

求解全局优化问题的改进灰狼算法

张 阳, 周溪召

(上海理工大学 管理学院, 上海 200093)

摘要: 灰狼算法是一种高效的优化技术, 但其在一些问题存在求解精度不高、收敛速度较慢和易于陷入局部最优的缺点。因此, 提出了一种改进的灰狼优化算法 (MGWO)。该算法引入了 3 种改进策略: 平衡算法全局搜索性和局部开发性的指数规律收敛因子调整策略、提高算法求解精度的自适应位置更新策略和修订动态权重策略。通过两组在 10 个基准测试函数上的对比实验, 验证 3 种改进策略的有效性。实验结果表明, 综合使用 3 种策略的 MGWO_4 明显提升了基本灰狼算法 (GWO) 的性能, 而且优于其他文献中的改进灰狼算法和其他数个优化算法。最后, 在工程设计问题上的实验结果进一步证明了 MGWO 高效的寻优能力。

关键词: 灰狼优化算法; 收敛因子; 全局优化; 元启发式算法

中图分类号: TP 301.6 **文献标志码:** A

Modified grey wolf optimization algorithm for global optimization problems

ZHANG Yang, ZHOU Xizhao

(Business School, University of Shanghai for Science and Technology, Shanghai 200093, China)

Abstract: The grey wolf optimization algorithm is an efficient optimization technology, however, it still has some shortcomings, such as low accuracy, slow convergence speed and easy to fall into local optimum. Therefore, a modified grey wolf optimization algorithm (MGWO) was proposed. Three improved strategies were introduced into the algorithm: the strategy of the convergence factor adjustment with exponential law, the adaptive position updating strategy of grey wolf and the revised dynamic weight strategy. Through the implementation of two groups of comparative experiments on 10 benchmark functions, the effectiveness of the three improved strategies was verified. The experimental results show that the MGWO-4 with the three improved strategies improves the performance of the basic grey wolf optimization algorithm (GWO) significantly, and is superior to the improved grey wolf algorithm and several other optimization algorithms in other literatures. The experimental results on an engineering design question further prove that MGWO is a powerful optimization technique.

Keywords: grey wolf optimization algorithm; convergence factor; global optimization; meta-heuristic algorithm

收稿日期: 2020-03-31

基金项目: 国家自然科学基金资助项目 (61273042)

第一作者: 张 阳 (1994-), 男, 硕士研究生。研究方向: 元启发式算法、人工神经网络。E-mail: 1459870532@qq.com

通信作者: 周溪召 (1964-), 男, 教授。研究方向: 交通工程、管理科学与工程。E-mail: xzzhou@usst.edu.cn

元启发式算法是一类在特定的资源(时间等)约束条件下求解问题近似最优的优化技术,具有简单、高效和计算成本小等优点。自1983年Kirkpatrick等提出模拟退火算法(simulated annealing algorithm, SA)^[1]和1992年Holland提出遗传算法(genetic algorithm, GA)^[2]以来,众多学者投入到这一研究领域。经过几十年的发展,许多元启发式算法相继被提出并已广泛应用于研究领域和实际场景,如粒子群算法(particle swarm optimization, PSO)^[3]、蚁群算法(ant colony optimization, ACO)^[4]、引力搜索算法(gravity search algorithm, GSA)^[5]、蘑菇繁殖算法(mushroom reproduction optimization, MRO)^[6]、人工蜂群算法(artificial bee colony algorithm, ABC)^[7]、教学优化算法(teaching-learning-based optimization, TLBO)^[8]和哈里斯鹰算法(Harris hawks optimization, HHO)^[9]等。但是,根据没有免费的午餐定理(NFL)^[10],没有一种算法适用于求解所有问题,因此,仍有必要对元启发式算法进行进一步的研究。

在文献[6]中,元启发式算法可以根据不同的灵感来源而分为3类:群智能算法、非群智能算法和物理/化学算法。2014年Mirjalili等^[11]受自然界中灰狼领导等级制度和群体狩猎行为启发,提出了一种新的群智能算法——灰狼算法(grey wolf optimization, GWO)。灰狼算法是一种强大的优化技术,已被成功应用于许多领域。谭念等^[12]提出了一种基于灰狼算法和支持向量机(GWO-SVM)的木材密度预测模型,为杉木性质定量分析提供了理论依据;姚鹏等^[13]提出了一种基于改进的扰动流体动态系统与灰狼优化理论的混合航路规划算法,为复杂地形环境下的无人机三维航路规划问题提供了解决方案;王书芹等^[14]提出了基于灰狼优化算法的长短期记忆模型(long short term memory, LSTM),解决了传统LSTM易于收敛到局部最优的问题。

然而,同其他元启发式算法一样,灰狼算法也存在着收敛速度较慢、求解精度不高、在一些复杂问题上容易陷入局部最优等缺点。为提升GWO的性能,许多学者进行了相关研究。Joshi等^[15]提出了一种基于精英策略的增强灰狼算法(enhanced grey wolf optimization, EGWO); Heidari等^[16]提出了结合列维飞行和贪婪搜索机制的灰狼算法(levy-embedded GWO, LGWO);文献[17]提出了基于模糊层次算子的灰狼算法,并提出了两

种个体位置更新时的比例权重策略; Saremi等^[18]提出了基于进化种群动力学(evolutionary population dynamics, EPD)的灰狼算法(GWO-EPD), EPD要求表现不佳的搜寻个体进行动态的位置调整;王敏等^[19]提出了一种基于反向学习策略、非线性收敛因子和变异算法的新型灰狼算法(novel GWO, NGWO)。在GWO中,收敛因子 a 在一定程度上决定了算法局部开发过程和全局搜索过程之间的平衡。本研究首次提出了一种新的指数变化形式的收敛因子,而且与以前的所有针对 a 的改进之处所不同的是:提出的收敛因子最终并不会减小为0。同时为了加快算法的收敛速度,重新定义了灰狼算法中的 α , β , δ 狼,并提出了一种基于自适应调整策略的灰狼位置更新公式。此外,针对文献中提出的动态权重策略进行修订,以获得更好的求解性能和鲁棒性。最终,使用了10个典型的基准测试函数验证了以上改进策略的有效性。

