

融合多策略的黄金正弦灰狼优化算法

陈 杰, 陈岁繁, 李其朋

(浙江科技学院 机械与能源工程学院, 杭州 310023)

摘 要: 【目的】为解决灰狼优化算法(grey wolf optimization, GWO)收敛精度不高, 收敛速度较慢和易陷入局部最优等不足, 提出一种融合多策略的黄金正弦灰狼优化算法(golden sine grey wolf optimization, G-GWO)。【方法】首先, 利用非线性调整收敛因子、动态调整比例权重和引入黄金正弦策略对 GWO 算法进行改进; 然后, 选取三类基准测试函数进行寻优实验, 并与 GWO 算法、其他智能优化算法和其他改进 GWO 算法进行对比, 从寻优的收敛精度、鲁棒性和收敛速度方面验证 G-GWO 算法的优越性; 最后, 建立板料冲压成形工艺参数与质量参数的 BP 神经网络(BP neural network, BPNN)代理模型, 选用 8 种算法分别优化 BP 神经网络的权值和阈值, 对比优化后的代理模型精度, 验证 G-GWO 算法在实际工程应用中的有效性。【结果】G-GWO 算法在三类基准测试函数的收敛精度、鲁棒性和收敛速度较其他算法均有较大优势, 优化后的代理模型最大减薄率相对误差为 3.47%, 最大增厚率相对误差为 4.99%。【结论】改进策略能提高 GWO 算法的性能, 这可作为建立高精度代理模型和后续的板料冲压工艺参数优化的参考。

关键词: 灰狼优化算法; 收敛因子; 比例权重; 黄金正弦; 代理模型

中图分类号: TP301.6

文献标志码: A

文章编号: 1671-8798(2023)06-0514-13

Golden sine grey wolf optimization algorithm integrating multiple strategies

CHEN Jie, CHEN Suifan, LI Qipeng

(School of Mechanical and Energy Engineering, Zhejiang University of
Science and Technology, Hangzhou 310023, Zhejiang, China)

Abstract: [Objective] Aiming at the defects of grey wolf optimization (GWO) algorithm, such as low convergence accuracy, slow convergence speed and proneness to convergence at local optimum, a golden sine grey wolf optimization (G-GWO) algorithm integrating multiple strategies was proposed. [Method] Firstly, the GWO algorithm was improved by nonlinearly

收稿日期: 2022-11-24

基金项目: 国家重点研发计划(科技助力经济 2020)项目(SQ2020YFF0423771); 浙江省科技计划项目(2020C01053, 2022C04022)

通信作者: 李其朋(1977—), 男, 山东省临邑人, 教授, 博士, 主要从事智能制造、数字孪生等研究。E-mail: liqipeng@zust.edu.cn。

adjusting the convergence factor, dynamically adjusting the proportional weights and introducing the golden sine strategy; then, three types of benchmark test functions were selected for the optimization search experiments and compared with the GWO algorithm, other intelligent optimization algorithms and other improved GWO algorithms to verify the superiority of the G-GWO algorithm in terms of convergence accuracy, robustness and convergence speed of the optimization search; finally, the BP neural network (BPNN) surrogate model of sheet metal stamping forming process parameters and quality parameters was established, and eight algorithms were selected to optimize the weights and thresholds of the BP neural network respectively, and the accuracy of the optimized surrogate model was compared to verify the effectiveness of the G-GWO algorithm in practical engineering applications. [Result] The convergence accuracy, robustness and convergence speed of the G-GWO algorithm in the three types of benchmark test functions are superior to other algorithms, and the relative error of the maximum thinning rate and the maximum thickening rate of the optimized surrogate model are 3.47% and 4.99%, respectively. [Conclusion] The improvement strategies can improve the performance of the GWO algorithm, which can provide a reference for establishing a high-precision surrogate model and subsequent optimization of the sheet stamping process parameters.

