

DOI:10.11918/202111069

增强全局搜索和自适应蜉蝣算法

王义,张达敏,邹诚诚

(贵州大学 大数据与信息工程学院,贵阳 550025)

摘要: 针对蜉蝣算法全局搜索能力较差和自适应能力弱等问题,提出一种增强全局搜索能力和自适应的蜉蝣算法——MIWMA。首先引入非均匀高斯变异策略对雄性蜉蝣和雌性蜉蝣进行位置更新,对全局最优位置变异引导其他个体向优良位置靠近,促使种群具有一定指导,从而提升全局搜索能力和增强种群多样性;其次,引入不完全伽马函数与Beta累加分布的自适应惯性权重对全局搜索和开发能力建立更好的平衡,平衡种群的全局搜索和局部搜索能力,进而提升算法收敛精度,利于种群全局搜索寻找最优解的潜力;引入局部停滞对抗策略,根据迭代停滞情况,调节蜉蝣速度更新的惯性部分和社会部分,使之具有最优搜索状态,增强算法全局搜索能力。利用经典测试函数集和IEEE CEC2021测试竞赛集进行测试优化比较,验证算法的有效性和稳健性,并利用Friedman和Wilcoxon秩和检验,分析表明:提出的算法有更好的稳定性、鲁棒性和可靠性。最后运用两个工程难题进行优化,结果验证了该算法在工程优化问题上的适用性,适合求解需求高精度的优化难题。

关键词: 蜉蝣算法;全局搜索;非均匀高斯变异;种群多样性;停滞策略

中图分类号: TP301.6 文献标志码: A 文章编号: 0367-6234(2022)11-0137-14

Enhance global search and adaptive mayfly algorithm

WANG Yi, ZHANG Damin, ZOU Chengcheng

(College of Big Data and Information Engineering, Guizhou University, Guiyang 550025, China)

Abstract: In view of the problems of poor global search ability and weak adaptive ability of mayfly algorithm, an enhance global search and adaptive mayfly algorithm (MIWMA) was proposed. Firstly, the non-uniform Gaussian mutation strategy was adopted to update the position of male mayfly and female mayfly, guide the global optimal position mutation to leading other individuals to approach the good position, and promote the population to have certain guidance, so as to improve the global search ability and enhance the diversity of the population. Secondly, the adaptive inertia weight of incomplete gamma function and beta cumulative distribution was introduced to establish a better balance between the global search and development ability, regulate the global search and local search ability of the population, and then improve the convergence accuracy of the algorithm, which is conducive to the potential of the global search of the population to find the optimal solution. The local stagnation countermeasure strategy was introduced. On the basis of the iterative stagnation, the inertia part and social part of mayfly speed update were adjusted to make it have the optimal search state and enhance the global search ability of the algorithm. The classic test function set and IEEE CEC2021 test competition set were used for test optimization comparison to verify the effectiveness and robustness of the algorithm. Results show that the proposed algorithm had better stability, robustness, and reliability by using Friedman and Wilcoxon rank sum test. Finally, two engineering problems were used for optimization. The results verified the applicability of the algorithm in engineering optimization problems and are suitable for solving optimization problems requiring high precision.

Keywords: mayfly algorithm; global search; non-uniform Gaussian mutation; population diversity; stagnation strategy

近年来,大量基于元启发的仿生优化算法用来解决科学领域的各种优化难题并有较好的优化效果,如工程难题、商业最优问题、经济学和复杂多目标优化问题。元启发算法因其灵活、结构简单、无梯

度机制等优点受到研究者的关注^[1]。近年来出现各种基于仿生优化算法和基于物理模型的算法,如基于模仿自然界中的鸟群和鱼群的觅食行为提出了粒子群算法(particle swarm optimization, PSO)^[2]、受麻雀的觅食行为和反捕食行为的启发提出麻雀搜索算法(sparrow search algorithm, SSA)^[3]、受狼群的等级领导行为和守猎过程的启发提出了灰狼算法(grey wolf optimization, GWO)^[4]、基于裸鼯鼠和繁育者的繁殖行为,培育精英后代提出裸鼯鼠算法

收稿日期: 2021-11-14
基金项目: 国家自然科学基金(62062021, 61872034);
贵州省科学技术基金(黔科合基础[2020]1Y254)
作者简介: 王义(1997—),男,硕士研究生;
张达敏(1967—),男,教授,硕士生导师
通信作者: 张达敏, 1203813362@qq.com

(naked mole-rat algorithm, NMRA)^[5]、模拟自然界中成虫发光萤火虫聚集行为提出萤火虫算法 (firefly algorithm, FA)^[6]、模拟生物在自然环境的遗传和进化过程提出遗传算法 (genetic algorithm, GA)、受领导者引导和跟随者跟进共同在海洋中和寻食启发提出樽海鞘群算法 (salp swarm algorithm, SSA)^[1] 以及 2021 年根据变色龙群体在搜索食物、360°眼睛转动和捕猎行为提出变色龙群算法 (chameleon swarm algorithm, CSA)^[7] 等。

2020 年,文献[8]根据雌雄蜉蝣群体的聚集、婚配行为和交配繁殖行为提出一种基于群体仿生优化算法——蜉蝣算法 (mayflies algorithm, MA)^[8],其灵感来自蜉蝣群体的社会行为,尤其是它们的交配过程,雄雌性蜉蝣成熟后婚配及求偶和繁殖而启发。从 MA 的结构表达形式上看,是由 PSO、GA 和 FA 算法的结合改进,由此 MA 也符合优胜劣汰规律的种群进化算法,具有较好的局部搜索性能和良好的种群多样性。遗憾的是,MA 虽具备上述良好的性能和优势,但前期的全局搜索能力不理想导致后期即使具备较好的开发能力也达不到理想求解精度和算法稳定性,且易陷入局部最优^[8]。但提出至今,在工程优化设计、车间调度和动力系统中的应用已有相关报道,说明 MA 在实际应用和求解上具备不同优越性能。MA 和其他群体智能算法一样,依然存在局部最优、搜索和开发之间得不到平衡和寻优精度低下,在实际求解问题时导致精度低下,求解结果不稳定等缺点。

近年,针对其他形式的群体智能算法的缺点,有不少研究者提出不同策略进行优化改进,改善其全局寻优性能和求解的稳定性;文献[9]利用 Circle 映射初始化鲸鱼优化算法的种群,使种群初始化过程中有更好的遍历性和降低群体的盲目性,且引入反向学习策略对鲸鱼的位置求出反向解,提高了鲸鱼优化算法的搜索能力和求解的稳定性;文献[10]将樽海鞘链按照不同的适应度划分成 3 个子群,引入敏感性分析、共生策略和非均匀高斯变异对樽海鞘链的 3 个子群针对性改进,结果表明改进的樽海鞘群体有更好的引导作用和较强的多样性,提升了算法的性能;文献[11]引入黄金正弦机制和自适应惯性权重,改善蝴蝶优化算法的全局搜索能力和搜索的自适应能力;文献[12]将速度边界和能量边界的方法引入粒子群算法,证明出粒子群的惯性部分和社会部分间建立了良好的平衡,结果表明它能跳出局部最优,具有更好的搜索能力。

为改善 MA 的全局寻优能力、算法的稳定性能和探索搜索的自适应性,提出增强全局搜索和自适

应的蜉蝣算法 (MIWMA),补偿算法的劣势保留已有优势结合,提高搜索能力和稳定性能,更好建立 MA 存在的搜索和开发之间得不到平衡的问题。为增强蜉蝣种群的多样性,引入非均匀变异和全局最优位置更新策略,同时不放弃适应度差的个体,并指引其向全局最优位置靠近,增强了蜉蝣群体搜索的多样性,提升算法跳出局部最优的能力;引入不完全伽马函数和 Beta 累加分布惯性权重,使其在已提出的惯性权重基础上能自适应动态调节惯性系数,增强全局搜索和开发之间的平衡,同时其自适应动态搜索能力得到提升,进一步提高了算法的搜索精度;引入局部停滞策略调节原有的搜索特性,在惯性部分和社会部分间进行调节,使蜉蝣群体间进行信息交流共享和合作,增强算法搜索能力。

1 蜉蝣算法

蜉蝣算法受雄性蜉蝣与雌性蜉蝣婚配的社会吸引行为启发的优化算法。其寻优过程为:雄性蜉蝣个体的寻优、雌性蜉蝣个体的寻优和优异个体的雌雄交配。设蜉蝣在 d 维空间的位置表示为 $X = (x_1, x_2, \dots, x_d)$, 其 d 维空间的速度表示为 $V = (v_1, v_2, \dots, v_d)$ 。

1.1 雄性蜉蝣的运动

当雄性蜉蝣投放在一个固定的区域内会发生聚集行为,向着群体中心位置靠近。蜉蝣个体的位置是按照自身经验或邻近个体的行为调节。用 x_{ij}^t 表示个体 i 在 j 维空间的第 t 次迭代的位置, v_{ij}^t 则表示个体 i 在 j 维空间的第 t 次迭代的速度。蜉蝣会向中心聚集不断地移动和在水面上舞蹈训练,蜉蝣的速度更新如下:

$$v_{ij}^{t+1} = \begin{cases} v_{ij}^t + a_1 e^{-\beta r_p^2} (p_j - x_{ij}^t) + \\ a_2 e^{-\beta r_g^2} (g_j - x_{ij}^t), f(y_i) > f(x_i) \\ v_{ij}^t + d \cdot r, f(y_i) \leq f(x_i) \end{cases} \quad (1)$$

式中: v_{ij}^{t+1} 是雄性蜉蝣 i 在 j 维空间上 $t+1$ 次迭代的速度, a_1 和 a_2 为雄性蜉蝣移动行为的吸引系数, p 为先前迭代的最好位置, g 是全局最优位置, β 为能见度系数,其目的是控制蜉蝣的能见范围。 d 代表舞蹈系数,不断吸引异性靠近它。 r 是一个随机系数,取值为 $[-1, 1]$ 。 r_p 和 r_g 分别为当前位置与 p 的距离和当前位置与 g 的距离。

由上, r_p 和 r_g 是最优位置与当前位置的距离,其距离的计算用欧氏距离表示,欧氏距离计算公式如下:

$$\|x_i - X_i\| = \sqrt{\sum_{j=1}^n (x_{ij} - X_{ij})^2} \quad (2)$$

式中: x_{ij} 为蜉蟥 i 的 j 维, X_i 为 p_i 或 g_i 的位置。

雄性蜉蟥的位置更新为

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (3)$$

式中: v_i^{t+1} 是雄性蜉蟥 i 的第 $t+1$ 次迭代的速度, y_i^t 为蜉蟥 i 的第 t 次迭代的位置。

1.2 雌性蜉蟥的运动

雌性蜉蟥与雄性蜉蟥相比的显著特点是雄性蜉蟥投放在固定区域内会发现聚集行为而雌性蜉蟥则不会, 雌性蜉蟥会与雄性蜉蟥靠近完成繁殖。雌性蜉蟥的位置更新如下:

$$y_i^{t+1} = y_i^t + v_i^{t+1} \quad (4)$$

雌性蜉蟥的速度更新表达如下:

$$v_{ij}^{t+1} = \begin{cases} v_{ij}^t + a_2 e^{-\beta r_m^2} (x_{ij}^t - y_{ij}^t), & f(y_i) > f(x_i) \\ v_{ij}^t + l_n \cdot r, & f(y_i) \leq f(x_i) \end{cases} \quad (5)$$