1 灰狼优化算法

灰狼优化算法是一种较新的优化技术,其主要思想是灰狼群体中的领导等级制度和群体狩猎模式。在自然界中,灰狼群体具有森严的社会等级制,如图1所示,狼群中所有个体划分为4个等级: α , β , δ 及 ω 。 α 是狼群的领袖; β 仅次于 α ,其职责是辅助头狼 α 制定决策及其他群体性活动;接下来是 δ ,狼群中的侦察兵、哨兵、长老、狩猎者及看守者都属于此类;等级最低的狼则为 ω 。除等级制度外,小组狩猎也是灰狼种群中一个有趣的社会行为。

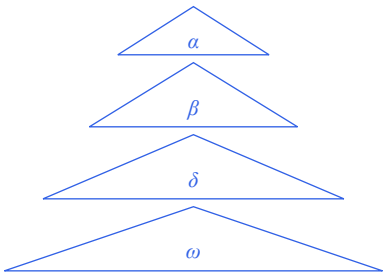


图1 灰狼种群社会等级结构

Fig.1 Social hierarchy structure of grey wolf population

在GWO中,适应度值最优的个体被视为 α ,适应度值第二和第三的分别被定义为 β 和 δ ,其余个体均为 ω 。在算法实施中,以下等式用来描述灰狼种群包围猎物的行为:

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

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

式中: t 表示当前迭代次数; $X(t)$ 表示灰狼个体; $X_p(t)$ 表示当前的灰狼个体; D 表示个体与猎物之间的距离; A 和 C 是系数向量, 分别用式(3)和式(4)来计算。

$$A = 2a \cdot r_1 - a \quad (3)$$

$$C = 2 \cdot r_2 \quad (4)$$

式中: r_1, r_2 是 $0 \sim 1$ 之间的随机向量; a 表示收敛因子, 可用等式 $a = 2 \times (1 - t/T)$ 来计算, T 为设定的最大迭代次数。

在狩猎时, GWO 假定 α, β, δ 对于猎物的潜在位置更具有感知性, 所以, 灰狼个体会分别判断自身与 α, β, δ 之间的距离 $D_\alpha, D_\beta, D_\delta$ (式(5)), 分别计算其朝向三者的移步距离 X_1, X_2, X_3 (式(6)), 并最终在向着三者的包围圈之内进行移步, 移动公式为式(7)。

$$D_\alpha = |C_1 \cdot X_\alpha - X(t)|, \\ D_\beta = |C_1 \cdot X_\beta - X(t)|, D_\delta = |C_1 \cdot X_\delta - X(t)| \quad (5)$$

$$X_1 = X_\alpha - A_1 \cdot D_\alpha, X_2 = X_\beta - A_1 \cdot D_\beta, \\ X_3 = X_\delta - A_1 \cdot D_\delta \quad (6)$$

$$X(t+1) = (X_1 + X_2 + X_3) / 3 \quad (7)$$

2 改进的灰狼优化算法

2.1 指数规律收敛因子调整策略

全局搜索能力和局部开发能力是描述元启发式算法性能的两种通用属性, 如何有效平衡两者也是元启发式算法首要思考的问题。全局搜索是指在搜索空间搜寻最优解, 强的全局搜索性可使算法有效避免陷入局部最优, 增强算法的寻优能力; 局部开发能力是指在局部区域进行精确搜索, 强的局部开发性能够提升算法的求解精度、加快算法收敛速度。一般情况下, 迭代的前期使用全局搜索策略, 而在后期则使用局部开发策略。在 GWO 中, 当 $|A| > 1$ 时, 灰狼种群在整个搜索域搜寻潜在猎物; 当 $|A| < 1$ 时, 灰狼种群将逐渐包围并捕获猎物。而 A 的值取决于收敛因子 a 的变化情况。

在 GWO 中, a 随迭代的更新方式采用一种线性递减策略, 可用等式 $a = 2(1 - t/T)$ 进行计算。在 Chiu 等^[20] 的研究中已经证明重要参量的不同更新

策略会极大影响算法的性能, 而且线性策略往往不是最有效的。因此, 本文提出了一种新的基于指数规律变化的收敛因子更新方式, 如式(8)所示。

$$a = 2e^{-t/T} \quad (8)$$

如图 2 所示, 线性表示 GWO 中所使用的收敛因子更新方式, 指数形式表示本文提出的新的收敛因子更新方式。

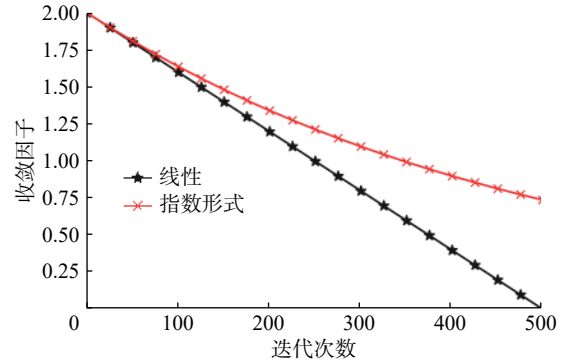


图 2 两种收敛因子

Fig.2 Two convergence factors

2.2 自适应位置更新策略

在 GWO 中, 初始化 α, β, δ 三组解会被记录并保留, 直至在迭代的过程出现适应值更好的个体代替它们为止。就是说, 如果在第 t 代, 种群中未出现优于记录的 α, β, δ 的解时, 当前的种群依然向着这三只狼更新位置。然而, 当这三者全部陷入局部最优时, 此时整个种群就难以寻求更好的解。可以理解为: 当狼群的决策者误判了猎物出现的地方, 那么, 所有灰狼的包围动作都将是无效的, 它们难以在错误的地方找到猎物。

本文提出了一种新的 β 与 δ 的定义方式以强化当前代最优个体的作用, 从而提升算法的全局搜索能力。在算法实施中, α 同 GWO 一样, 但是, β 与 δ 被定义为局部变量, 它们是第 t ($t = 1, 2, \dots, T$) 代中的适应度值最优和第二的灰狼个体。

