法相比,更优的收敛精度与速度,特别在收敛速度上,得益于莱维飞行较大的寻优范围,IGWO 在这三个函数上的初始收敛速度均远优于对比算法;在多峰函数 F8、F10,IGWO 不仅能跳出局部最优值寻优到理论最优值,且仅在 50 代以内就完成收敛,体现了 IGWO 卓越的探索能力。总体而言,反映了 IGWO 算法较优的收敛速度、精度及较好的逃离局部最优值的能力。

3.4 与其他群智能算法及改进算法对比

为了验证本文算法在求解优化问题时的寻优性能以及鲁棒性。将本文所提的 IGWO,与 GWO 算法、最新的金豹优化算法 (golden jackal optimization algorithm, GJO)^[16]、经典智能优化算法 (particle swarm optimization, PSO)^[17]、经典鲸鱼算法 (whale optimization algorithm, WOA)^[18]、最新改进的灰狼算法 (a bidirectional search

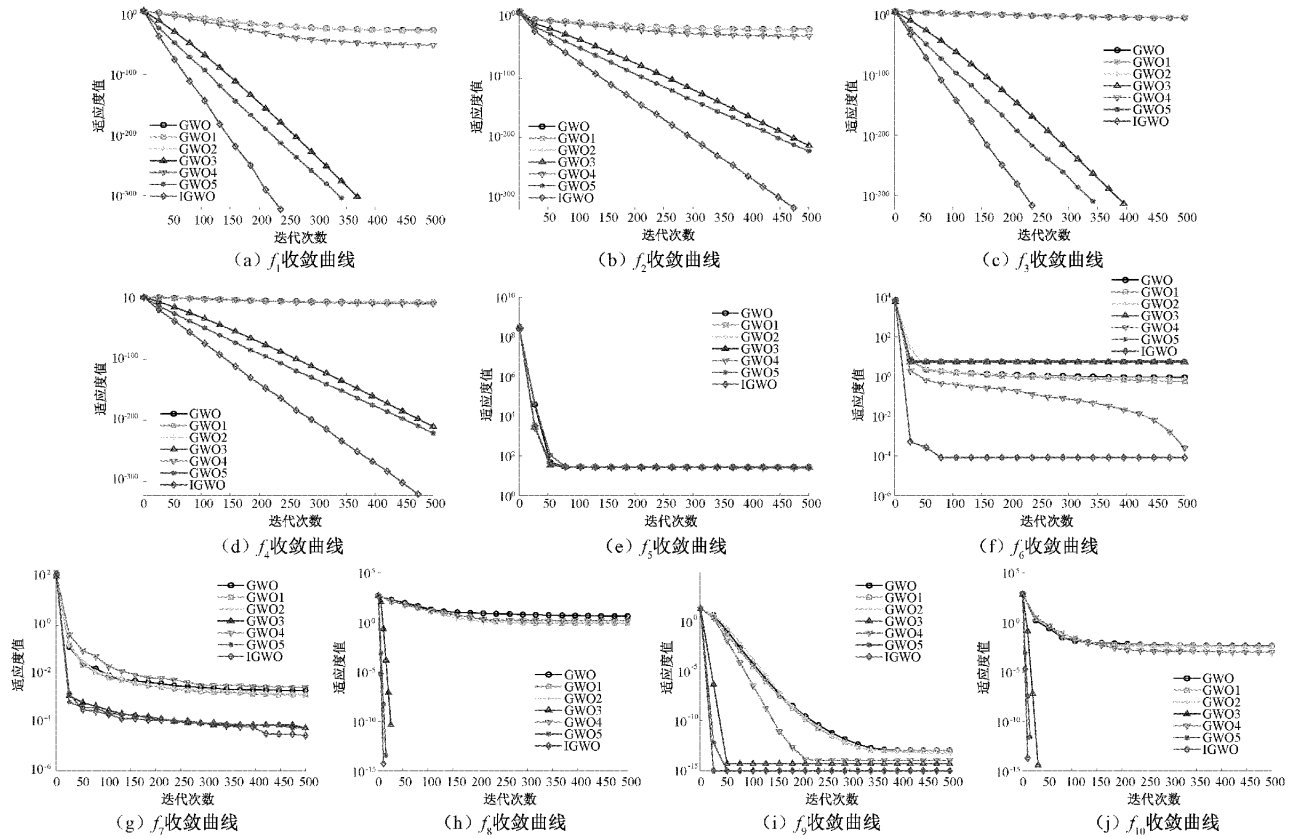


图 9 不同改进策略收敛曲线图

wolf optimizer based on stagnation detection, DBGWO)^[5]、经典改进灰狼算法(an improved grey wolf optimization algorithm, CGWO)^[4]共 6 种算法,在 30 维对上述 10 个基