式中: v_{ij}^t 为蜉蟥 i 在 j 维空间上的速度, y_{ij}^t 为蜉蟥 i 在 j 维空间上的位置。 a_2 为雄雌蜉蟥吸引系数, β 为能见度系数, r_m 为雄性蜉蟥和雌性蜉蟥的距离。 l_n 是随机游走系数, r 为随机数, 取值为 $[-1, 1]$ 。

1.3 雌雄蜉蟥的交配

蜉蟥的交配过程可用交叉算子表示, 从雄性蜉蟥中选择一个父本, 雌性蜉蟥选取母本。父本选择的方式与雄雌吸引的方式一致。优胜劣汰机制, 将最优个体的雄性和雌性蜉蟥进行繁殖得到最优个体, 依次类推, 得到的两个子代表达式为

$$S_{\text{offspring1}} = L \cdot S_{\text{male}} + (1 - L) \cdot S_{\text{female}} \quad (6)$$

$$S_{\text{offspring2}} = (1 - L) \cdot S_{\text{male}} + L \cdot S_{\text{female}} \quad (7)$$

式中: L 为服从高斯分布的随机数, 取值范围为 $L \in [0, 1]$; S_{male} 是父本, S_{female} 是母本。

2 增强全局搜索和自适应权重的蜉蟥算法

2.1 非均匀变异下的位置更新策略

2.1.1 引入上一次位置更新指引策略

蜉蟥群体每经一次迭代更新, 都会保留历史的最好位置。历史最优位置可为后续搜索提供新方向, 新位置既受历史最优位置的指引又受上一次迭代位置的影响, 可有效避免算法易陷入局部最优的缺陷, 提高算法的搜索能力。引入历史位置和上一次指引的位置更新表达式为

$$x_i^t = x_i^{t-1} + c \cdot (g - x_i^{t-1}) \quad (8)$$

式中: t 为当前迭代次数, x_i^t 为当前迭代的位置, x_i^{t-1} 为上一次迭代位置, g 为历史迭代的全局最优位置。 c 为取值 $[0, 1]$ 之间随机数。

2.1.2 非均匀的高斯变异

由式(8)可看出, 历史迭代的最优位置有可能不是目标优化中的最优位置, 完全按照历史最优位置去搜索, 将会导致陷入局部最优达不到搜索效果, 多样性得不到保证。针对这一缺陷, 对全局最优位置 g 变异, 增强搜索多样性, 使全局搜索能力得到进一步增强, 利用一种非均匀随机高斯变异策略。非均匀随机高斯变异策略的表达式如下:

$$\Delta(t, N) = N(1 - r^{(1 - \frac{t}{T_{\max}})^2}) \quad (9)$$

$$N \sim N((g - x_i^{t-1}), \sigma) \quad (10)$$

$$\sigma = \sigma_0^2 e^{-\frac{t}{T_{\max}}} \quad (11)$$

式中: $T_{\max}$ 为最大迭代次数, Δ 为非均匀高斯变异的步长, N 服从参数的高斯分布, σ 为高斯分布的标准差。 σ_0^2 初始迭代方差, 用来控制变异范围, r 为随机数。结合式(9)~(11)对雄性群体和雌性群体的位置变异更新, 新的位置更新表达式为

$$x_i^t = x_i^{t-1} + c(\Delta \cdot g_x - x_i^{t-1}) \quad (12)$$

$$y_i^t = y_i^{t-1} + c(\Delta \cdot g_y - y_i^{t-1}) \quad (13)$$

引入非均匀高斯变异下的位置更新策略有以下特点: 在迭代前期, Δ 相较于后期较大, 引入位置更新后, 前期 g 有较强的变异效果, 使其向上一次迭代的位置引导向 g 靠近, 能增强种群的多样性和前期具有较强的全局寻优能力; 前期具备较强的全局寻优能力, 其 g 与 x_i^{t-1} 位置距离较大, 同时保持前期较大的方差, 生成的高斯分布随机数增强变异步长和策略的寻优性能; 后期, g 与 x_i^{t-1} 位置距离缩小, 同时保持方差逐渐变小, 搜索趋于稳定, 蜉蟥向局部最优位置精细搜索增强开发能力。

2.2 不完全伽马函数和 Beta 累加分布权重

2.2.1 非线性惯性权重指引策略

惯性权重因子对算法的搜索能力和开发能力具有一定的指导作用, 体现出雄性蜉蟥和雌性蜉蟥能够借鉴一定先验行为的能力。文献[13]指出, 较大的惯性权重具有良好的全局搜索能力; 较小的惯性权重具有较好的开发能力, 使其能得到更好局部搜索能力。引入一种非线性递减的自适应惯性权重, 惯性权重因子的表达式如下:

$$\omega(t) = \left(1 - \frac{t}{T_{\max}}\right)^{\frac{\alpha}{\sqrt{T_{\max}}}} \quad (14)$$

式中: t 为当前的迭代次数, $T_{\max}$ 为最大迭代次数, α 为惯性权重的控制参数。即不同的 α 值对 MA 的搜索具有一定的影响, 图 1 为 $\alpha = 1.0, 1.5, 2.0, 2.5, 3.0, 3.5$, 迭代 500 次的 ω 权重变化。

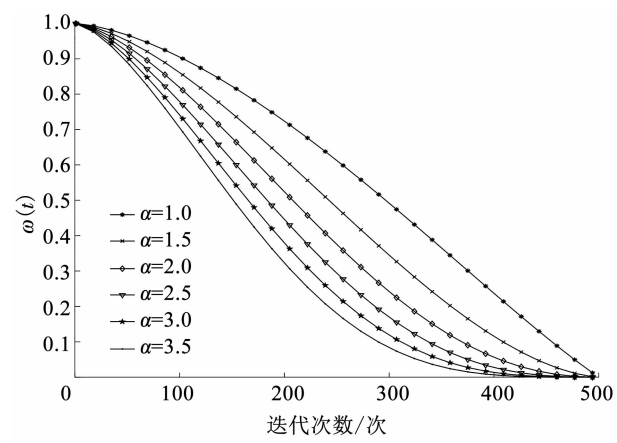


图 1 不同参数的惯性权重

Fig. 1 Inertial weights of different parameters

2.2.2 融合不完全伽马函数和 Beta 累加分布函数

上述 $\omega(t)$ 具有确定性的数学表达,即整个搜索和开发过程会按照 $\omega(t)$ 的关系执行,容易按照一定规律搜索,算法同样会进行局部最优的可能,做不到真正的自适应机制。为使 ω 能进行自适应动态调整,引入不完全伽马函数^[14]和 Beta 分布累加函数^[15],其新的数学表达式如下:

$$\omega'(t) = \frac{1}{\lambda} \omega(t) \cdot \Gamma\left(\lambda, 1 - \frac{t}{T_{\max}}\right) + \nu B(\sigma; b_1, b_2)$$

(15)

$$\Gamma(\lambda, u) = \int_0^{\lambda} e^{-t} t^{u-1} dt$$

(16)

$$B(\sigma; b_1, b_2) = \frac{1}{B(b_1, b_2)} \int_0^{\sigma} t^{b_1-1} (1-t)^{b_2-1} dt$$

(17)

式中: $\Gamma(\lambda, u)$ 为不完全的伽马函数, $\lambda > 0$ 的随机变量,取值为 $\lambda = 0.1$; ν 为偏移调整因子,调整 ω 的偏移程度,取 $\nu = 0.1$; $B(\sigma; b_1, b_2)$ 为服从 Beta 分布的累加分布函数的逆函数,经大量实验结果取 $b_1 = 1$, $b_2 = 2$, σ 取值与式(11)一致。

引入不完全伽马函数和 Beta 累加分布函数的自适应惯性权重,雄性蜉蝣与雌性的位置更新表达式为

$$x_i^{t+1} = \omega'(t) \cdot x_i^t + v_i^{t+1}$$

(18)

$$y_i^{t+1} = \omega'(t) \cdot y_i^t + v_i^{t+1}$$

(19)

为了选取一个合适的 α 值并探究引入不完全伽马函数和 Beta 累加分布的惯性权重优势,利用经典测试函数 f_1 对不同参数和不同权重独立运行 30 次,实验迭代 500 次,维度为 30,运行的结果见表 1。

由表 1 结果可知,参数 α 的选取直接影响算法寻优能力。当 $\alpha = 1$ 时,MA 具有较好的搜索能力,因此本文取 $\alpha = 1$ 。同样,表 1 是 ω 和 ω' 在不同参数和同一参数下的对比结果,在同样的 α 的取值下,

ω' 具有更好的收敛精度,目标求解的性能更好。说明引入的不完全伽马函数和 Beta 累加分布的惯性权重优势较 ω 更明显,与 ω 相比具有更好的自适应动态调整能力。

表 1 不同 α 值和适应权重的搜索结果

Tab. 1 Search results with different α values and adaptive weights

α	ω		ω'	
	平均值	标准差	平均值	标准差
1.0	6.23×10^{-72}	1.11×10^{-71}	1.52×10^{-80}	6.79×10^{-78}
1.5	5.62×10^{-69}	3.23×10^{-68}	5.26×10^{-68}	6.41×10^{-70}
2.0	9.57×10^{-69}	7.53×10^{-69}	1.02×10^{-78}	5.92×10^{-78}
2.5	1.33×10^{-65}	2.31×10^{-64}	9.55×10^{-78}	7.41×10^{-78}
3.0	8.28×10^{-61}	5.52×10^{-61}	8.31×10^{-76}	3.80×10^{-76}
3.5	5.52×10^{-59}	4.32×10^{-60}	2.11×10^{-75}	3.21×10^{-76}

注:加粗为最优值。

2.3 局部停滞对抗策略

在基本 MA 表达式中,社会部分中雄性蜉蝣和雌性蜉蝣的速度更新总会满足以下关系:

$$v_i = a_i e^{-\beta r^2} (p_j - p_i)$$

(20)

式中若 p_j 和 p_i 相邻很近,其欧氏距离 r 就很小,由于式(20)中含有指数关系式,呈现指数衰减慢,但后面的距离非常小;同理, p_j 和 p_i 相邻较远,指数衰减严重,因此社会部分难以达到平衡。针对这一缺陷,对蜉蝣的速度更新机制进行改进。

首先,引入监测搜索停滞系数^[12],用于控制是否出现局部迭代收敛的情况,停滞系数为

$$C_t = \min\left\{1, \max\left\{\frac{t - \psi_t}{\tilde{T}} - 1, 0\right\}\right\}$$

(21)