同时, 为了更好地平衡算法的全局搜索和局部开发过程, 本文提出了一种新的自适应的移步策略, 其数学表达为

$$X(t+1) = \frac{(X_1 + X_2 + X_3)}{3} \left(1 - \frac{t}{T}\right) + X_1 \frac{t}{T} \quad (9)$$

2.3 修订的动态权重策略

在文献[17]中指出了 GWO 位置更新公式的缺点, X_1, X_2, X_3 求平均值的方式无法凸显 α, β, δ 三者之间的重要度, 为此, 作者提出了两种新的比例权重策略, 分别是加权平均策略和基于适应

度值的比例权重策略, 并进行了实验验证。文献 [21] 提出了一种基于指导位置向量权值的动态比例权重。文献 [22] 对不同的权重策略进行了分析与实验研究, 并从理论上证明了动态权重策略能高效寻优的原因。

当前灰狼个体到 α, β, δ 之间的距离权重

$$W_1 = \frac{|X_1|}{|X_1| + |X_2| + |X_3|} \tag{10}$$

$$W_2 = \frac{|X_2|}{|X_1| + |X_2| + |X_3|} \tag{11}$$

$$W_3 = \frac{|X_3|}{|X_1| + |X_2| + |X_3|} \tag{12}$$

最终, 灰狼位置更新公式可表示为

$$X(t+1) = \frac{|X_3|}{|X_1| + |X_2| + |X_3|} \tag{13}$$

然而, 在实际应用中, 式 (10)~(12) 很可能出现分母为 0 的情况。因此, 需添加一个很小的常数 ε , 取值为 10^{-16} 。它们被修改为

$$W_1 = \frac{|X_1|}{|X_1| + |X_2| + |X_3| + \varepsilon} \tag{14}$$

$$W_2 = \frac{|X_2|}{|X_1| + |X_2| + |X_3| + \varepsilon} \tag{15}$$

$$W_3 = \frac{|X_3|}{|X_1| + |X_2| + |X_3| + \varepsilon} \tag{16}$$

结合前面的自适应位置更新策略, 最终的灰狼位置更新方式可表示为

$$X(t+1) = \frac{W_1 X_1 + W_2 X_2 + W_3 X_3}{3} \left(1 - \frac{t}{T}\right) + X_1 \frac{t}{T} \tag{17}$$

2.4 MGWO 算法步骤

综合使用以上 3 种改进策略, 本文提出的改进灰狼优化算法 (MGWO) 步骤如下:

- a. 参数初始化。设定种群规模 N , 总迭代次数 T , 常数 ε ; 初始化 a, A, C 。
- b. 在待解问题的搜索空间内随机生成灰狼初始种群。
- c. 计算种群中所有灰狼个体的适应值, 选择适应度值最优的个体, 并记录其位置为 X_α 。令 $t = 1$, 迭代开始。
- d. 判断算法是否满足终止条件, 若满足, 则输出最优解; 否则, 执行步骤 e。
- e. 根据式 (8) 计算收敛因子 a 。
- f. 灰狼种群按适应度值排序, 排在前两位的位置分别被记录为 X_β 和 X_δ 。
- g. 根据式 (14)~(16) 计算动态比例权重, 灰

- 狼个体根据式 (17) 更新位置。
- h. 将超出搜索空间的灰狼个体放回搜索空间。
- i. 更新 X_α , 令 $t = t + 1$, 返回步骤 d。

3 实验与分析

为了实验验证提出的 3 种改进策略的有效性, 选取了国际上通用的 10 种基准测试函数进行仿真实验。实验运行环境为 Intel(R) CPU 3550M, 主频 2.30 GHz, 内存 4 GB, Windows 10 64 位操作系统, 所有代码通过 Python 3.7 进行编程。在此基础上, 进行了 3 组对比实验。为保证无偏性, 所有算法的种群数 N 均设定为 30, 总迭代次数设定为 500。为了排除随机性的影响, 所有实验独立运行 30 次, 取 30 次的平均值和标准差作为算法性能的度量标准。

3.1 基准测试函数

基准函数的数学表达如表 1 所示, 包括 7 个单峰测试函数 (F1~F7) 和 3 个多峰测试函数 (F8~F10)。单峰测试函数是算法求解精度和收敛速度的度量, 而多峰测试函数可以很好地表示算法的全局搜索能力和避免局部最优的能力。部分测试函数的 3 维图像如图 3 所示。

表 1 基准测试函数
Tab.1 Benchmark function

函数	函数表达式	维度	范围	解
F1	$f_1(x) = \sum_{i=1}^n X_i^2$	30	[-100,100]	0
F2	$f_2(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	30	[-10,10]	0
F3	$f_3(x) = \sum_{i=1}^n \left(\sum_{j=1}^i X_j \right)^2$	30	[-100,100]	0
F4	$f_4(x) = \max_i \{ x_i , 1 \leq i \leq n\}$	30	[-100,100]	0
F5	$f_5(x) = \sum_{i=1}^{n-1} \left[100(x_i + 1 - x_i^2)^2 + (x_i - 1)^2 \right]$	30	[-30,30]	0
F6	$f_6(x) = \sum_{i=1}^n ([x_i + 0.5])^2$	30	[-100,100]	0
F7	$f_7(x) = \sum_{i=1}^n i x_i^4 + \text{random}[0, 1]$	30	[-128,128]	0
F8	$f_8(x) = \sum_{i=1}^n [x_i^2 - 10 \cos(2\pi x_i) + 10]$	30	[-5.12,5.12]	0
F9	$f_9(x) = 20 \exp \left(-0.2 \frac{1}{n} \sum_{i=1}^n x_i^2 \right) - \exp \left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i) \right) + e + 20$	30	[-32,32]	0
F10	$f_{10}(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos \frac{x_i}{\sqrt{i}} + 1$	30	[-600,600]	0