Keywords: grey wolf optimization algorithm; convergence factor; proportional weight; golden sine; surrogate model

灰狼优化算法(grey wolf optimization,GWO)是由 Seedily Mirjalil 等人提出的一种群体智能优化算法^[1],具有原理简单、调整参数少和易于实现等优点,被广泛应用于电力系统优化^[2]、车间调度^[3]、图像处理^[4]、机器人路径规划^[5]、特征选择^[6]等研究领域。但 GWO 算法存在收敛精度不高、收敛速度较慢和因搜索后期种群多样性的减少容易陷入局部最优等问题^[7],为了提高 GWO 算法的性能,许多研究者提出了不同的改进意见。Liu 等^[8]提出了一种具有伪节点交叉操作和自适应控制参数 a 的改进算法,提高了算法在跳出局部最优和搜索精度上的优势。Gao 等^[9]引入自然对数的底数 e 建立非线性收敛因子和可变权重策略,提高了算法的搜索能力和降低陷入局部最优的概率。Rodríguez 等^[10]提出基于加权平均和不同的适应度加权以及模糊权重的不同分层运算变体,验证了分层算子模糊变量的 GWO 算法性能最好。Xu 等^[11]使用基于混沌映射的动态权重来引导灰狼进行搜索,克服了过早收敛到局部最优的缺点。韩太林等^[12]提出改变候选解策略和动态权重因子的改进 GWO 算法,提高了灰狼个体多样性,降低了陷入局部最优的风险。Meng 等^[13]结合交叉算子和贪婪机制改进 GWO 算法,其中水平交叉算子细化首领狼,增强全局探索能力;垂直交叉算子保持种群的多样性,防止陷入局部最优中。Long 等^[14]基于物理学中的折射原理,提出了折射学习的新算子,有助于 GWO 算法跳出局部最优。Yu 等^[15]将佳点集初始化策略和甲虫天线搜索机制引入 GWO 算法中,提高了算法初始化分散能力和全局探索能力。Mohammed 等^[16]将 GWO 算法和鲸鱼算法(whale optimization algorithm,WOA)相结合,协调算法全局探索和局部开发的平衡。

上述研究成果虽能够改善 GWO 算法的性能,但算法收敛精度不高和搜索后期易陷入局部最优的问题依旧存在。为进一步提高算法的收敛精度和收敛速度,解决搜索后期易陷入局部最优等问题,本研究提出一种融合多策略的黄金正弦灰狼优化算法(golden sine grey wolf optimization,G-GWO)。通过对三类基准测试函数的寻优测试结果对比,以及对 Wilcoxon 秩和检验的统计结果分析,验证了 G-GWO 算法在收敛精度和鲁棒性方面有显著的提高;通过对 BP 神经网络(BP neural network,BPNN)建立板料工艺参数与质量参数的代理模型精度不足问题的优化结果对比表明,G-GWO 算法在实际应用中较其他算法

具有一定的优势。

1 GWO 算法原理

GWO 算法在求解优化问题时,设灰狼种群数为 N ,搜索的空间维度为 D ,第 i 只灰狼在 D 维空间中的个体位置为 $\mathbf{X}_i = \{\mathbf{X}_{i,1}, \mathbf{X}_{i,2}, \dots, \mathbf{X}_{i,D}\}$, $i=1,2,\dots,N$ 。将种群中的最优解记为 α ,次优解记为 β ,次次优解记为 δ ,其余解为 ω 。

灰狼在狩猎过程中包围猎物,对包围行为进行数学建模,提出了以下等式:

$$\mathbf{D} = |\mathbf{C}\mathbf{X}_p(t) - \mathbf{X}(t)|. \quad (1)$$

$$\mathbf{X}(t+1) = \mathbf{X}_p(t) - \mathbf{A}\mathbf{D}. \quad (2)$$

式(1)和(2)中: t 为当前迭代次数; \mathbf{A} 和 \mathbf{C} 均为系数向量; $\mathbf{X}_p(t)$ 为猎物的位置向量; $\mathbf{X}(t)$ 为当前灰狼的位置向量; \mathbf{D} 为当前灰狼与猎物的距离; $\mathbf{X}(t+1)$ 为灰狼位置的更新。

系数向量 \mathbf{A} 和 \mathbf{C} 的表达式为

$$\mathbf{A} = 2a\mathbf{r}_1 - a. \quad (3)$$

$$\mathbf{C} = 2\mathbf{r}_2. \quad (4)$$

式(3)和(4)中: $\mathbf{r}_1, \mathbf{r}_2$ 为 $[0,1]$ 中的随机向量; a 为影响 \mathbf{A} 变化的收敛因子,表达式为

$$a = 2 \left(1 - \frac{t}{t_{\max}} \right). \quad (5)$$

式(5)中: t 为当前迭代次数; t_{\max} 为迭代的最大次数。

在狩猎过程中,利用 α 狼、 β 狼和 δ 狼的不同位置来预估猎物的位置,数学模型如下:

$$\begin{cases} \mathbf{D}_\alpha = |\mathbf{C}_1\mathbf{X}_\alpha - \mathbf{X}|; \\ \mathbf{D}_\beta = |\mathbf{C}_2\mathbf{X}_\beta - \mathbf{X}|; \\ \mathbf{D}_\delta = |\mathbf{C}_3\mathbf{X}_\delta - \mathbf{X}|. \end{cases} \quad (6)$$