式中: $\tilde{T}$ 为迭代停滞步长的阈值,本文取 $\tilde{T} = 50$ 。 t 为当前迭代的次数, ψ_t 为最大的 $s \leq t$, 即第 s 与 $s-1$ 次迭代的适应度不同而到当前迭代的 t 却相同的 ψ_t , 或速度的平均均方速度超过速度阈值的值^[12], 即

$$\frac{1}{d_n} \sum_{i=1}^n \|v_i(s)\|^2 \geq v_{\max}$$

定义平衡算子表达式为

$$\eta = \eta_- C_t + \eta_+ (1 - C_t)$$

(22)

式中: η 为平衡调节算子,调节蜉蝣的速度更新,根据 C_t 选择出合适的调节值调整惯性部分和社会认知部分,加强群体间的信息交流协作。取 $\eta_+ = 1$ 和 $\eta_- = 0.1$ 。引入调节算子,雄性蜉蝣的速度表达式为

$$v_{ij}^{t+1} = \begin{cases} (1 - \omega' \eta) v_{ij}^t + \eta a_2 e^{-\beta r_{ij}^2} (x_{ij}^t - y_{ij}^t), & f(y_i) > f(x_i) \\ (1 - \omega' \eta) v_{ij}^t + \eta \cdot l_{ij} \cdot r, & f(y_i) \leq f(x_i) \end{cases}$$

(23)

雄性蜉蝣的速度表达式为

$$v_{ij}^{t+1} = \begin{cases} (1 - \omega' \eta) v_{ij}^t + \eta a_1 e^{-\beta r_p^2} (p_j - x_{ij}^t) + \\ \eta a_2 e^{-\beta r_g^2} (g_j - x_{ij}^t), f(y_i) > f(x_i) \\ (1 - \omega' \eta) v_{ij}^t + \eta \cdot d \cdot r, f(y_i) \leq f(x_i) \end{cases} \quad (24)$$

式中 ω' 为式(15)自适应惯性权重。在式(20)的监测搜索停滞系数,判断是否存在搜索停滞,利用式(21)得出停滞的步长值。由式(22)和式(23)知,蜉蝣的速度更新具有惯性部分和社会部分。惯性部分用于调节蜉蝣自身的状态,遵循前期较强的全局

搜索能力及后期的开发能力,使其随迭代的增加保持平衡;社会部分则是蜉蝣群体的信息交流和吸引。当蜉蝣搜索状态陷入局部最优时,按照调节算子平衡惯性部分和社会部分,使其调节自身状态和吸引、信息交流间合作,快速跳出局部最优,增强算法的搜索能力。

2.4 MIWMA 算法流程图及复杂度分析

2.4.1 MIWMA 算法流程图

综合上述引入的 3 种策略,文中提出的 MIWMA 算法的流程图和步骤见图 2。

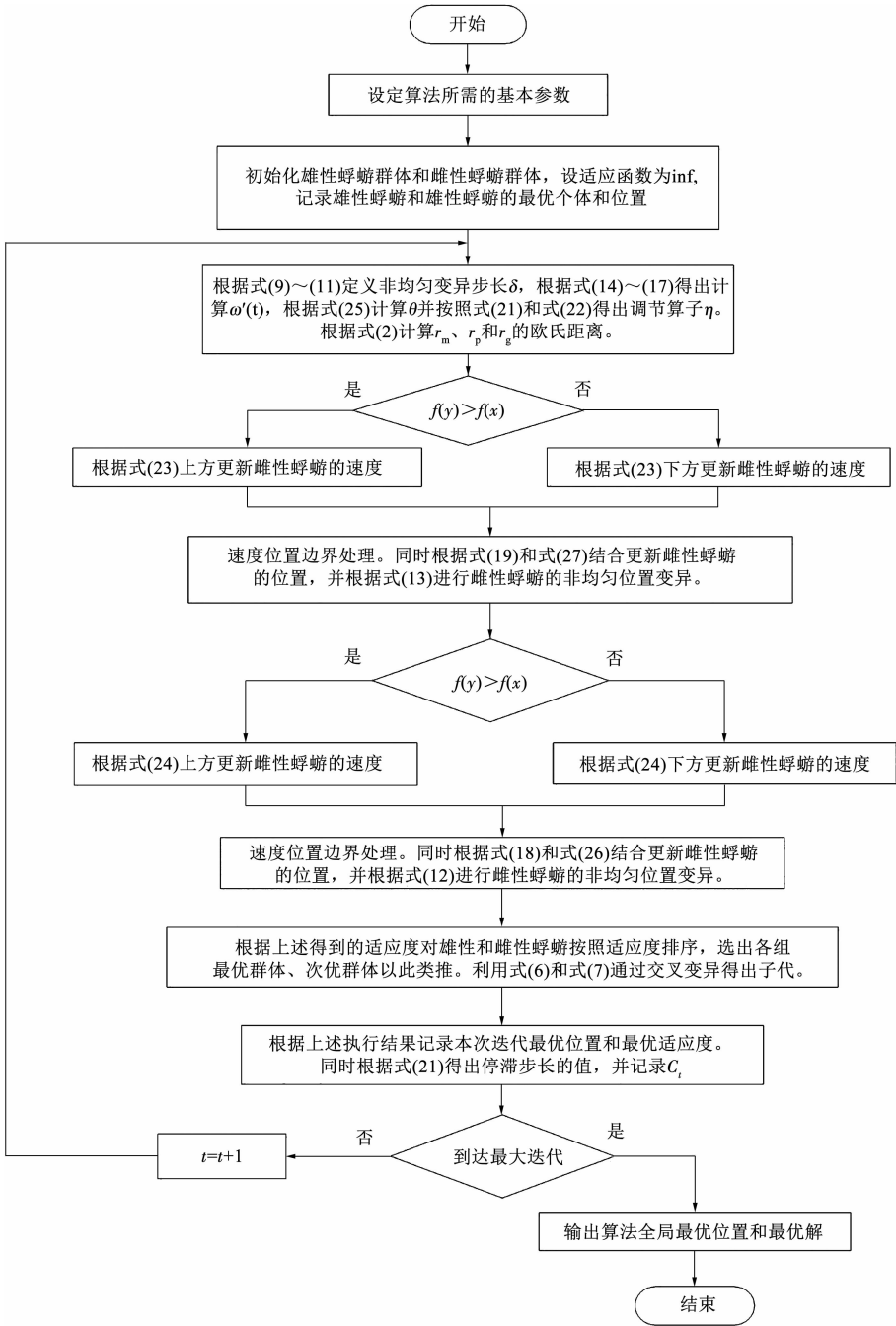


图 2 MIWMA 算法执行流程图

Fig. 2 MIWMA algorithm execution flow chart

2.4.2 MIWMA 时间复杂度分析

根据运算规则,设 MA 算法的目标运行计算量为 $T(n)$,维度为 D ,雌性蜉蝣种群大小为 N_1 ,雄性蜉蝣种群大小为 N_2 ,蜉蝣的交配数量为 n ,因此交配过程的复杂度记为 $O(\frac{n}{2})$,最大迭代次数为 $t_{\max}$ 。因此,MA 的初始化复杂度为 $O(1)$,雌性蜉蝣的更新过程为 $O(DN_1)$,雄性蜉蝣的更新过程为 $O(DN_2)$,因此 MA 的时间复杂度为 $O(1 + t_{\max}(DN_1 + DN_2 + \frac{n}{2}))$ 。

对 MA 引进策略中,引入了非均匀变异的位置更新策略,对每一维位置都变异更新,其此过程的复杂度为 $O(DN_1 + DN_2)$,引入不完全伽马函数和 Beta 累加分布的权重策略的复杂度为 $O(1)$,引入的停滞策略的复杂度为 $O(1)$ 。两种策略都嵌入在速度和位置更新中,因此 MIWMA 的时间复杂度依然为 $T(n) = O(1 + t_{\max}((DN_1 + DN_2) + \frac{n}{2}))$,由此改进算法,并未明显提升复杂度。

3 实验分析与性能评估

为证实 MIWMA 的有效性和优越性,对不同策略进行实验测试,文中引入两个实验测试。其中,引入非均匀变异下的位置更新记为 MMA;引入不完全伽马函数和 Beta 累加分布权重记为 WMA;引入停滞对抗措施策略记为 IMA;组合优化策略记为 MIWMA。另外,引入最近的其他优异的改进的智能算法对比,如自适应惯性权重的樽海鞘群算法(AIWSSA)^[16]、共生非均匀高斯变异的樽海鞘群算法(MSNSSA)^[10]以及稳定边界和收敛增强的改进粒子群算法(IPSO)^[12]进行复现比较。

3.1 参数设置

实验选取种群规模为 40(雄性 20,雌性 20),种群交叉个数为 20。为了公平比较,设置 MSNSSA、AIWSSA 以及 IPSO 的种群大小均为 40,实验结果均独立运行 50 次,各算法具体参数设置见表 2。

3.2 实验记录与分析

实验 1:为验证改进的算法与其他算法的数据有效性比较,实验 1 利用 12 个经典基准测试函数测试,包括 5 个单峰(unimodal)函数,4 个多峰(multimodal)函数及 3 个维度固定(fix-dimension)的测试函数。函数类型有可分(separable)和不可分(non-separable)。测试函数的具体信息见表 3。

表 2 不同算法的参数设置

Tab. 2 Parameter setting of different algorithms

算法名称	参数设置
MIWMA	$a = 1.0, l_{\eta} = 1, d = 5, \beta = 2, a_1 = 1.2, a_2 = 1.6$
MMA	$\sigma_0^2 = 1$,其余同 MIWMA
IMA	$\eta_- = 0.1, \eta_+ = 1, l_{\eta}, d, \beta, a_1, a_2$ 同上
WMA	$a = 1.0, \lambda = 0.1, \nu = 0.1, l_{\eta}, d, \beta, a_1, a_2$ 同上
MA	$a_1 = 1.2, a_2 = 1.6, a = 1.0, l_{\eta} = 1, d = 5, \beta = 2$
MSNSSA	$m = 2.5, R = 1, 2$
AIWSSA	$\alpha = 1, \beta = 0.5, m = 2.5$
IPSO	$a_1 = a_2 = 2.5$

表 3 基准测试函数

Tab. 3 Benchmark function

函数	函数名	特征	维度	搜索空间	最优值
f_1	Sphere	US	30	$[-100, 100]$	0
f_2	Schwefel's 2.22	UN	30	$[-10, 10]$	0
f_3	Schwefel's 1.2	UN	30	$[-50, 50]$	0
f_4	Schwefel's 2.21	UN	30	$[-50, 50]$	0
f_5	Quartic	US	200	$[-1.28, 1.28]$	0
f_6	Ackley's	MN	50	$[-32, 32]$	0
f_7	Rastrigin's	MS	100	$[-5.12, 5.12]$	0
f_8	Penalized	MN	30	$[-50, 50]$	0
f_9	Griewank	MN	30	$[-600, 600]$	0
f_{10}	Kowalik's	FN	4	$[-5, 5]$	3.075×10^{-4}
f_{11}	Hartman's	FN	6	$[0, 1]$	-3.32
f_{12}	Shekel's	FN	4	$[0, 10]$	-5.08