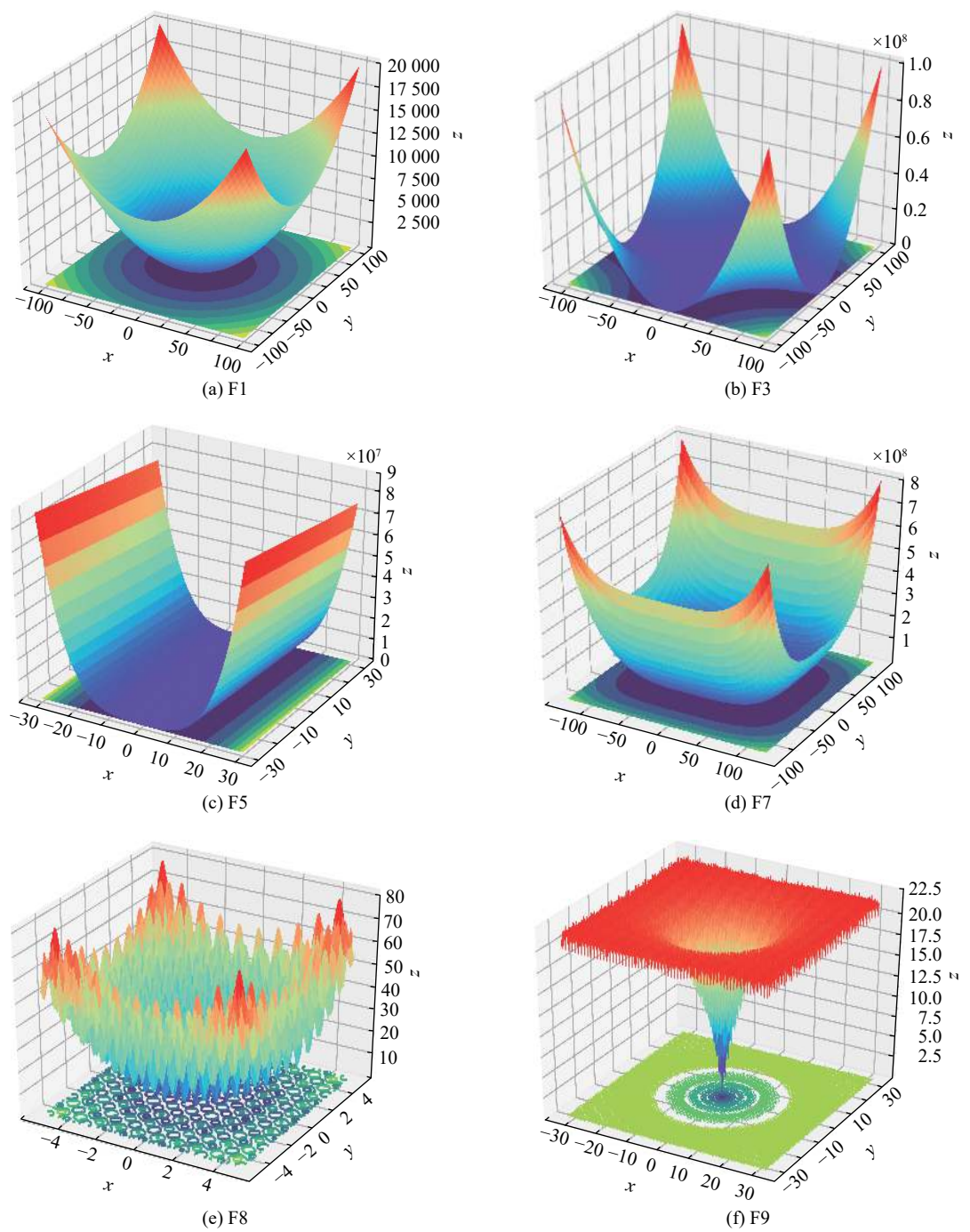


图 3 部分基准函数的二维搜寻空间

Fig.3 Two-dimensional search space of a part of benchmark function

3.2 与 GWO 和其他改进灰狼算法的比较

为了测试算法的搜寻性能,将提出的策略与其他文献的策略进行比较。各比较算法及其描述如表 2 所示。实验结果如表 3 所示。表 3 中列出了各算法在基准测试函数上的测试结果。表格中粗体显示的是最佳求解结果。表 2 中从 EGWO 到 GWO-Fuzzy 的实验结果直接来源于文献 [15-19]。其中,EGWO 的实验环境为 QT Creator 2.4.0; LGWO

实验运行环境为 Intel(R)Core i2, 主频 2 GHz, 内存 4 GB, Windows 7 64 位操作系统, 编程软件为 Matlab R2013a。

从表 3 可以看出,提出的 MGWO_1, MGWO_2 算法在函数 F1~F4 和 F6~F9 上均优于 GWO 算法,而且在大多数问题上略优于其他文献提出的改进灰狼算法,这说明了提出的改进策略的有效性。同时,依据数值结果, MGWO_2 相比 MGWO_1

表 2 对比算法及描述

Tab.2 Comparison algorithms and its description

算法	算法描述
GWO	标准灰狼算法
EGWO	文献[15]提出的增强型灰狼算法
LGWO	基于列维飞行和贪婪选择策略的灰狼算法 ^[16]
NGWO	文献[19]提出的基于非线性收敛因子的灰狼算法
GWO-weight	基于适应度值的比例权重策略的改进灰狼算法 ^[17]
GWO-Fuzzy	基于模糊层次算子的改进灰狼算法 ^[17]
GWO-EPD	基于动态种群的灰狼算法 ^[18]
MGWO_1	①指数规律收敛因子调整策略
MGWO_2	②自适应位置更新策略
MGWO_3	①和②
MGWO_4	①、②和修订的动态调整策略

获得更高的求解精度，这说明灰狼个体的自适应位置更新策略是一种更优的改进策略。

MGWO_3 综合使用了指数规律收敛因子调整策略和自适应位置更新策略，从结果可以看出，在函数 F1~F4 上，MGWO_3 明显优于除 MGWO_4 外的其他对比算法。在函数 F7 上，MGWO_3 排在 MGWO_4 和 NGWO 之后，胜过其他 8 种对比算法。在函数 F8 上，MGWO_3 仅次于 MGWO_4，NGWO 和 LGWO。在函数 F9 上，MGWO_3 仅次于 MGWO_4 和 LGWO，优于其他 8 种算法。但是，在 F6 和 F10 上，MGWO_3 的结果并不理想。