式(6)中: $\mathbf{D}_\alpha, \mathbf{D}_\beta$ 和 \mathbf{D}_δ 分别为当前灰狼趋向于 α, β, δ 狼之间的近似距离; $\mathbf{C}_1, \mathbf{C}_2$ 和 \mathbf{C}_3 均为式(4)中定义的系数向量。

每只灰狼按照以下方式更新自己的位置:

$$\begin{cases} \mathbf{X}_1 = \mathbf{X}_\alpha - \mathbf{A}_1\mathbf{D}_\alpha; \\ \mathbf{X}_2 = \mathbf{X}_\beta - \mathbf{A}_2\mathbf{D}_\beta; \\ \mathbf{X}_3 = \mathbf{X}_\delta - \mathbf{A}_3\mathbf{D}_\delta. \end{cases} \quad (7)$$

$$\mathbf{X}(t+1) = \frac{\mathbf{X}_1 + \mathbf{X}_2 + \mathbf{X}_3}{3}. \quad (8)$$

式(7)和(8)中: $\mathbf{X}_\alpha, \mathbf{X}_\beta, \mathbf{X}_\delta$ 分别为 α, β, δ 狼的当前位置; $\mathbf{X}_1, \mathbf{X}_2, \mathbf{X}_3$ 分别为 ω 狼向 α, β, δ 狼的前进步长和方向; $\mathbf{X}(t+1)$ 为 ω 狼的最后位置。

2 G-GWO 算法

2.1 收敛因子非线性调整策略

GWO 算法的探索 and 开发能力很大程度上依赖于 \mathbf{A} 的取值,当 $|\mathbf{A}| > 1$ 时,灰狼群体通过扩大搜索范围继续搜寻猎物,即全局搜索;而当 $|\mathbf{A}| < 1$ 时,灰狼群体则通过收缩搜索范围来进攻猎物,即局部开发。由式(3)可知, \mathbf{A} 的取值取决于收敛因子 a 的变化。因此,收敛因子 a 的取值影响着算法的全局搜索和局部开发能力。

由式(5)可知,GWO 算法中收敛因子 a (传统 a) 由 2 线性递减至 0,无法较好地实现“搜索前期应保持在大范围的搜索空间中搜索,后期则快速收敛至猎物位置”的目标^[17]。为在迭代前期扩大搜索范围,迭代后期快速收敛,笔者提出一种新的非线性变化控制参数改进收敛因子 a (改进 a),改进 a 的表达式如

式(9),改进 a 和传统 a 的变化对比如图 1 所示。

$$a = 2 \times \left[1 - \left(\frac{t}{t_{\max}} \right)^2 \right]^{\frac{1}{2}} \quad (9)$$

2.2 基于步长欧氏距离原理的比例权重动态调整策略

GWO 算法将三类首领狼(α 狼、 β 狼和 δ 狼)设置为同等重要程度,静态更新 ω 狼的位置(如式(8)所示),虽然这种位置更新方式对大多数常规问题有效,但无法满足高维、复杂的多模态问题求解。因此,笔者提出了基于步长欧氏距离原理的比例权重动态调整策略,改进后的比例权重动态调整策略表达式如式(10)和式(11)所示。

若 $|\mathbf{X}_2| > |\mathbf{X}_3| > |\mathbf{X}_1|$ 或 $|\mathbf{X}_3| > |\mathbf{X}_2| > |\mathbf{X}_1|$ 或 $|\mathbf{X}_3| > |\mathbf{X}_1| > |\mathbf{X}_2|$, 则

$$\begin{cases} W_1 = \frac{|\mathbf{X}_1| + |\mathbf{X}_2| + |\mathbf{X}_3|}{|\mathbf{X}_1|}; \\ W_2 = \frac{|\mathbf{X}_1| + |\mathbf{X}_2| + |\mathbf{X}_3|}{|\mathbf{X}_2|}; \\ W_3 = \frac{|\mathbf{X}_1| + |\mathbf{X}_2| + |\mathbf{X}_3|}{|\mathbf{X}_3|}; \\ \mathbf{X}_i(t+1) = \frac{W_1 \mathbf{X}_1 + W_2 \mathbf{X}_2 + W_3 \mathbf{X}_3}{W_1 + W_2 + W_3}. \end{cases} \quad (10)$$