实验 1 记录算法的最优值、平均值、标准差、每次独立运行的平均耗时(T/s),见表 4。表 4 中,从最优值上看,除函数 f_8 以外,整体而言 MIWMA 与 MA 相比具有显著性优势。对于函数 f_8 ,整体而言,引入非均匀变异的位置更新策略和不完全函数与 Beta 累加分布惯性权重最能凸显出算法的优化性能。另外,不同策略(MMA、WMA、IMA)与 MA 对比,都优于原始 MA,且函数 f_7 和 f_9 在多峰函数测试下能够达到理论值 0, f_7 的维度取值为 100 维,同样具有显著性优势。另外,由表中可看出,对 MSNSSA 和 AIWSSA 以及 IPSO 复现算法而言,性能介于 MA 与 MIWMA 之间,即对复现的算法较原始 MA 有效,但不及 MIWMA。对于函数($f_{10} \sim f_{12}$)而言,由于维度是固定的,维度的大小不高,虽然测试函数中含有参数,但求解相对容易,所以引入的策略和原始 MA 和其他复现算法能够达到较好最优值。通过表 7 的秩和检验效果看,引入的策略与其他算法也有更好的效果。由此,MIWMA 的优势较为明显。

表 4 经典测试函数下的优化结果对比

Tab. 4 Comparison of optimization results under classic test functions

函数	算法	最优值	平均值	标准差	T/s	函数	算法	最优值	平均值	标准差	T/s
f_1	MIWMA	3.81×10^{-143}	1.40×10^{-135}	6.50×10^{-134}	1.41	f_7	MIWMA	0	0	0	1.61
	MMA	7.35×10^{-99}	2.44×10^{-91}	1.33×10^{-92}	1.33		MMA	0	0	0	1.65
	WMA	5.77×10^{-95}	1.52×10^{-70}	6.79×10^{-70}	1.43		WMA	0	1.37×10^{-1}	7.72×10^{-1}	1.38
	IMA	6.99×10^{-22}	2.35×10^{-23}	1.27×10^{-22}	1.37		IMA	3.15×10^{-3}	5.59×10^{-2}	9.92×10^{-2}	1.43
	MSNSSA	5.51×10^{-27}	8.22×10^{-25}	5.31×10^{-25}	2.19		MSNSSA	0	0	0	1.33
	MA	4.27×10^{-13}	7.58×10^{-10}	2.09×10^{-9}	1.38		MA	5.96×10^0	1.27×10^1	4.09×10^0	1.51
	AIWSSA	4.11×10^{-113}	3.00×10^{-94}	7.76×10^{-94}	1.17		AIWSSA	0	0	0	1.66
	IPSO	2.82×10^{-55}	1.46×10^{-52}	3.06×10^{-52}	1.87		IPSO	0	5.59×10^{-1}	3.08×10^{-1}	1.33
f_2	MIWMA	3.42×10^{-69}	2.05×10^{-60}	9.61×10^{-60}	1.51	f_8	MIWMA	5.45×10^{-14}	6.43×10^{-13}	6.72×10^{-13}	2.51
	MMA	5.67×10^{-66}	5.58×10^{-55}	3.03×10^{-54}	1.53		MMA	1.05×10^{-16}	6.08×10^{-14}	1.14×10^{-13}	2.54
	WMA	2.94×10^{-47}	1.39×10^{-33}	7.60×10^{-33}	1.42		WMA	2.16×10^{-4}	1.87×10^{-2}	4.91×10^{-2}	2.07
	IMA	3.47×10^{-22}	1.58×10^{-19}	3.90×10^{-19}	1.42		IMA	4.87×10^{-5}	5.26×10^0	3.18×10^0	2.10
	MSNSSA	8.94×10^{-68}	2.01×10^{-31}	2.53×10^{-32}	1.22		MSNSSA	1.88×10^{-3}	3.76×10^{-2}	4.30×10^{-2}	2.44
	MA	2.85×10^{-8}	1.89×10^{-5}	6.02×10^{-5}	1.48		MA	2.65×10^{-11}	4.96×10^{-1}	6.95×10^{-1}	2.34
	AIWSSA	1.20×10^{-129}	2.76×10^{-48}	1.51×10^{-47}	1.52		AIWSSA	6.77×10^{-7}	4.19×10^{-5}	3.72×10^{-5}	3.05
	IPSO	7.49×10^{-32}	8.72×10^{-30}	1.98×10^{-29}	1.55		IPSO	2.13×10^{-4}	1.39×10^{-3}	7.23×10^{-3}	1.75
f_3	MIWMA	3.15×10^{-135}	2.55×10^{-113}	1.7×10^{-112}	1.89	f_9	MIWMA	0	0	0	2.18
	MMA	9.88×10^{-115}	5.81×10^{-107}	2.10×10^{-106}	1.98		MMA	1.33×10^{-15}	2.52×10^{-2}	2.91×10^{-2}	1.91
	WMA	1.99×10^{-89}	5.38×10^{-77}	2.95×10^{-76}	1.81		WMA	0	0	0	1.44
	IMA	2.13×10^{-18}	6.08×10^{-12}	1.09×10^{-5}	1.98		IMA	1.05×10^{-10}	3.34×10^{-1}	4.73×10^{-1}	1.54
	MSNSSA	2.98×10^{-61}	5.62×10^{-60}	6.16×10^{-60}	1.38		MSNSSA	3.86×10^{-12}	2.55×10^{-4}	2.07×10^{-4}	1.59
	MA	7.53×10^{-3}	8.09×10^{-2}	6.26×10^{-2}	2.03		MA	2.35×10^{-9}	1.02×10^{-12}	1.02×10^{-12}	1.73
	AIWSSA	2.10×10^{-112}	2.80×10^{-92}	1.05×10^{-91}	2.08		AIWSSA	0	0	0	2.18
	IPSO	1.79×10^{-18}	8.38×10^{-13}	3.17×10^{-12}	2.08		IPSO	2.16×10^{-11}	9.91×10^{-3}	3.52×10^{-3}	1.94
f_4	MIWMA	5.55×10^{-70}	1.88×10^{-60}	1.17×10^{-59}	1.51	f_{10}	MIWMA	3.07×10^{-4}	3.07×10^{-4}	4.76×10^{-5}	1.78
	MMA	7.33×10^{-51}	3.63×10^{-50}	1.97×10^{-50}	1.55		MMA	3.07×10^{-4}	3.07×10^{-4}	4.04×10^{-5}	1.61
	WMA	6.16×10^{-48}	3.96×10^{-29}	2.15×10^{-28}	1.37		WMA	3.07×10^{-4}	3.54×10^{-4}	2.35×10^{-4}	1.38
	IMA	2.62×10^{-16}	1.06×10^{-10}	3.54×10^{-10}	1.39		IMA	3.07×10^{-4}	6.33×10^{-4}	1.23×10^{-4}	1.92
	MSNSSA	1.82×10^{-31}	2.34×10^{-31}	3.88×10^{-32}	1.67		MSNSSA	3.08×10^{-4}	7.21×10^{-4}	3.78×10^{-4}	1.29
	MA	8.13×10^{-2}	5.56×10^{-1}	3.79×10^{-1}	1.54		MA	3.07×10^{-4}	2.31×10^{-3}	6.12×10^{-3}	1.78
	AIWSSA	3.18×10^{-54}	6.40×10^{-46}	2.58×10^{-45}	1.55		AIWSSA	3.07×10^{-4}	3.31×10^{-4}	4.76×10^{-5}	1.78
	IPSO	9.92×10^{-21}	3.53×10^{-18}	4.95×10^{-18}	1.37		IPSO	3.08×10^{-4}	3.08×10^{-4}	2.57×10^{-4}	1.51
f_5	MIWMA	4.17×10^{-7}	5.62×10^{-5}	4.08×10^{-5}	3.04	f_{11}	MIWMA	-3.32×10^0	-3.32×10^0	1.97×10^{-2}	1.79
	MMA	7.91×10^{-6}	5.62×10^{-5}	5.40×10^{-5}	1.71		MMA	-3.32×10^0	-3.24×10^0	5.82×10^{-2}	1.56
	WMA	6.82×10^{-5}	1.50×10^{-3}	9.44×10^{-4}	1.57		WMA	-3.32×10^0	-3.31×10^0	5.11×10^{-2}	1.42
	IMA	2.46×10^{-4}	2.04×10^{-3}	1.10×10^{-3}	1.59		IMA	-3.32×10^0	-3.30×10^0	5.35×10^{-2}	1.99
	MSNSSA	3.14×10^{-6}	3.26×10^{-5}	1.85×10^{-5}	1.29		MSNSSA	-3.20×10^0	-3.18×10^0	1.92×10^{-2}	1.69
	MA	8.75×10^{-3}	2.17×10^{-2}	8.72×10^{-3}	1.82		MA	-3.32×10^0	-3.29×10^0	5.34×10^{-1}	1.77
	AIWSSA	1.21×10^{-5}	2.21×10^{-4}	1.63×10^{-4}	1.87		AIWSSA	-3.33×10^0	-3.23×10^0	5.87×10^{-2}	1.79
	IPSO	3.04×10^{-4}	1.96×10^{-3}	1.17×10^{-3}	1.95		IPSO	-3.32×10^0	-3.22×10^0	5.19×10^{-2}	1.81
f_6	MIWMA	8.88×10^{-16}	8.88×10^{-16}	0	2.31	f_{12}	MIWMA	-5.08×10^0	-5.08×10^0	4.20×10^{-17}	1.83
	MMA	8.88×10^{-16}	8.88×10^{-16}	5.11×10^{-15}	2.30		MMA	-5.08×10^0	-5.08×10^0	4.09×10^{-15}	1.85
	WMA	8.88×10^{-16}	8.88×10^{-16}	7.52×10^{-16}	1.37		WMA	-1.04×10^1	-8.62×10^0	2.54×10^0	2.16
	IMA	4.44×10^{-15}	3.89×10^{-1}	1.30×10^0	1.45		IMA	-1.04×10^1	-5.28×10^0	3.25×10^0	2.47
	MSNSSA	8.88×10^{-16}	8.88×10^{-16}	8.88×10^{-16}	1.49		MSNSSA	-1.04×10^1	-5.61×10^0	1.68×10^0	2.11
	MA	1.50×10^0	2.77×10^0	8.45×10^{-1}	2.31		MA	-1.04×10^1	-7.48×10^0	3.65×10^0	1.85
	AIWSSA	8.88×10^{-16}	8.88×10^{-16}	8.88×10^{-16}	2.37		AIWSSA	-5.08×10^0	-5.08×10^0	4.20×10^{-5}	1.83
	IPSO	4.44×10^{-15}	4.44×10^{-15}	4.44×10^{-15}	2.22		IPSO	-1.04×10^1	-6.22×10^0	5.11×10^0	2.10