MGWO_4 结合了本文提出的 3 种改进策略，从表 3 可以看出，它在函数 F1，F3，F8 和 F10 上均收敛到理论最优值。在函数 F2，F4，F7 和 F9 上，其求解精度明显优于其他算法。然而，对于函数 F5 和 F6，MGWO_4 表现一般。

在所有测试函数上各算法 30 次独立运行的运行总时间列于表 4。从运行时间来看，在函数 F1~F4，F7 和 F8 上，GWO-EPD 所耗费的时间成本最小。在函数 F5 上，MGWO_2 的运行时间要

表 3 11 种对比算法在 10 个测试函数上的实验结果

Tab.3 Experimental results of 11 comparison algorithms on 10 test functions

函数		GWO	EGWO	LGWO	NGWO	GWO-weight	GWO-Fuzzy	GWO-EPD	MGWO_1	MGWO_2	MGWO_3	MGWO_4
F1	平均值	1.46E-27	1.17E-36	3.17E-30	1.16E-47	7.12E-31	9.47E-32	2.95E-20	7.97E-37	1.79E-40	4.96E-58	0
	标准差	2.68E-27	1.05E-36	4.07E-20	2.72E-47	2.40E-30	4.74E-31	4.09E-20	1.27E-36	3.65E-40	1.14E-57	0
F2	平均值	1.01E-16	2.84E-23	5.39E-19	2.92E-28	8.07E-19	1.73E-19	2.03E-12	4.21E-22	5.20E-25	3.65E-35	3.14E-205
	标准差	7.08E-17	1.47E-23	1.07E-02	4.16E-28	7.15E-19	3.50E-19	2.40E-12	2.18E-22	5.52E-25	6.61E-35	0
F3	平均值	1.48E-05	6.19E-05	8.12E-08	9.98E-12	3.30E-06	5.34E-01	6.21E-01	2.15E-07	1.48E-12	3.01E-16	0
	标准差	3.21E-05	1.21E-04	2.05E+00	1.99E-11	1.26E-05	3.96E+00	1.30E+00	6.10E-07	4.25E-12	1.50E-15	0
F4	平均值	6.13E-07	2.43E-08	1.17E-08	7.15E-13	1.66E-07	4.81E-08	9.20E-05	4.73E-09	1.12E-10	1.40E-15	1.23E-181
	标准差	5.34E-07	4.08E-08	1.32E+00	7.60E-13	4.00E-07	1.05E-07	8.56E-05	3.92E-09	1.95E-10	2.20E-15	0
F5	平均值	2.69E+01	4.86E+01	8.35E+00	2.61E+01	2.69E+01	5.15E+01	2.74E+01	2.71E+01	2.72E+01	2.72E+01	2.89E+01
	标准差	6.58E-01	9.44E+00	5.34E+00	3.96E-01	7.18E-01	1.61E+01	6.82E-01	8.11E-01	7.81E-01	7.26E-01	1.62E-01
F6	平均值	8.04E-01	0	2.69E-04	5.62E-01	7.56E-01	5.72E+00	1.01E+00	7.96E-01	9.01E-01	1.44E+00	5.58E+00
	标准差	3.12E-01	0	2.30E-05	2.32E-01	3.74E-01	3.25E+00	2.71E-01	3.06E-01	3.02E-01	3.67E-01	6.16E-01
F7	平均值	2.77E-03	4.27E-03	3.02E-03	1.05E-03	1.90E-03	1.70E-03	6.20E-03	2.22E-03	1.82E-03	1.35E-03	5.98E-05
	标准差	9.03E-04	1.50E-03	1.10E-03	2.32E-04	1.00E-03	9.21E-04	3.03E-03	1.24E-03	1.13E-03	9.05E-04	5.14E-05
F8	平均值	3.87E+00	1.35E+02	9.46E-02	0	2.12E+00	9.69E-01	1.35E+01	2.30E+00	8.92E-01	2.87E-01	0
	标准差	4.44E+00	1.27E+01	2.16E+01	0	4.81E+00	2.98E+00	2.70E+01	4.10E+00	2.53E+00	1.55E+00	0
F9	平均值	1.02E-13	1.72E+00	2.12E-15	1.05E-14	4.83E-14	2.38E-14	2.46E-11	3.27E-14	1.23E-14	6.78E-15	8.88E-17
	标准差	1.85E-14	0	4.30E-02	2.39E-15	7.76E-15	1.60E-14	1.53E-11	3.59E-15	3.38E-15	1.45E-15	1.07E-15
F10	平均值	2.47E-03	1.31E-08	2.42E-05	0	4.20E-03	7.29E-04	5.61E-03	2.60E-03	5.19E-03	5.72E-03	0
	标准差	5.07E-03	3.12E-08	8.39E-05	0	8.20E-03	3.00E-03	9.44E-03	5.92E-03	9.63E-03	9.01E-03	0

表 4 对比算法运行时间

Tab.4 Running time of the comparison algorithms

函数	GWO	GWO_EPD	MGWO_1	MGWO_2	MGWO_3	MGWO_4
F1	120	90	127	115	135	195
F2	122	94	114	128	143	203
F3	113	89	114	110	137	186
F4	111	104	107	106	129	180
F5	154	150	138	130	150	211
F6	96	104	146	113	135	210
F7	101	94	135	113	142	207
F8	126	125	165	151	194	223
F9	125	128	168	163	177	223
F10	128	132	173	154	189	216

小于其他的对比算法。在函数 F6, F9 和 F10 上, GWO 的运行时间最短, 优于其他算法。在函数 F7 和 F8 上, GWO_EPD 的运行时间依然具有竞争性。