准函数上进行寻优对比,其中引用文献[5]的实验数据,复现文献[4,16-18]实验,各算法主参数设置如表 3 所示,具体运行结果如表 4 所示,其中“/”为文献未进行此类测试。

表 3 主参数

算法	主参数
GWO	$a = (2 \rightarrow 0)$ 线性递减
GJO	$E_0 = rand(-1, 1)$ $E_1 = (1.5 \rightarrow 0)$ 线性递减
PSO	$V_{max} = 6$ $w = (0.9 \rightarrow 0.6)$ $c_1 = c_2 = 1.5$ 线性递减
WOA	$b = 1$
DBGWO	$a = (2 \rightarrow 0)$ $\mu = 0.01$ 非线性递减
CGWO	$a = (2 \rightarrow 0)$ 非线性递减
IGWO	$a_1 = (2 \rightarrow 0)$ $a_2 = (1 \rightarrow 0)$ 非线性递减 $\gamma_1 = 0.5, \gamma_2 = 300, \gamma_3 = 3, \lambda_1 = 0.5, \lambda_2 = 0.15, \lambda_3 = 0.5, a_{min} = 0$

由表 4 可知,在这 10 个基准测试函数的寻优实验中,本文所提出的 IGWO 的寻优效果明显优于其他 6 种对比算法,对于单峰函数 F1、F2、F3、F4, IGWO 算法都能找到理论最优值,且寻优精度和稳定性优于 PSO、GJO、WOA、DBGWO、CGWO;对于单峰函数 F5,虽然精度排名第二,与其他算法精度相差不大,但其标准差优于其他算法,说明 IGWO 算法较其他算法更加稳定;而对于多峰函数 F8、F10, IGWO 也能稳定地寻找到最优值,说明该算法的优越性;对于多峰函数 F6、F7、F9,虽然 IGWO 并未寻找到最优

值,但寻优精度和稳定性仍位列第一。在运行时间方面, IGWO 运行时间排名前列,说明改进策略的引进没有提升 GWO 算法的时间复杂度,反而降低了执行时间。综上分析表明,在寻优过程中 IGWO 的平均值和标准差明显优于其他对比算法(除函数 F5 中平均值劣于 DBGWO),说明改进后的灰狼算法具备较强的全局勘探和局部开发能力。