平均值能够呈现独立运行 50 次后的平均效果。从单峰和多峰测试看,平均值整体都要比 MA、MSNSSA、AIWSSA 以及 IPSO 的效果更好。对于函数 f_8 ,其效果不及 MMA 策略的 MIWMA,体现出非均匀变异的位置更新对优化的敏感度非常高,比混合策略还要好。标准差可以反映出独立运行结果的偏离程度,独立运行次数越多,标准差就显得更加精确。对单峰函数和多峰函数,其融合策略的标准差最小,即平均值的偏离程度较小,进一步体现了 MIWMA 的稳定性较好。从维度固定的函数可看出,最优值上与其他算法相比性能相当,平均值直接折射出 MIWMA 的优势。由此 $f_{10} \sim f_{12}$ 的平均值效果最好,且达到理论值,其次是 MMA 和 AIWSSA。标准差衡量它们的偏离程度,虽然 f_{10} 和 f_{11} 的标准差没有其他好,均值不在最优范围,整体而言没有实现最优。

从耗时分析上看,MA、WMA、IMA、MMA 和 MIWMA 的耗时基本持平。因为 MA 引入的几个策略从复杂度上看并没有增加过多的复杂度开销,MIWMA 和 MA 处于同一量级,只是增加了少量语

句的执行。与其他算法比较而言,MA 时间复杂度略高于 IPSO、MSNSSA 和 AIWSSA,因为运行 MA 涉及到雄性和雌性蜉蝣的更新和交配,且后续的变异等增加了运算时间,耗时多数都在 0.5 ~ 2.5 s 之间,单次运行时间也不会太长。但从表 4 中看出,部分函数下 MA 的时间比改进策略还长,主要原因就是受到运行环境的影响,但都不会相差太多,耗时差处于可接受范围内。

算法在整个搜索的平均情况,可以从图形上直观看出各算法搜索的优劣,绘制部分函数收敛曲线见图 3,直观看出迭代 500 次的平均收敛特性。由图 3 可看出,MIWMA 的精度最高,且与其他算法相比整个过程具备较好的搜索性能。从曲线上看,MIWMA 算法在前期具备很好的搜索效果,说明引入的自适应惯性权重策略发挥着较好的搜索指引作用,动态调整能力较好;另外,算法在后期的搜索中也具备较好的寻优能力,迭代曲线持续降低,则引入非均匀的高斯变异和局部停滞发挥了较好的搜索能力。由此,从收敛曲线分析,3 种改进策略的有效性得到进一步验证。

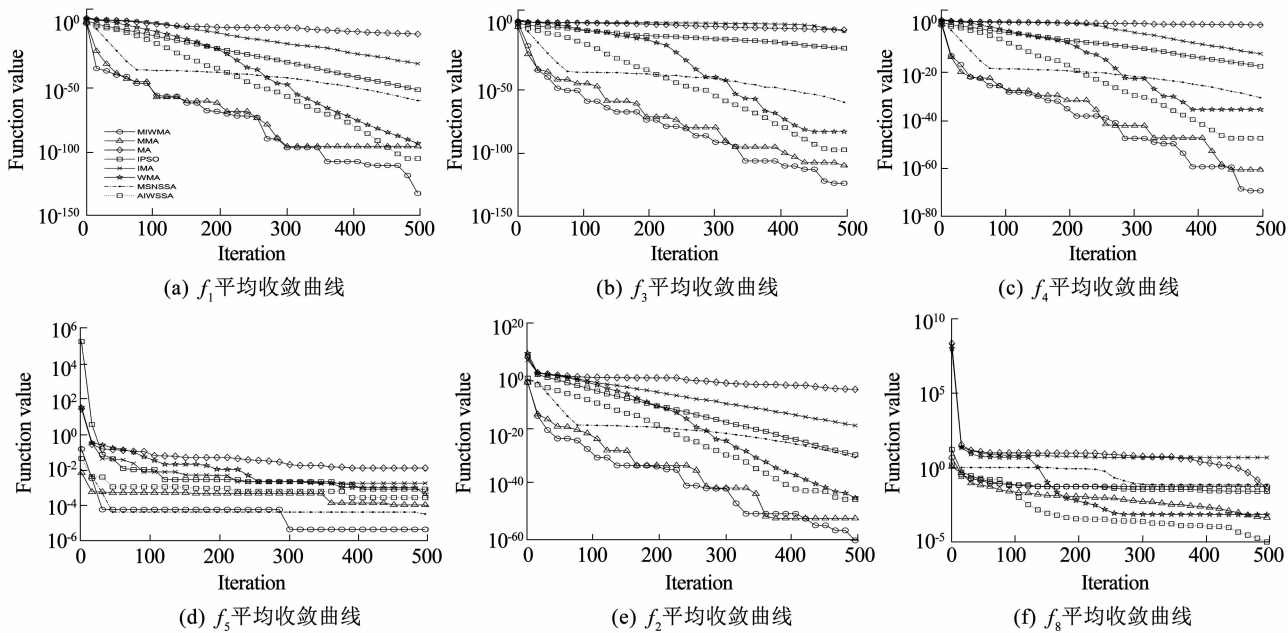


图 3 部分测试函数平均收敛曲线

Fig. 3 Average convergence curves of some test functions

实验 2:为了进一步显现改进的 MIWMA 的优势,对比其有效性和稳定性等性能,实验 2 利用 IEEE CEC2021 函数集中基本、转换和旋转组合集进行性能测试。选取 CEC2021 中单峰、多峰、混合(hybrid)和复合(composition)类别的函数进行测试。函数集的具体信息见表 5,函数定义见文献[17]。

实验 2 验证每种策略的性能对比,设置最大迭

代次数为 20 000 次,维度为 20。同样引入改进 MSNSSA、AIWSSA 和 IPSO 对比,对比结果见表 6。

由表 6,分别记录 CEC01 ~ CEC10 在不同算法运行下的平均值和标准差。由于 IEEE CEC 函数具有复杂的函数特征,难以找到目标函数值。由表中可看出,MIWMA 中只有少部分目标值劣于其他算法,总体上看依然具有较好的性能,靠近目标最优值

的函数个数最多。可判断引入策略是有效的,且从标准差来看也具备良好的稳定性。因此,本文从两个实验证实了 MIWMA 的有效性和算法的性能。

3.3 与其他智能算法对比

为更清晰突出算法的先进性,引入近年其他研究者的成果进行对比,包括:麻雀搜索算法(SSA)^[2]、差分进化的裸鼹鼠算法(DE-NMRA)^[5]、正弦差分进化算法(SinDE)^[18]、协方差矩阵适应进化算法(CMA-ES)^[18]、记忆指导的正弦余弦算法(MG-CSA)^[18]、迭代映射和单纯形法的灰狼算法(SMIGWO)^[14]、嵌入 Circle 映射和小孔成像反向学习的鲸鱼优化算法(MGWO)^[9]以及自适应正态云模型的引力樽海鞘群算法(CGSSA)^[19]。对比结果见表 7。

表 5 CEC2021 函数具体信息

Tab. 5 Detailed information of CEC2021 functions

函数	特征	维度	取值范围	最优值
CEC01	UN	20	[− 100,100]	100
CEC02	MN	20	[− 100,100]	1 100
CEC03	MN	20	[− 100,100]	700
CEC04	MN	20	[− 100,100]	1 900
CEC05	HF	20	[− 100,100]	1 700
CEC06	HF	20	[− 100,100]	1 600
CEC07	HF	20	[− 100,100]	2 100
CEC08	CF	20	[− 100,100]	2 200
CEC09	CF	20	[− 100,100]	2 400
CEC10	CF	20	[− 100,100]	2 500

表 6 CEC2021 优化对比

Tab. 6 Optimization comparison of CEC2021 functions

函数	MIWMA		MA		MMA		IMA	
	平均值	标准差	平均值	标准差	平均值	标准差	平均值	标准差
CEC01	$1.214\ 7\times10^2$	4.073×10^1	$1.311\ 3\times10^5$	1.050×10^7	$2.337\ 2\times10^3$	1.589×10^3	$3.457\ 6\times10^3$	2.484×10^3
CEC02	$2.181\ 6\times10^3$	2.141×10^2	$4.885\ 6\times10^3$	2.716×10^2	$3.457\ 6\times10^3$	2.483×10^2	$2.196\ 9\times10^3$	2.521×10^2
CEC03	$7.330\ 6\times10^2$	7.887×10^0	$1.015\ 3\times10^3$	2.823×10^1	$8.433\ 8\times10^2$	3.757×10^1	$7.581\ 8\times10^2$	1.617×10^1
CEC04	$1.903\ 4\times10^3$	1.585×10^0	$1.901\ 4\times10^3$	8.338×10^{-1}	$1.908\ 1\times10^3$	3.221×10^0	$1.916\ 6\times10^3$	4.352×10^0
CEC05	$3.795\ 5\times10^4$	2.748×10^4	$1.612\ 2\times10^5$	2.273×10^4	$1.570\ 7\times10^5$	2.492×10^4	$3.886\ 2\times10^4$	1.630×10^4
CEC06	$1.838\ 7\times10^3$	5.581×10^1	$3.625\ 7\times10^3$	7.661×10^2	$1.839\ 5\times10^3$	6.017×10^1	$1.978\ 7\times10^3$	2.216×10^2
CEC07	$1.000\ 5\times10^4$	6.724×10^3	$2.723\ 6\times10^4$	9.325×10^3	$1.255\ 7\times10^4$	6.798×10^3	$1.187\ 6\times10^4$	4.722×10^3
CEC08	$2.295\ 6\times10^3$	9.323×10^{-1}	$2.308\ 0\times10^3$	1.068×10^1	$2.308\ 3\times10^3$	1.619×10^0	$2.304\ 6\times10^3$	1.359×10^0
CEC09	$2.902\ 1\times10^3$	8.174×10^1	$2.899\ 6\times10^3$	9.500×10^1	$2.856\ 5\times10^3$	4.615×10^1	$2.817\ 8\times10^3$	5.677×10^1
CEC10	$2.905\ 2\times10^3$	5.208×10^0	$3.134\ 6\times10^3$	1.378×10^1	$2.891\ 0\times10^3$	4.034×10^1	$2.964\ 9\times10^3$	3.795×10^1