综上所述, 本文提出的 3 种改进策略都是有效的。综合所有数值结果, MGWO_4 明显优于其他 10 种对比算法, 它具有很高的寻优能力, 获得了较高的求解精度, 同时从标准差来看, 具有较高的鲁棒性。MGWO_3, NGWO 和 LGWO 仅次于 MGWO_4, 它们在特定的问题上表现突出, 且都明显改进了 GWO 算法的性能。但是, 也应该认识到, MGWO_4 和 MGWO_3 高效寻优结果的代价是一定程度上时间成本的增加。而 MGWO_1 和 MGWO_2 可能是最佳选择, 它们对 GWO 的性能提升幅度不如 MGWO_4 和 MGWO_3 那么大, 但运行时间无明显变化。具体选择哪种算法, 使用者要视自己的需求情况而定, 如果对待求解问题的精度要求较高, MGWO_4 和 MGWO_3 算法显然是更好的选择。

3.3 与其他元启发式算法的比较

为进一步验证本文改进策略的有效性, 将表现较佳的 MGWO_3, MGWO_4 与 GA, TLBO, HHO 进行对比实验。其中, GA 和 TLBO 是经典的优化算法, HHO 是该领域最新的研究成果。其中, 对于 GA, 使用轮盘赌的染色体复制策略、均匀杂交策略和随机选择突变 3 种策略来进行编码, 染色体交配概率设置为 0.8, 突变概率设为 0.3。其余算法均只有种群规模和总迭代次数 2 个通用参数。实验结果如表 5 所示。

表 5 5 种对比算法在 10 个测试函数上的实验结果

Tab.5 Experimental results of 5 comparison algorithms on 10 test functions

函数		GA	TLBO	HHO	MGWO_3	MGWO_4
F1	平均值	9.95E+00	2.00E-107	3.47E-97	4.96E-58	0
	标准差	4.25E+00	3.81E-107	1.74E-96	1.14E-57	0
F2	平均值	1.33E+00	2.92E-53	1.64E-48	3.65E-35	3.14E-205
	标准差	2.39E-01	5.78E-53	8.62E-48	6.61E-35	0
F3	平均值	1.28E+04	1.09E-105	1.14E-76	3.01E-16	0
	标准差	2.75E+03	2.93E-105	6.11E-76	1.50E-15	0
F4	平均值	1.19E+01	1.11E-50	1.88E-57	1.40E-15	1.23E-181
	标准差	1.77E+00	1.66E-50	5.87E-57	2.20E-15	0
F5	平均值	4.70E+02	2.89E+01	6.34E-03	2.72E+01	2.89E+01
	标准差	2.36E+02	3.12E-02	1.65E-02	7.26E-01	1.62E-01
F6	平均值	1.04E+01	5.83E+00	9.09E-05	1.44E+00	5.58E+00
	标准差	4.72E+00	6.23E-01	2.15E-04	3.67E-01	6.16E-01
F7	平均值	6.42E+02	4.41E-04	3.87E-02	1.35E-03	5.98E-05
	标准差	6.50E+02	2.49E-04	2.73E-02	9.05E-04	5.14E-05
F8	平均值	5.30E+00	0	0	2.87E-01	0
	标准差	1.43E+00	0	0	1.55E+00	0
F9	平均值	1.83E+00	2.99E-15	4.40E-16	6.78E-15	8.88E-17
	标准差	2.73E-01	6.38E-16	0	1.45E-15	1.07E-15
F10	平均值	1.10E+00	0	0	5.72E-03	0
	标准差	3.51E-02	0	0	9.01E-03	0

从表 5 中的数值结果可知, 对于函数 F1 和 F3, MGWO_4 在 500 次迭代内收敛到理论最优值 0, 远远优于其他的优化算法; 对于函数 F8 和 F10, MGWO_4, TLBO 和 HHO 这 3 种算法均达到理论最优值 0, 但是, 从收敛曲线(图 4)可以看出, MGWO_4 的收敛速度明显高于 TLBO 和 HHO, 所以, 它依然是最佳算法; 对于函数 F2 和 F4, MGWO_4 分别得到了 3.14E-205 和 1.23E-181 的最优解, 其求解精度相当高, 远胜于其他的优化算法; 对于函数 F5 和 F6, HHO 算法取得了最好的寻优结果, 其次是 MGWO_3 与 MGWO_4; 对于函数 F7, MGWO_4 获得最佳求解结果, 其次是 TLBO 与 MGWO_3。值得注意的是, 对于 8 个基准函数(F1, F2, F3, F4, F7, F8, F9, F10), MGWO_4 不仅取得了最优的平均值, 而且其标准差也是最小的, 这说明了 MGWO_4 算法具有较高的鲁棒性。