为了更加直观地观察每个算法在不同函数下的排序结果,采用雷达图对 Rank 结果进行绘图,如图 10 所示,图

中圆点曲线为 IGWO 结果曲线,其所围成的面积最小,说明 IGWO 整体上具有最好的寻优性能。

表 4 各算法求解测试函数的结果对比

函数	结果	GWO	GJO	PSO	WOA	DBGWO ^[5]	CGWO	IGWO
f_1	均值	1.48×10^{-27}	3.11×10^{-40}	1.81×10^1	1.49×10^{-73}	1.92×10^{-270}	0	0
	标准差	2.13×10^{-27}	1.19×10^{-39}	1.94	5.94×10^{-73}	0	0	0
	运行时间	0.130	0.221	0.207	0.032	/	0.092	0.081
	Rank	6	5	7	4	3	1	1
f_2	均值	5.94×10^{-17}	8.35×10^{-24}	1.79×10^1	9.44×10^{-51}	4.43×10^{-137}	1.51×10^{-186}	0
	标准差	1.88×10^{-17}	5.62×10^{-23}	1.42	4.93×10^{-50}	1.85×10^{-137}	3.62×10^{-186}	0
	运行时间	0.152	0.316	0.294	0.058	/	0.113	0.097
	Rank	6	5	7	4	3	2	1
f_3	均值	3.13×10^{-5}	3.94×10^{20}	1.08×10^2	4.24×10^4	9.52×10^{239}	0	0
	标准差	9.23×10^{-5}	1.87×10^{-21}	2.56×10^1	1.36×10^4	0	0	0
	运行时间	0.136	0.262	0.189	0.034	/	0.098	0.088
	Rank	5	4	6	7	3	1	1
f_4	均值	4.82×10^{-7}	2.63×10^{-16}	1.70	4.34×10^1	2.56×10^{-130}	4.23×10^{-182}	0
	标准差	3.79×10^{-6}	4.12×10^{-16}	1.24×10^{-1}	2.87×10^1	4.80×10^{-112}	1.57×10^{-181}	0
	运行时间	0.132	0.249	0.257	0.035	/	0.095	0.079
	Rank	5	4	6	7	3	2	1
f_5	均值	2.73×10^1	2.50×10^1	4.31×10^3	2.78×10^1	7.94	2.70×10^1	2.32×10^1
	标准差	9.36×10^{-1}	1.58×10^{-1}	1.07×10^3	4.92×10^{-1}	1.16×10^1	3.74×10^{-1}	1.35×10^{-1}
	运行时间	0.124	0.179	0.163	0.029	/	0.087	0.078
	Rank	5	3	7	6	1	4	2
f_6	均值	9.74×10^{-1}	1.67×10^{-1}	1.69×10^1	3.61×10^{-1}	1.83×10^{-4}	5.56	8.25×10^{-5}
	标准差	3.23×10^{-1}	4.61×10^{-1}	2.35	2.46×10^{-1}	1.01×10^{-4}	5.03×10^{-1}	1.37×10^{-5}
	运行时间	0.137	0.272	0.201	0.042	/	0.101	0.092
	Rank	5	3	7	4	2	6	1
f_7	均值	2.00×10^{-3}	6.91×10^{-4}	8.62×10^1	4.60×10^{-3}	1.40×10^{-3}	9.32×10^{-5}	2.90×10^{-5}
	标准差	1.10×10^{-3}	5.32×10^{-4}	2.02×10^1	5.50×10^{-3}	8.22×10^{-4}	7.44×10^{-5}	2.54×10^{-5}
	运行时间	0.201	0.482	0.451	0.105	/	0.140	0.126
	Rank	5	3	7	6	4	2	1
f_8	均值	4.47	8.11×10^{-1}	2.38×10^2	0	0	0	0
	标准差	3.72	1.35	2.01×10^1	0	0	0	0
	运行时间	0.139	0.265	0.219	0.045	/	0.108	0.092
	Rank	6	5	7	1	1	1	1
f_9	均值	1.05×10^{-13}	1.49×10^{-14}	4.54	4.44×10^{-15}	2.30×10^{-15}	4.32×10^{-15}	8.88×10^{-16}
	标准差	2.35×10^{-14}	7.52×10^{-15}	1.51×10^{-1}	2.47×10^{-15}	1.77×10^{-15}	6.49×10^{-16}	0
	运行时间	0.138	0.336	0.347	0.039	/	0.103	0.090
	Rank	6	5	7	4	2	3	1
f_{10}	均值	4.80×10^{-3}	5.24×10^{-2}	7.12×10^{-1}	3.60×10^{-3}	0	0	0
	标准差	7.20×10^{-3}	2.81×10^{-2}	6.54×10^{-2}	2.70×10^{-2}	0	0	0
	运行时间	0.155	0.251	0.219	0.073	/	0.105	0.095
	Rank	5	6	7	4	1	1	1

3.5 IGWO 种群寻优平衡性分析

为了直观的体现 IGWO 算法不同迭代时刻寻优能力

的变化,以分析其全局寻优能力和局部寻优能力的平衡性。以函数 F7 (Quartic) 为例,对 3 种算法进行比较。

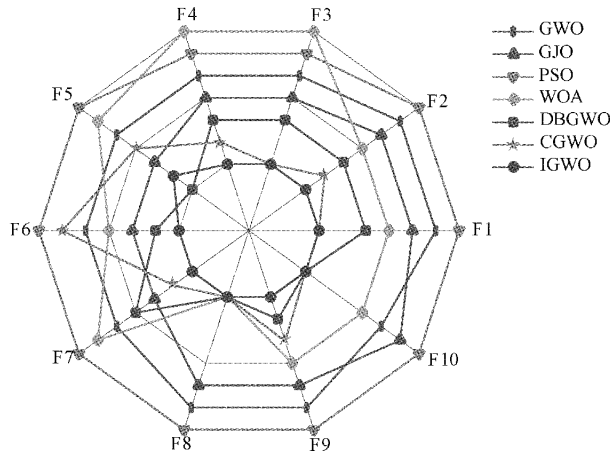


图 10 基准测试函数雷达图

图 11 是 3 种算法的种群全局勘探与局部开发比例图^[19]，通过种群中寻优个体的比例分布反映整个种群的搜索状态。根据图 11 可以看出，IGWO 的种群最后在全局勘探和局部开发间取得了良好的平衡。正常情况下，局部开发要趋近于 1，因为进化后期种群逐渐逼近最优解，整个解集