函数	WMA		MSNSSA		AIWSSA		IPSO	
	平均值	标准差	平均值	标准差	平均值	标准差	平均值	标准差
CEC01	$4.115\ 3\times10^3$	1.913×10^3	$1.383\ 4\times10^3$	2.125×10^3	$5.096\ 8\times10^6$	1.762×10^6	$4.209\ 1\times10^4$	7.842×10^3
CEC02	$2.038\ 2\times10^3$	3.181×10^2	$2.575\ 3\times10^3$	4.833×10^2	$2.847\ 3\times10^3$	3.356×10^2	$2.375\ 3\times10^3$	1.337×10^3
CEC03	$7.600\ 7\times10^2$	1.294×10^1	$8.489\ 1\times10^2$	1.187×10^1	$7.632\ 5\times10^2$	2.562×10^1	$8.023\ 8\times10^2$	2.827×10^1
CEC04	$1.909\ 8\times10^3$	4.409×10^0	$1.903\ 5\times10^3$	7.015×10^0	$1.905\ 2\times10^3$	2.356×10^1	$1.916\ 7\times10^3$	2.718×10^0
CEC05	$5.167\ 4\times10^4$	2.691×10^4	$1.245\ 8\times10^5$	4.729×10^4	$1.026\ 0\times10^5$	6.334×10^4	$1.916\ 3\times10^5$	4.729×10^4
CEC06	$1.722\ 3\times10^3$	9.405×10^1	$1.811\ 2\times10^3$	1.003×10^2	$1.916\ 5\times10^3$	1.972×10^2	$1.896\ 1\times10^3$	1.041×10^2
CEC07	$1.705\ 3\times10^4$	3.745×10^3	$1.420\ 1\times10^4$	7.805×10^3	$1.060\ 1\times10^4$	4.160×10^3	$9.925\ 5\times10^3$	5.213×10^3
CEC08	$2.300\ 8\times10^3$	1.449×10^0	$2.302\ 1\times10^3$	2.587×10^0	$2.295\ 1\times10^3$	1.886×10^0	$2.300\ 4\times10^3$	8.971×10^0
CEC09	$2.700\ 9\times10^3$	3.218×10^1	$2.870\ 6\times10^3$	2.751×10^1	$2.837\ 2\times10^3$	1.217×10^1	$2.811\ 2\times10^3$	3.551×10^1
CEC10	$2.961\ 0\times10^3$	1.678×10^1	$2.946\ 3\times10^3$	6.292×10^1	$2.901\ 6\times10^3$	2.850×10^1	$2.906\ 1\times10^3$	2.163×10^1

根据表 7 与其他智能算法的对比,虽然其部分函数与其他算法相比不是最优效果,从总体上看,依然具有较好的寻优优势。

3.4 统计分析

实验中对各算法独立运行 50 次,8 个算法以及

12 个测试函数进行对比测试,数据量达到统计分析要求。可以从统计学的角度分析 MIWMA 的优越性,判断是否存在统计学的显著差异。首先,利用非参数检验的 Wilcoxon 秩和检验^[20]方法计算出相应的 P 值,对经典测试函数进行分析;然后,利用

Friedman’s 统计检验^[21] 得出经典测试函数与 CEC 集测试函数的算法排名和综合性评估。表 8 为 Wilcoxon 秩和检验在 $\alpha = 5\%$ 的显著性水平下的 P 值。若 $P < 0.05$, 则拒绝原假设接受备择假设, 即

拒绝两组数据没有显著性的差异, 得出两组数据存在差异的结论。表 8 中, $+/-/=$ 分别表示 MIWMA 算法与其他算法对比的性能为优于/劣于/性能相当的个数, 其中 NaN 标记性能相当。

表 7 与其他智能算法的均值对比
Tab. 7 Comparison of mean value with other intelligent algorithms

算法	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9
MIWMA	1.44×10^{-135}	2.05×10^{-60}	2.55×10^{-113}	1.88×10^{-60}	5.62×10^{-5}	8.88×10^{-16}	0	6.43×10^{-13}	0
SSA ^[3]	2.38×10^{-35}	1.28×10^{-29}	1.95×10^{-47}	9.51×10^{-28}	3.37×10^{-3}	8.88×10^{-16}	8.62×10^{-17}	1.70×10^{-13}	1.43×10^{-5}
DE-NMRA ^[5]	1.01×10^{-33}	2.41×10^{-18}	8.51×10^{-32}	4.96×10^{-24}	2.89×10^1	9.04×10^{-14}	0	2.09×10^{-6}	1.24×10^0
SinDE ^[18]	1.19×10^{-127}	3.81×10^{-51}	9.22×10^{-102}	7.40×10^{-42}	9.88×10^{-4}	8.61×10^{-15}	7.72×10^{-69}	9.97×10^{-7}	2.05×10^0
CMA-ES ^[18]	3.13×10^{-189}	8.54×10^{-58}	3.92×10^{-251}	2.11×10^{-55}	5.76×10^{-4}	3.95×10^{-15}	5.37×10^{-54}	1.05×10^{-12}	3.36×10^{-49}
MG-SCA ^[18]	2.32×10^{-112}	7.61×10^{-56}	0	5.39×10^{-57}	8.32×10^{-5}	8.88×10^{-16}	9.76×10^{-6}	9.44×10^{-13}	7.55×10^{-26}
SMIGWO ^[14]	1.77×10^{-126}	6.29×10^{-53}	4.08×10^{-122}	—	8.18×10^{-5}	—	—	8.18×10^{-5}	1.62×10^{-16}
MWOA ^[9]	0	8.09×10^{-60}	0	4.07×10^{-46}	4.97×10^{-5}	8.88×10^{-16}	0	3.07×10^{-12}	0
CGSSA ^[19]	1.67×10^{-54}	2.91×10^{-48}	1.85×10^{-45}	1.36×10^{-27}	8.82×10^{-5}	8.88×10^{-16}	0	7.71×10^{-13}	0

表 8 经典测试函数下 MIWMA 与其他算法对比的秩和检验 P 值
Tab. 8 P values of rank sum test of MIWMA compared with other algorithms under classic test functions

函数	MA	MMA	WMA	IMA	MSNSSA	AIWSSA	IPSO
f_1	3.01×10^{-11}	5.26×10^{-11}	3.01×10^{-11}	3.01×10^{-11}	2.94×10^{-11}	3.01×10^{-11}	3.01×10^{-11}
f_2	3.01×10^{-11}	7.08×10^{-8}	3.01×10^{-11}	3.01×10^{-11}	2.94×10^{-11}	1.49×10^{-11}	3.01×10^{-11}
f_3	3.01×10^{-11}	3.15×10^{-5}	3.01×10^{-11}	3.01×10^{-11}	2.94×10^{-11}	3.33×10^{-11}	3.01×10^{-11}
f_4	3.01×10^{-11}	4.42×10^{-3}	3.01×10^{-11}	3.01×10^{-11}	2.94×10^{-11}	3.01×10^{-11}	3.01×10^{-11}
f_5	3.01×10^{-11}	3.18×10^{-4}	5.49×10^{-11}	3.33×10^{-11}	8.64×10^{-1}	3.52×10^{-7}	3.01×10^{-11}
f_6	1.21×10^{-12}	NaN	5.12×10^{-1}	1.07×10^{-12}	NaN	NaN	1.68×10^{-14}
f_7	1.21×10^{-12}	NaN	6.24×10^{-10}	1.21×10^{-12}	NaN	NaN	2.42×10^{-5}
f_8	6.37×10^{-2}	3.01×10^{-11}	1.10×10^{-6}	9.83×10^{-8}	3.76×10^{-2}	3.01×10^{-11}	2.03×10^{-9}
f_9	1.21×10^{-12}	NaN	NaN	1.21×10^{-12}	NaN	NaN	NaN
f_{10}	1.03×10^{-7}	1.41×10^{-9}	1.01×10^{-8}	1.94×10^{-3}	8.21×10^{-8}	5.51×10^{-3}	8.21×10^{-8}
f_{11}	2.49×10^{-8}	1.86×10^{-5}	2.01×10^{-8}	4.68×10^{-8}	7.95×10^{-1}	1.00×10^{-3}	6.19×10^{-7}
f_{12}	7.55×10^{-2}	3.01×10^{-11}	3.01×10^{-11}	3.78×10^{-1}	1.57×10^{-11}	3.00×10^{-11}	3.66×10^{-4}
$+/-/=$	10/2/0	9/0/3	10/1/1	11/1/0	7/2/3	9/0/3	11/0/1

由表 8 可知, 大部分的秩和检验 P 值小于 0.05, 从而说明融合策略算法的性能较其他算法的差异是显著的, 即可认定该算法有优异的收敛性能。上述是不同测试函数下评估不同算法的性能, 但还需要进一步验证整个算法在统计学上是否有差异。因此, 利用 Friedman’s 的统计检验, 利用卡方检验(χ^2)^[22] 来评估所得临界值的差异, 判断拒绝或接受原假设得出结论。设显著水平 $\alpha = 0.05$, 对 8 个算法进行性能检验, 见表 9。排名计算公式如下:

$$R_j = \frac{1}{N} \sum_{i=1}^N r_i^j \tag{25}$$

式中: N 为测试函数个数, r_i^j 为测试函数 i 的算法 j

的排名, R_j 越小, 性能越好, 反之则性能越劣。由表 9, MIWMA 的排名最小, 说明其性能是最优越的, 其次是 MMA、AIWSSA、WMA, 最后是 IMA 和 MA, 说明提出的 MIWMA 较其他算法和原始算法都有效, 排名也在其之前。根据表 4 中的结果, 算法个数为 8, 其自由度为 7, 测试函数为 12 个, 卡方值为 31.534 4。查表得自由度为 7, 显著水平 $\alpha = 0.05$ 下, 卡方值为 14.067 14, 由此计算的卡方值远大于查表临界的卡方值, 说明所给的数据存在显著性差异。根据自由度、卡方值与 P 值的换算, 得出 P 值为 $4.953\,47 \times 10^{-5}$ 。因此, 计算 P 值明显小于 0.05, 由此可判定这些算法在精

度上存在显著性差异,所以总体 MIWMA 存在统计学差异,性能表现最好。

表 9 经典函数 Friedman's 测试下算法的统计排名
Tab. 9 Statistical ranking of algorithms under Friedman's test of classic functions

算法	平均排名	总排名	算法	平均排名	总排名
MIWMA	1.708 333	1	MSNSSA	4.791 667	5
MMA	2.791 667	2	MA	7.166 667	8
WMA	4.416 667	4	AIWSSA	3.125 000	3
IMA	6.250 000	7	IPSO	5.750 000	6

上述只是说明总体的 MIWMA 存在统计学差异,但具体的组间差异还需进一步比对。因此,需要进行两两比较调整 α 值,根据 P 值从小到大进行排序,将最小的 P 值和 α/i 进行比较, i 代表算法个数。引入 Holm-Bonferroni 进行后续校正^[23]。MIWMA 的排名和各方面效果都很好,设 MIWMA 为控制算法。进行后续检验来进一步区分算法。表 10 为控制算法与其他算法的校正结果。

表 10 经典函数 Holm's 后续检验下的结果
Tab. 10 Results of Holm's subsequent test of classic function

i	MIWMA vs.	z 值	P 值	α/i	决策
7	MA	4.221 854 9	2.423×10^{-5}	0.007 143	拒绝
6	IMA	3.815 360 0	0.008 333 3	0.008 333	拒绝
5	IPSO	3.557 600 0	0.000 376 9	0.010 000	拒绝
4	MSNSSA	3.111 200 0	0.001 863 3	0.012 500	拒绝
3	WMA	2.755 900 0	0.005 853 1	0.166 670	拒绝
2	AIWSSA	1.952 100 0	0.050 926 3	0.025 000	接受
1	MMA	1.337 860 0	0.180 942 1	0.050 000	接受