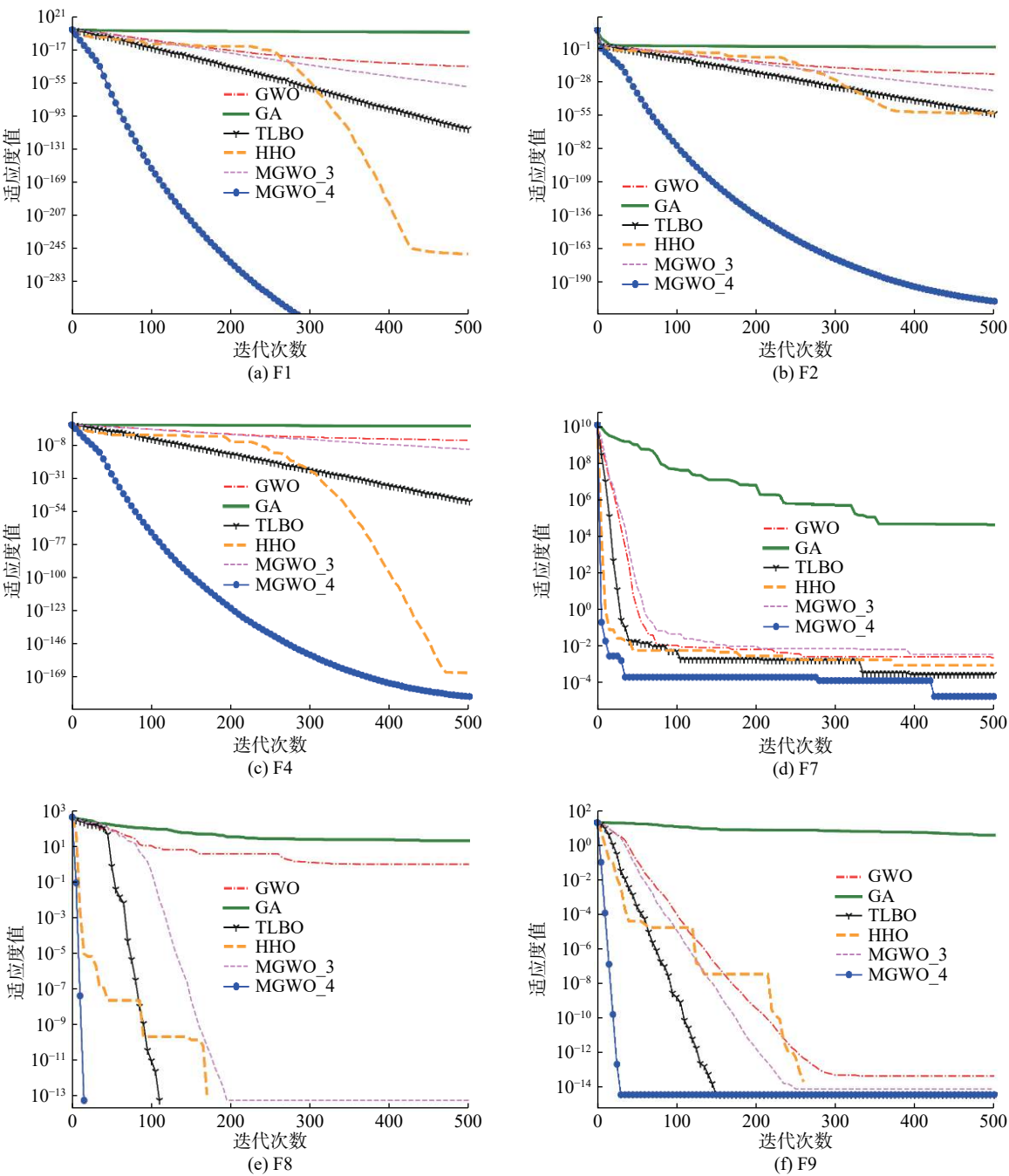


图 4 6 种比较算法在部分测试函数上的收敛曲线

Fig.4 Convergence curves of 6 comparison algorithms on some test functions

表 6 列出了 GA, TLBO, HHO, MGWO_3 和 MGWO_4 在 10 个测试函数上 30 次运行的总时间。从表 6 中的数据可以看出, TLBO 和 HHO 算法的求解时间明显优于其他的对比者, 特别是对于单峰测试函数, 这两种算法的运行时间很短。而 MGWO_3 和 MGWO_4 算法的运行时间相对较长。

6 种算法(GWO, GA, TLBO, HHO, MGWO_3 和 MGWO_4)在部分测试函数上的收敛曲线如图 4 所示。由图 4 可知, MGWO_3 提升了 GWO

的收敛速度和求解精度, MGWO_4 具有相当高的收敛速度和求解精度, 远优于 GWO 和其他的对比优化算法。

以上两组对比实验验证了本文提出的 3 种改进策略的实施效果, 从运行时间来看, MGWO_1 和 MGWO_2 在提升 GWO 的性能的基础上运行时间无明显变化, 而 MGWO_4 和 MGWO_3 虽然在求解精度和收敛速度上有大幅度的提升, 但带来了程序运行的时间成本的增加。综上所述, 可以

表 6 5 种对比算法在 10 个测试函数上的运行时间

Tab.6 Running time of the 5 comparison algorithms on 10 test functions

函数	GA	TLBO	HHO	MGWO_3	MGWO_4
F1	94	43	38	135	195
F2	100	52	47	143	203
F3	96	32	38	137	186
F4	88	30	34	129	180
F5	108	80	81	150	211
F6	96	44	50	135	210
F7	103	156	54	142	207
F8	118	124	113	194	223
F9	119	118	127	177	223
F10	126	125	137	189	216

看出本文提出的 MGWO 算法依然是一种更好的优化算法。

3.4 拉伸/压缩弹簧设计优化问题

以上两组对比实验所针对的问题都是无约束优化问题, 为进一步验证本文改进策略的有效性, 使用约束优化问题继续进行实验。拉伸/压缩弹簧设计优化问题是一个经典的带约束的工程优化问题, 该问题的目的是在满足 3 个设计变量的 4 个约束条件的情况下求得弹簧重量的最小值。3 个设计变量分别是线径 d , 平均线圈直径 D 和有效线圈数 N ; 4 个约束条件与剪应力、弹簧颤动频率和最小挠度相关。图 5 描述了这一问题和其设计变量, 该问题的数学表达如下:

设计变量

$$x = [x_1, x_2, x_3] = [d, D, N]$$

优化函数

$$f(x) = (x_3 + 2)x_2x_1^2$$

约束条件:

$$g_1(x) = 1 - \frac{x_2^3x_3}{71\,785x_1^4} \leq 0$$

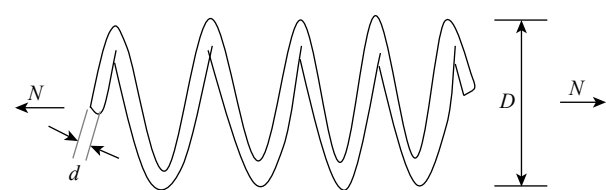


图 5 拉伸/压缩弹簧设计优化问题

Fig. 5 Tension/compression spring design problem

$$g_2(x) = \frac{4x_2^2 - x_1x_2}{12\,566(x_2x_1^3 - x_1^4)} + \frac{1}{5\,108x_1^2} - 1 \leq 0$$

$$g_3(x) = 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0$$

$$g_4(x) = \frac{x_1 + x_2}{1.5} - 1 \leq 0$$

设计变量范围:

$$0.05 \leq x_1 \leq 2.00, 0.25 \leq x_2 \leq 1.30, 2.00 \leq x_3 \leq 15.0$$

使用 7 种对比算法来求解这一约束优化问题, 算法在 30 次实验中所获得的最佳实验结果如表 7 所示。从表 7 中的数据可以看出, MGWO_2 取得了最好的结果, 优于其他对比算法。MGWO_1 和 MGWO_3 的数值结果也要优于 GWO 和其他算法。MGWO_4 虽然在无约束优化问题时表现突出, 但在求解拉伸/压缩弹簧设计优化问题时不如其他 3 种 MGWO 算法和 GWO 算法。