是集中在最优解附近的，因此后期主要是局部开发。此外，全局勘探和局部开发的百分比随着迭代的进行，应该有一个交替的过程：局部开发比例逐渐增加，最终接近 1；全局勘探比例逐渐减少，渐趋于 0。由图 11 也可以发现，IGWO 的收敛速度非常快，种群很快就靠近到最优解，同时根据文献[20]所述，勘探和开发在搜索过程中的占比分别为 10% 和 90% 且达到平衡时，算法有最佳性能。可以看出 IGWO 基本符合要求，种群最后勘探占比 10% 左右，开发占比 90% 左右。对比图 11(a)GWO, IGWO 的两种比例稳定在 1:9，而 GWO 渐趋于 0；对比图 11(b)WOA, IGWO 两种占比随迭代过程平稳进行，而且前期受收敛因子和莱维飞行策略的影响，前期的比例在一个大区间里进行着多次较大的扰动，便于快速勘探至全局最优，后期为了跳出局部极值，两种比例稳定在 1:9 之间，并且有着稳定的扰动频率，而 WOA 的两种勘探比前期递减过快，全局勘探比例急剧减少，不利于找到全局最优值，缺乏较好的稳定性与扰动频率。

综上所述，IGWO 种群在全局不同迭代时刻保持稳定地寻优能力与平衡性，更利于算法寻优。

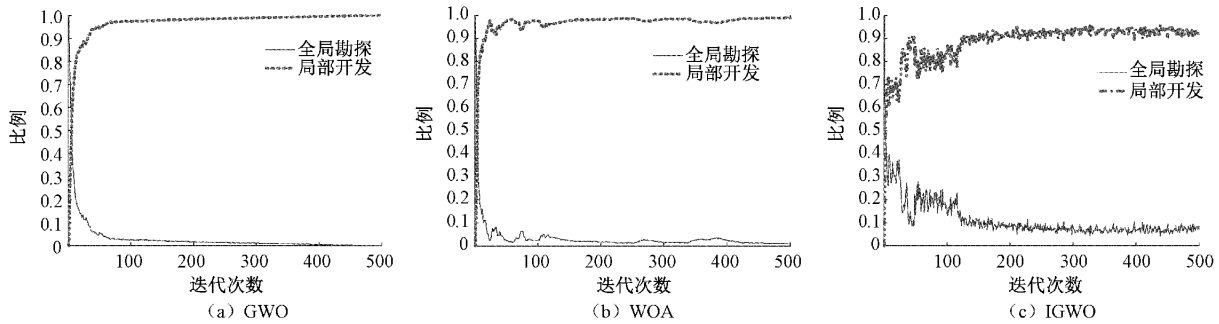


图 11 全局勘探与局部开发比例图

4 IGWO 算法工程算例应用分析

近年来，工程算例应用作为优化问题中的一个热门领域，受到广泛的关注。为了进一步体现 IGWO 在处理实际工程应用案例的优越性，采用文献[21]中的罚函数作为约束条件，对焊接梁设计问题进行算法优化，并将 IGWO 优化结果与灰狼优化(GWO)算法、金豹优化算法(GJO)算法、鲸鱼优化算法(WOA)、基于非线性控制参数组合策略的灰狼优化算法(improved grey wolf algorithm based on nonlinear control parameter combination strategy, NCPGWO)^[22]、基于翻筋斗觅食策略的灰狼优化算法(disturbance and somersault foraging-grey wolf optimization, DSFGWO)^[23]以及文献[21]中其他算法进行对比。