根据表 10 中的结果,有 MIWMA vs. MMA 和 MIWMA vs. AIWSSA 的判定决策是接受,即 $0.180\ 942\ 1 > 0.05$ 和 $0.050\ 926\ 3 > 0.025$,即 MIWMA 与 MMA、AIWSSA 的组间差异不明显,与其他性能比较差异明显,MA 的统计分析结果最差。表中虽不能拒绝 AIWSSA 的假设,但从均值、平均排名等信息也能充分说明与 MIWMA 存在差异的。从表中可看出,MIWMA 与 MMA 判决结果为接受,这与 MIWMA 的统计分析不显著,说明非均匀变异的位置更新策略最有效,较于其他两个改进策略而言更具有显著性。统计分析证明该统计方法进行整体算法性能的比较是科学、可靠的。结果反映 MIWMA 算法的优越性和可靠性。

表 11 是 CEC2021 测试函数的统计检验对比。由表 11,MIWMA 的平均排名最低,说明它的性能在比较的几个算法中是最优的。同样,根据上述方法,

设置 MIWMA 为后续检验的控制算法进一步分析它们之间的差异。

表 11 CEC2021 的 Friedman's 测试下的统计排名
Tab. 11 Statistical ranking under Friedman's test of CEC2021

算法	平均排名	总排名	算法	平均排名	总排名
MIWMA	2.500 0	1	WMA	3.700 0	2
MA	6.900 0	8	AIWSSA	4.200 0	4
MMA	4.050 0	3	MSNSSA	4.600 0	7
IMA	4.550 0	5	IPSO	4.600 0	6

利用 CEC2021 的算法个数为 8,自由度为 7,卡方值为 22.933,大于临界卡方值 14.067 14,计算出 $P=0.001\ 751$,小于 0.05,由此存在显著的统计学差异。表 12 可看出,MIWMA vs. WMA 和 MIWMA vs. AIWSSA 的判决是接受,其他均拒绝。这说明,WMA 和 AIWSSA 在 CEC2021 性能测试具有较好的性能,从表 12 中的数据看,相对劣于 MIWMA。与其他函数相比,具有统计学上的统计差异,表现最差的为 MA。结合经典测试函数与 CEC2021 测试函数而言,MIWMA 在两种测试函数的基础上都能够表现较好的性能,从统计学上分析 MIWMA 的性能的可靠性。

表 12 CEC2021 的 Holm's 后续检验下的结果
Tab. 12 Results of Holm's follow-up test for CEC2021

i	MIWMA vs.	z 值	P 值	α/i	决策
7	MA	4.017 823 3	5.89×10^{-5}	0.007 14	拒绝
6	MSNSSA	2.882 518 6	0.003 952	0.008 33	拒绝
5	IPSO	2.882 775 1	0.003 952	0.010 00	拒绝
4	MMA	2.601 007 9	0.002 950	0.012 50	拒绝
3	IMA	1.826 226 4	0.067 850	0.016 67	拒绝
2	AIWSSA	1.552 020 1	0.120 662	0.025 00	接受
1	WMA	1.095 100 3	0.273 517	0.050 00	接受

4 工程优化分析

为证实提出的融合策略在实际工程案例中的可行性,引入两种工程优化案例为测试背景,用于解决工程中的 NP 难题和目标约束问题. 选用三杆桁架设计问题和拉/压弹簧设计问题进行各算法的优化对比,利用静态罚函数法^[24]对约束条件进行处理,使各种对比方法达到公平的比较。其罚函数定义为

$$\zeta(z) = f(z) + \left[\sum_{i=1}^m l_i \cdot \max(0, t_i(z))^\alpha + \sum_{j=1}^n o_j |U_j(z)|^\varphi \right] \quad (26)$$

式中: $\zeta(z)$ 为目标函数, o_j 和 l_i 为惩罚系数, $t_i(z)$ 和 $U_j(z)$ 为约束条件。为了与其他算法对比,取值与文献[25]和文献[16]设置的参数值保持一致,取 α 和

φ 的值分别为 2 和 1, 惩罚因子取 1 000。每个算法分别对具体工程优化 50 次。

4.1 拉/压弹簧设计问题

拉/压弹簧设计问题是一种经典的约束优化问题, 其目标是减轻拉/压弹簧设计的重量, 见图 4。其主要约束条件有 4 个, 还包括剪应力、端振频率和最小扰度。参数主要有导线直径(d), 平均线圈直径(D)和有效线圈数(N)等。为方便, 将其表示为 $\mathbf{X} = (x_1, x_2, x_3)$, 其中向量 $\mathbf{X}$ 的元素分别对应 (d, D, N)。因此, 其数学模型为

$$\text{Minimize } f(\mathbf{X}) = (x_3 + 2)x_2x_1^2 \quad (27)$$
$$\text{s. t. } g_1(\mathbf{X}) = 1 - \frac{x_2^3x_3}{71\,785x_1^4} \leq 0$$
$$g_2(\mathbf{X}) = \frac{4x_2^2 - x_1x_2}{12\,566(x_2x_1^3 - x_1^4)} + \frac{1}{5\,108x_1^2} - 1 \leq 0$$
$$g_3(\mathbf{X}) = 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0$$
$$g_4(\mathbf{X}) = \frac{x_1 + x_2}{1.5} - 1 \leq 0 \quad (28)$$

式中: $0.05 \leq x_1 \leq 2, 0.25 \leq x_2 \leq 1.3, 2 \leq x_3 \leq 15$ 。对拉/压弹簧问题, 用 7 种不同算法进行对比。为比较明显的优势, 引入 SSA^[1]、PSO^[2]、MFO^[26]、CSA^[7]、SCA 分别优化, 结果见表 13。

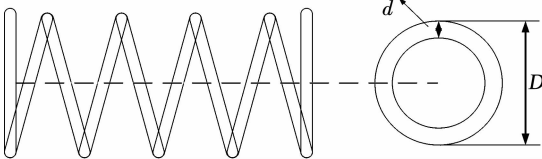


图 4 拉/压弹簧设计示意

Fig. 4 Schematic diagram of tension/pressure spring design

表 13 不同算法优化拉/压弹簧设计问题对比
Tab. 13 Comparison of design problems of tension/pressure springs optimized by different algorithms

算法	x_1	x_2	x_3	$f(\mathbf{X})$	平均值
MIWMA	0.051 08	0.341 22	11.338 40	0.012 664	0.012 667
MA	0.051 17	0.350 33	11.545 93	0.012 677	0.012 677
SSA ^[1]	0.051 20	0.345 21	12.004 31	0.012 676	0.012 679
PSO ^[2]	0.051 72	0.357 64	11.244 54	0.012 674	0.012 675
SCA	0.050 78	0.334 78	12.722 69	0.012 873	0.012 889
MFO ^[26]	0.050 01	0.313 50	14.226 23	0.012 753	0.012 753
CSA ^[7]	—	—	—	0.012 669	0.012 674

由表 13, MIWMA 算法在拉/压弹簧问题具有良好的搜索能力; 且独立运行 50 次后, 均值与最优值之间也较为稳定, 且优于其他对比算法。不难从表 13 中看出, $f(\mathbf{X})$ 通过 MIWMA 的算法优化后, 最优值也仅在 0.000 01 的精度上出现差别。算法的搜索都达到最优取值范围的附近, 通过优化也只能

进一步得出在更小搜索空间中也有好的搜索能力。

4.2 三杆桁架设计问题

三杆桁架设计是满足压力条件下使得结构的重量最小为目标优化问题。三杆桁架的示意图见图 5。其中包括 3 个决策变量, 分别为 A_1, A_2 和 A_3 。由于 A_1, A_3 对称, 因此 $A_1 = A_3$ 。同理, 为方便表示, 将变量表示为 $\mathbf{X} = (x_1, x_2)$ 。其数学模型如下:

$$\text{Minimize } f(\mathbf{X}) = (2\sqrt{2}x_1 + x_2)l \quad (29)$$
$$\text{s. t. } g_1(\mathbf{X}) = \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leq 0$$
$$g_2(\mathbf{X}) = \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2}P - \sigma \leq 0$$
$$g_3(\mathbf{X}) = \frac{1}{x_1 + \sqrt{2}x_2}P - \sigma \leq 0$$
$$0 \leq x_i \leq 1, i = 1, 2 \quad (30)$$

式中, 取 $l = 100\text{ cm}$, $P = 2\text{ kN/cm}^2$, $\sigma = 2\text{ kN/cm}^2$ 。不同优化算法的结果见表 14。该问题在 MIWMA 和 MA 的基础上, 还引入 SSA、SMA^[27]、MFO 以及其他通过改进的算法 ATMDE 和 AIWSSA 在相应文献进行比较, 保证对比公平。

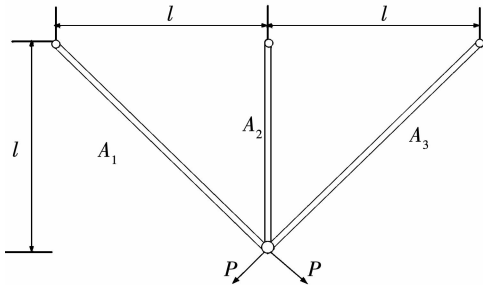


图 5 三杆桁架示意

Fig. 5 Schematic diagram of three-bar truss

由表 14, MIWMA 对三杆桁架设计的工程问题同样能保持良好的性能, 工程优化问题的局部搜索有更好寻优效果, 并优于后面通过改进的算法, 适用于解决工程优化中需求精度高的问题。通过引入的工程优化案例可看出, 表 13 和表 14 体现出在工程应用中也有更好的局部搜索能力。

表 14 不同算法的三杆桁架设计求解对比
Tab. 14 Comparison of three-bar truss design solutions of different algorithms

算法	x_1	x_2	$f(\mathbf{X})$	平均值
MIWMA	0.788 6	0.408 2	263.891 47	263.891 50
MA	0.788 6	0.408 2	263.891 51	263.891 57
SSA	0.788 7	0.407 9	263.891 52	263.891 90
SMA	0.785 7	0.417 6	263.998 30	268.886 30
MFO	0.788 4	0.408 7	263.891 50	263.904 10
ATMDE ^[26]	0.788 7	0.409 5	263.959 00	—
AIWSSA ^[27]	0.788 7	0.408 3	263.895 80	—

5 结 论

针对 MA 算法搜索和开发的不平衡和易陷入局部最优的缺点进行改进,提出一种增强全局搜索和自适应的蜉蝣算法——MIWMA。引入非均匀变异和上一次的位置更新策略指引其向精英位置移动,提高其多样性和搜索能力;引入不完全伽马函数与 Beta 累加函数的分布数克服非线性自适应惯性权重的缺陷,动态调整算法搜索的自适应能力,进一步提高算法的收敛精度;引入停滞策略使其跳出局优,调节群体间信息交流合作,从而增强搜索性能。为证实 MIWMA 的性能和先进性,分别引入经典测试函数集验证算法的有效性和可靠性,引入 CEC2021 函数验证算法的性能。最后验证在工程难题中的应用潜能,在工程问题具有一定适用能力。对于大规模复杂问题、通信网络的资源分配和多约束调度下的最优问题,是下一步的研究重点。