表 7 拉伸/压缩弹簧问题的结果比较

Tab.7 Comparison of results for the tension/compression spring problem

算法	d	D	N	最优重量
GWO	0.052571	0.377981	10.16171	0.012704484
GA	0.05540139	0.43405084	8.272921337	0.01368598
HHO	0.058305417	0.537772911	5.334416806	0.013408566
MGWO_1	0.051654329	0.355321844	11.39343483	0.012697766
MGWO_2	0.051495448	0.35201031	11.57779045	0.012674247
MGWO_3	0.051410392	0.349747548	11.73754971	0.012698891
MGWO_4	0.05	0.315418406	14.30511153	0.012857331

4 结束语

针对基本 GWO 求解精度不高、收敛速度较慢和易于陷入局部最优的问题, 提出了一种改进的灰狼优化算法。首先, 提出了一种收敛因子指数变化规律的修正策略, 处于这种策略下, a 在迭代末期并不会减小为 0, 实验结果验证了其有效性。其次, 通过重新定义 β 和 δ 而强化每代最佳搜寻个体的作用, 进而提出了一种灰狼个体自适应位置更新策略。此外, 引入动态权重策略, 并对其进行了修订。接着, 针对无约束优化问题的对比测试实验证明了改进策略的有效性, 尤其是综合使用 3 种改进策略的 MGWO_4 具有较强的局部最优避免性、较快的收敛速度、较高的求解精度, 它

明显提升了基本 GWO 的性能,更好地平衡了算法的全局搜索性和局部开发性。最后,对于工程设计约束优化问题的实验结果进一步证明了本文提出的 MGWO 的有效性。

未来的研究思路是编写二进制的 MGWO,并将 MGWO 运用到更多的实际问题中。

参考文献:

- [1] KIRKPATRICK S, GELATT JR J C, VECCHI M P. Optimization by simulated annealing[J]. *Science*, 1983, 220(4598): 671–680.
- [2] MÜHLENBEIN H, SCHLIERKAMP-VOOSEN D. Predictive models for the breeder genetic algorithm I. Continuous parameter optimization[J]. *Evolutionary Computation*, 1993, 1(1): 25–49.
- [3] EBERHART R C, SHI Y. Guest editorial special issue on particle swarm optimization[J]. *IEEE Transactions on Evolutionary Computation*, 2004, 8(3): 201–203.
- [4] DORIGO M, DI CARO G. Ant colony optimization: a new meta-heuristic[C]//*Proceedings of 1999 Congress on Evolutionary Computation*. Washington: IEEE, 2002: 6–9.
- [5] RASHEDI E, NEZAMABADI-POUR H, SARYAZDI S. GSA: a gravitational search algorithm[J]. *Information Sciences*, 2009, 179(13): 2232–2248.
- [6] BIDAR M, KANAN H R, MOUHOUB M, et al. Mushroom reproduction optimization (MRO): a novel nature-inspired evolutionary algorithm[C]//*Proceedings of 2018 IEEE Congress on Evolutionary Computation (CEC)*. Rio de Janeiro, Brazil: IEEE, 2018, 1762–1771.
- [7] KARABOGA D, BASTURK B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm[J]. *Journal of Global Optimization*, 2007, 39(3): 459–471.
- [8] RAO R V, SAVSANI V J, VAKHARIA D P. Teaching-learning-based optimization: a novel method for constrained mechanical design optimization problems[J]. *Computer-Aided Design*, 2011, 43(3): 303–315.
- [9] HEIDARI A A, MIRJALILI S, FARIS H, et al. Harris hawks optimization: algorithm and applications[J]. *Future Generation Computer Systems*, 2019, 97: 849–872.
- [10] WOLPERT D H, MACREADY W G. No free lunch theorems for optimization[J]. *IEEE Transactions on Evolutionary Computation*, 1997, 1(1): 67–82.
- [11] MIRJALILI S, MIRJALILI S M, LEWIS A. Grey wolf optimizer[J]. *Advances in Engineering Software*, 2014, 69: 46–61.
- [12] 谭念, 王学顺, 黄安民, 等. 基于灰狼算法 SVM 的 NIR 杉木密度预测 [J]. *林业科学*, 2018, 54(12): 137–141.
- [13] 姚鹏, 王宏伦. 基于改进流体扰动算法与灰狼优化的无人机三维航路规划 [J]. *控制与决策*, 2016, 31(4): 701–708.
- [14] 王书芹, 华钢, 郝国生, 等. 基于灰狼优化算法的长短期记忆网络在时间序列预测中的应用 [J]. *中国科技论文*, 2017, 12(20): 2309–2314.
- [15] JOSHI H, ARORA S. Enhanced grey wolf optimization algorithm for global optimization[J]. *Fundamenta Informaticae*, 2017, 153(3): 235–264.
- [16] 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.
- [17] RODRÍGUEZ L, CASTILLO O, SORIA J, et al. A fuzzy hierarchical operator in the grey wolf optimizer algorithm[J]. *Applied Soft Computing*, 2017, 57: 315–328.
- [18] SAREMI S, MIRJALILI S Z, MIRJALILI S M. Evolutionary population dynamics and grey wolf optimizer[J]. *Neural Computing and Applications*, 2015, 26(5): 1257–1263.
- [19] 王敏, 唐明珠. 一种新型非线性收敛因子的灰狼优化算法 [J]. *计算机应用研究*, 2016, 33(12): 3648–3653.
- [20] CHIU C Y, SHIH P C, LI X C. A dynamic adjusting novel global harmony search for continuous optimization problems[J]. *Symmetry*, 2018, 10(8): 337.
- [21] 郭振洲, 刘然, 拱长青, 等. 基于灰狼算法的改进研究 [J]. *计算机应用研究*, 2017, 34(12): 3603–3606, 3610.
- [22] 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.

(编辑: 石 瑛)