4.1 焊接梁设计问题

焊接梁设计问题是一个最小化问题，其中优化算法是为降低设计制造成本 $f(x)$ ，即焊缝宽度(h)、横梁宽度(d)、长

度(l)、厚度(b)，使得制造焊接梁的费用最小：

$$x = [x_1 x_2 x_3 x_4] = [h l t b] \quad (28)$$

$$f(x) = 1.104 71 \cdot x_1^2 \cdot x_2 + 0.048 11 \cdot x_3 \cdot x_4 \cdot (14.0 + x_2) \quad (29)$$

$$\begin{aligned} g_1(X) &= \tau(x) - \tau_{\max} \leq 0 & g_2(X) &= \sigma(x) - \sigma_{\max} \leq 0 \\ g_3(X) &= \delta(x) - \delta_{\max} \leq 0 & g_4(X) &= x_1 - x_4 \leq 0 \end{aligned} \quad (30)$$

$$g_5(X) = P - P_c \leq 0 \quad g_6(X) = 0.125 - x_1 \leq 0 \quad (31)$$

$$g_7(X) = 1.104 71 \cdot x_1^2 + 0.048 11 \cdot x_3 \cdot x_4 \cdot (14.0 + x_2) - 5.0 \leq 0 \quad (32)$$

4.2 测试结果与分析

8 种算法求解结果如表 5，其中 IGWO 平均值最优，寻优性能最好；由图 12 可以看出 IGWO 具有较快的收敛速度和跳出局部极值的能力，进一步证实 IGWO 的实用性。

表5 焊接梁优化问题平均解

算法	h	l	t	b	平均值
GWO	0.202 3	3.489 1	9.054 2	0.215 4	1.726 5
GJO	0.563 6	0.406 2	0.343 1	1.378 7	2.128 3
WOA	1.393 3	1.152 7	4.811 6	1.149 9	2.155 7
HSSAHHO ^[21]	0.205 7	3.470 5	9.036 7	0.205 7	1.724 8
SCHOA ^[21]	0.205 7	3.470 5	9.030 6	0.205 6	1.722 9
NCPGWO	0.198 6	3.374 2	9.197 1	0.199 1	1.675 9
DSFGWO	0.198 3	3.346 5	9.195 4	0.198 8	1.673 2
IGWO	0.198 8	3.337 6	9.192 1	0.198 5	1.672 0

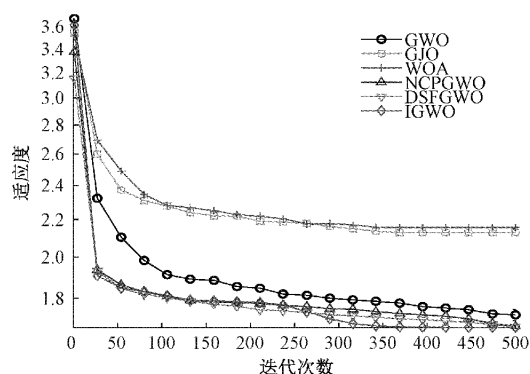


图12 焊接梁优化问题各算法收敛曲线

5 结 论

针对标准灰狼算法(GWO)收敛速度慢,易陷入局部最优等缺点,本文通过佳点集理论初始化种群,提高算法搜索速率与稳定性;采用双收敛因子,收敛因子基于种群位置非线性自适应更新,在种群迭代全期平衡全局勘探与局部搜索能力;采用改进的 ILS 策略协调更新机制,并引入莱维飞行与欧氏动态权重策略,便于种群跳出局部最优值,提升寻优准确率。随后通过 10 个基准测试函数的仿真分析及种群寻优平衡性对比,结果表明 IGWO 具有更优的寻优精度、稳定性及收敛速度,最后将 IGWO 应用于工程优化问题,进一步验证了 IGWO 的优越性。

在未来研究中,会进一步加强对于灰狼优化算法的改进和实验,提高该算法针对更高维复杂问题的适用性,并将其应用到机器学习参数优化和可靠性仿真评价等领域,进一步落实其实际应用。

参考文献

[1] MIRJALILI S, MIRJALILI S M, LEWIS A D. Grey wolf optimizer[J]. *Advances in Engineering Software*, 2014, 69:46-61.

[2] 滕志军,吕金玲,郭力文,等.一种基于 Tent 映射的混合灰狼优化的改进算法[J]. *哈尔滨工业大学学报*, 2018, 50(11):40-49.

[3] 王海峰,李萍,王博,等.灰狼算法优化 BP 神经网络的

图像去模糊复原[J]. *液晶与显示*, 2019, 34(10): 992-999.