参考文献

- [1] 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: 118. DOI:10.1016/j.advengsoft.2017.07.002
- [2] RICCI F, ROKACH L, SHAPIRA B, et al. Recommender systems handbook[M]. Berlin: Springer, 2011: 1. DOI:10.1007/978-0-387-85820-3
- [3] XUE Jiankai, SHEN Bo. A novel swarm intelligence optimization approach; sparrow search algorithm[J]. *Systems Science & Control Engineering*, 2020, 8(1): 22. DOI:10.1080/21642583.2019.1708830
- [4] PANDA M, DAS B, PATI B B. Grey wolf optimization for global path planning of autonomous underwater vehicle[C]//*Proceedings of the Third International Conference on Advanced Informatics for Computing Research*. New York: Association for Computing Machinery, 2019: 177. DOI:10.1145/3339311.3339314
- [5] SALGOTRA R, SINGH U, SINGH G, et al. A self-adaptive hybridized differential evolution naked mole-rat algorithm for engineering optimization problems [J]. *Computer Methods in Applied Mechanics and Engineering*, 2021, 383: 113916. DOI:10.1016/J.CMA.2021.113916
- [6] YANG Xinshe. Firefly algorithms for multimodal optimization[C]//*Proceedings of International Symposium on Stochastic Algorithms*. Heidelberg: Springer, 2009: 169. DOI:10.1007/978-3-642-04944-6_14
- [7] MALIK S B. Chameleon swarm algorithm; a bio-inspired optimizer for solving engineering design problems[J]. *Expert Systems with Applications*, 2021, 174: 114685. DOI:10.1016/J.ESWA.2021.114685
- [8] ZERVOUDAKIS K, TSAFARAKIS S. A mayfly optimization algorithm[J]. *Computers & Industrial Engineering*, 2020, 145: 106559. DOI:10.1016/j.cie.2020.106559
- [9] 张达敏,徐航,王依柔,等. 嵌入 Circle 映射和逐维小孔成像反向学习的鲸鱼优化算法[J]. *控制与决策*, 2021, 36(5): 1173. ZHANG Damin, XU Hang, WANG Yirou, et al. Whale optimization algorithm for embedded circle mapping and one-dimensional oppositional learning based small hole imaging [J]. *Control and Decision*, 2021, 36(5): 1173. DOI:10.13195/j.kzyjc.2019.1362
- [10] 陈忠云,张达敏,辛梓芸. 多子群的共生非均匀高斯变异樽海鞘群算法[J/OL]. *自动化学报*[2022-04-14]. CHEN Zhongyun, ZHANG Damin, XIN Ziyun. Multi-subpopulation based symbiosis and non-uniform Gaussian mutation salp swarm algorithm [J/OL]. *Acta Automatica Sinica* [2022-04-14]. DOI:10.16383/j.aas.c190684
- [11] 高文欣,刘升,肖子雅,等. 收敛因子和黄金正弦指引机制的蝴蝶优化算法[J]. *计算机工程与设计*, 2020, 41(12): 3384. GAO Wenxin, LIU Sheng, XIAO Ziya, et al. Butterfly optimization algorithm based on convergence factor and gold sinusoidal guidance mechanism [J]. *Computer Engineering and Design*, 2020, 41(12): 3384. DOI:10.16208/j.issn1000-7024.2020.12.013
- [12] TONG X T, CHOI K P, LAI T L, et al. Stability bounds and almost sure convergence of improved particle swarm optimization methods[J]. *Research in the Mathematical Sciences*, 2021, 8(2): 18. DOI:10.1007/S40687-020-00241-4
- [13] SHI Yuhui, EBERHART R C. A modified particle swarm optimizer [C]//*Proceedings of 1998 IEEE International Conference on Evolutionary Computation*. Piscataway: IEEE, 1998:69. DOI:10.1109/ICEC.1998.699146
- [14] 王梦娜,王秋萍,王晓峰. 基于 Iterative 映射和单纯形法的改进灰狼优化算法[J]. *计算机应用*, 2018, 38(增刊2): 16. WANG Mengna, WANG Qiuping, WANG Xiaofeng. Improved grey wolf optimization algorithm based on iterative mapping and simplex method[J]. *Journal of Computer Applications*, 2018, 38(S2): 16
- [15] 黄洋,鲁海燕,许凯波,等. 一种动态调整惯性权重的简化均值粒子群优化算法[J]. *小型微型计算机系统*, 2018, 39(12): 2590. HUANG Yang, LU Haiyan, XU Kaibo, et al. Simplified mean particle swarm optimization algorithm with dynamic adjustment of inertia weight[J]. *Journal of Chinese Computer Systems*, 2018, 39(12): 2590
- [16] 白钰,彭珍瑞. 基于自适应惯性权重的樽海鞘群算法[J]. *控制与决策*, 2022, 37(1): 237. BAI Yu, PENG Zhenrui. Salp swarm algorithm based on adaptive inertia weight [J]. *Control and Decision*, 2022, 37(1): 237. DOI:10.13195/j.kzyjc.2020.0454
- [17] MOHAMED A W, HADI A, MOHAMED A K, et al. Problem definitions and evaluation criteria for the CEC 2021 special session and competition on single objection bound constrained numerical optimization [C]//*Proceedings of 2021 IEEE Congress On Evolutionary Computation*. [S.l.]: IEEE, 2020
- [18] GUPTA S, DEEP K, ENGELBRECHT A P. A memory guided sine cosine algorithm for global optimization [J]. *Engineering Applications of Artificial Intelligence*, 2020, 93: 103718. DOI:10.1016/j.engappai.2020.103718
- [19] 张铸,张仕杰,饶盛华,等. 基于自适应正态云模型的引力樽海鞘群算法[J]. *控制与决策*, 2022, 37(2): 344

- ZHANG Zhu, ZHANG Shijie, RAO Shenghua, et al. Gravity salp swarm algorithm based on adaptive normal cloud model[J]. Control and Decision, 2022, 37(2): 344. DOI:10.13195/j.kzyjc.2020.1292
- [20] WILCOXON F. Individual comparisons by ranking methods[J]. Biometrics Bulletin, 1945, 1(6): 80. DOI:10.2307/3001968
- [21] CARABANTE K M, CHOKUMNOYPORN N, SRIWATTANA S, et al. Comparing Friedman versus Mack-Skillings data analyses on duplicated rank data: a case of visual color intensity[J]. Journal of the Science of Food and Agriculture, 2019, 99(13): 5696. DOI: 10.1002/jsfa.9832
- [22] WOSNIOK W, HAECKEL R. A new indirect estimation of reference intervals: truncated minimum chi-square (TMC) approach[J]. Clinical Chemistry and Laboratory Medicine, 2019, 57(12): 1933. DOI:10.1515/ccm-2018-1341
- [23] LIU S H, MERNIK M, HRNČIĆ D, et al. A parameter control method of evolutionary algorithms using exploration and exploitation measures with a practical application for fitting Sova's mass transfer model[J]. Applied Soft Computing, 2013, 13(9): 3792. DOI: 10.1016/j.asoc.2013.05.010
- [24] JAYSWAL A, PREETI. An exact l1 penalty function method for multi-dimensional first-order PDE constrained control optimization problem[J]. European Journal of Control, 2020, 52: 34. DOI: 10.1016/j.ejcon.2019.07.004
- [25] FU Chunming, XU Yadong, JIANG Chao, et al. Improved differential evolution with shrinking space technique for constrained optimization[J]. Chinese Journal of Mechanical Engineering, 2017, 30(3): 553. DOI:10.1007/s10033-017-0130-4
- [26] MIRJALILI S. Moth-flame optimization algorithm: a novel nature-inspired heuristic paradigm[J]. Knowledge-Based Systems, 2015, 89: 228. DOI:10.1016/j.knsys.2015.07.006
- [27] KUMAR C, RAJ T D, PREMKUMAR M, et al. A new stochastic slime mould optimization algorithm for the estimation of solar photovoltaic cell parameters[J]. Optik, 2020, 223: 165277. DOI:10.1016/j.ijleo.2020.165277
- (编辑 苗秀芝)

(上接第 121 页)

- [6] LATOUR M, PILUSO V, RIZZANO G. Experimental analysis on friction materials for supplemental damping devices[J]. Construction and Building Materials, 2014, 65: 159. DOI: 10.1016/j.conbuildmat.2014.04.092
- [7] LATOUR M, PILUSO V, RIZZANO G. Experimental analysis of beam-to-column joints equipped with sprayed aluminium friction dampers[J]. Journal of Constructional Steel Research, 2018, 146: 33. DOI: 10.1016/j.jcsr.2018.03.014
- [8] LEMOS A, SILVA L S D, LATOUR M, et al. Numerical modelling of innovative DST steel joint under cyclic loading[J]. Archives of Civil and Mechanical Engineering, 2018, 18(3): 687. DOI: 10.1016/j.acme.2017.10.008
- [9] ANOUSHEHEI M, DANESHJOO F, MAHBOUBI S, et al. Experimental investigation on hysteretic behavior of rotational friction dampers with new friction materials[J]. Steel and Composite Structures, 2017, 24(2): 239. DOI: 10.12989/scs.2017.24.2.239
- [10] NAEEM A, KIM J. Seismic retrofit of structures using rotational friction dampers with restoring force[J]. Advances in Structural Engineering, 2020, 23(16): 3525. DOI: 10.1177/1369433220939213
- [11] 隋伟宁, 周魁, 董锐. 新型转动型摩擦阻尼器设计及应用[J]. 建筑结构, 2018, 48(增刊2): 404
- SUI Weining, ZHOU Kui, DONG Rui. Design and application of a new type rotational friction damper[J]. Building Structure, 2018, 48(S2): 404. DOI: 10.19701/j.jzjg.2018.S2.080
- [12] 隋伟宁, 周魁, 王占飞. 转动型摩擦阻尼器力学性能及抗震效果分析[J]. 工业建筑, 2018, 48(4): 137
- SUI Weining, ZHOU Kui, WANG Zhanfei. Analysis of mechanical properties and seismic behavior of rotation type friction dampers[J]. Industrial Construction, 2018, 48(4): 137. DOI: 10.13204/j.gyjz201804027
- [13] GOLMOGHANY M Z, ZAHRAI S M. Improving seismic behavior using a hybrid control system of friction damper and vertical shear panel in series[J]. Structures, 2021, 31: 369. DOI: 10.1016/j.istruc.2021.02.007
- [14] LEE J, KANG H, KIM J. Seismic performance of steel plate slit-friction hybrid dampers[J]. Journal of Constructional Steel Research, 2017, 136: 128. DOI: 10.1016/j.jcsr.2017.05.005
- [15] MIRZABAGHERI S, SANATI M, AGHAKOUCHAK A A, et al. Experimental and numerical investigation of rotational friction dampers with multi units in steel frames subjected to lateral excitation[J]. Archives of Civil and Mechanical Engineering, 2015, 15(2): 479. DOI: 10.1016/j.acme.2014.05.009
- (编辑 苗秀芝)