[4] 王秋萍,王梦娜,王晓峰.改进收敛因子和比例权重的灰狼优化算法[J]. *计算机工程与应用*, 2019, 55(21): 60-65,98.

[5] 张大明,徐嘉庆,赵彦清,等.基于停滞检测的双向搜索灰狼优化算法[J]. *计算机应用研究*, 2022, 39(6): 1725-1730.

[6] HEIDARI A A, PAHLAVANI P. An efficient modified grey wolf optimizer with lévy flight for optimization tasks [J]. *Applied Soft Computing*, 2017, 60:115-134.

[7] 曾蝶,陈立万,赵尚飞,等.多策略灰狼算法在 WSN 上的覆盖优化研究[J]. *电子测量技术*, 2023, 46(7): 45-52.

[8] 刘紫燕,吴应雨,梁静,等.基于杂交策略的自适应灰狼优化算法[J]. *计算机应用研究*, 2022, 39(1):113-117.

[9] 崔琳,张熠鑫.改进鸡群优化算法的二维 MUSIC 谱峰搜索研究[J]. *电子测量与仪器学报*, 2020, 34(3): 142-148.

[10] 张孟健,张浩,陈曦,等.基于 Cubic 映射的灰狼优化算法及应用[J]. *计算机工程与科学*, 2021, 43(11): 2035-2042.

[11] 陈立万,曾蝶,赵尚飞,等.基于 EGWOEO 算法的三维无线传感网络覆盖优化[J]. *电子测量技术*, 2023, 46(4):25-34.

[12] YANG X S, DEB S. Multiobjective cuckoo search for design optimization [J]. *Computers & Operations Research*, 2013, 40(6):1616-1624.

[13] 吴忠强,杜春奇,李峰,等.基于蝙蝠算法的永磁同步电机健康状态监测[J]. *仪器仪表学报*, 2017, 38(3): 695-702.

[14] MOLINA D, LATORRE A, HERRERA F. SHADE with iterative local search for large-scale global optimization [C]. 2018 IEEE Congress on Evolutionary Computation(CEC), IEEE, 2018:1-8.

[15] 汪超,王丙柱,岑豫皖,等.基于多样性全局最优引导和

- 反向学习的离子运动算法[J]. 控制与决策, 2020, 35(7):1584-1596.
- [16] CHOPRA N, ANSARI M M. Golden jackal optimization: A novel nature-inspired optimizer for engineering applications [J]. Expert Systems with Application, 2022, 198:116924.
- [17] CHEGINI S N, BAGHERI A. PSOSCALF: A new hybrid PSO based on Sine Cosine Algorithm and Levy flight for solving optimization problems[J]. Applied Soft Computing, 2018, 73:697-726.
- [18] MIRJALILI S, LEWIS A. The whale optimization algorithm [J]. Advances in Engineering Software, 2016, 95: 51-67.
- [19] HUSSAIN K, SALLEH M N M, CHENG S, et al. On the exploration and exploitation in popular swarm-based metaheuristic algorithms[J]. Neural Computing and Applications, 2019, 31: 7665-7683.
- [20] MORALES-CASTAÑEDA B, ZALDIVAR D, CUEVAS E, et al. A better balance in metaheuristic algorithms: Does it exist? [J]. Swarm and Evolutionary Computation, 2020, 54: 100671.
- [21] KAUR M, KAUR R, SINGH N, et al. SChoA: A newly fusion of sine and cosine with chimp optimization algorithm for HLS of datapaths in digital filters and engineering applications [J]. Engineering with Computers, 2021, 42:1-29.
- [22] 张孟健, 龙道银, 杨小柳, 等. 基于非线性控制参数组合策略的灰狼优化算法[J]. 计算机应用与软件, 2021, 38(5):250-255,322.
- [23] 王正通, 程凤芹, 尤文, 等. 基于翻筋斗觅食策略的灰狼优化算法 [J]. 计算机应用研究, 2021, 38(5): 1434-1437.

作者简介

文昌俊, 博士, 教授, 主要研究方向为质量控制与工程应用。

陈凡, 硕士研究生, 主要研究方向为群智能优化算法与应用。

E-mail: 1047799142@qq.com

陈洋洋, 硕士研究生, 主要研究方向为群智能优化算法与可靠性分析。

何永豪, 硕士研究生, 主要研究方向为群智能优化算法与农机控制。