若 $|\mathbf{X}_1| > |\mathbf{X}_2| > |\mathbf{X}_3|$ 或 $|\mathbf{X}_1| > |\mathbf{X}_3| > |\mathbf{X}_2|$ 或 $|\mathbf{X}_2| > |\mathbf{X}_1| > |\mathbf{X}_3|$, 则

$$\begin{cases} W_1 = \frac{|\mathbf{X}_1|}{|\mathbf{X}_1| + |\mathbf{X}_2| + |\mathbf{X}_3|}; \\ W_2 = \frac{|\mathbf{X}_2|}{|\mathbf{X}_1| + |\mathbf{X}_2| + |\mathbf{X}_3|}; \\ W_3 = \frac{|\mathbf{X}_3|}{|\mathbf{X}_1| + |\mathbf{X}_2| + |\mathbf{X}_3|}; \\ \mathbf{X}_i(t+1) = \frac{W_1 \mathbf{X}_1 + W_2 \mathbf{X}_2 + W_3 \mathbf{X}_3}{3}. \end{cases} \quad (11)$$

在式(10)和式(11)中,根据 α 、 β 和 δ 狼相对 ω 狼的位置分别计算三类首领狼的步长欧氏距离 \mathbf{X}_1 、 \mathbf{X}_2 和 \mathbf{X}_3 ,动态调整 ω 狼的位置更新比例权重 W_1 、 W_2 和 W_3 ,使每次迭代过程中,离 ω 狼步长距离最近的 α 狼在下一步的位置更新过程中占有较大的权重,之后是离 ω 狼步长距离较近的 β 狼,并以离 ω 狼步长距离较远的 δ 狼的最低权重结束。

2.3 黄金正弦策略

针对 GWO 算法在搜索后期由于种群多样性的减少而易陷入局部最优的问题,将黄金正弦策略^[18]融合到 GWO 算法中,以降低算法搜索后期陷入局部最优的可能性。黄金正弦策略通过使用正弦函数对单位圆扫描来模拟搜索空间,并结合黄金分割系数进行迭代搜索的方式,更好地遍历算法搜索空间,防止算法陷入局部最优值。引入黄金正弦策略后的位置更新过程如下:

首先,将给定的 N 只狼在范围 $[l_b, u_b]$ 中随机分布:

$$\mathbf{X}_{ij} = l_b + r_3(u_b - l_b), i \in [1, N], j \in [1, D]. \quad (12)$$

式(12)中: r_3 为 $[0, 1]$ 范围内的随机数。在第 t 次迭代中,第 i 只狼所在的位置为 $\mathbf{X}_i = \{\mathbf{X}_{i,1}, \mathbf{X}_{i,2}, \dots, \mathbf{X}_{i,D}\}$,其中 D 为问题的维数;之后将所有的灰狼存放在 N 行 D 列的矩阵 \mathbf{P} 中。

其次,当所有灰狼完成随机分布后,引入黄金正弦策略计算在 $t+1$ 次迭代中第 i 只狼的位置,其位置更新为

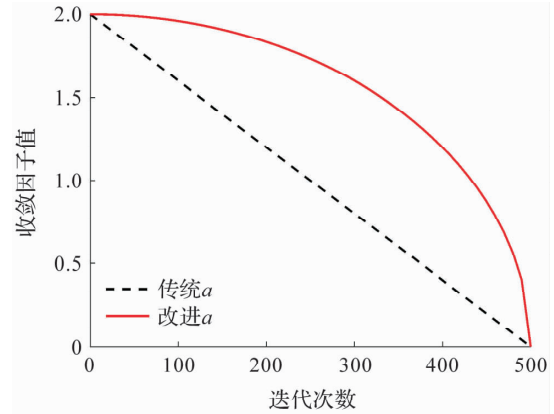


图1 改进 a 与传统 a 的变化对比

Fig. 1 Comparison of changes between improved a and traditional a

$$\mathbf{X}_{iG}(t+1) = \mathbf{X}_i(t) |\sin r_4| + r_5 \sin r_4 |x_1 \mathbf{P}_i(t) - x_2 \mathbf{X}_i(t)|. \quad (13)$$

式(13)中: r_4 为 $[0, 2\pi]$ 范围内的随机数; r_5 为 $[0, \pi]$ 范围内的随机数; $\mathbf{P}_i(t)$ 为在第 t 次迭代中第 i 只狼的最优位置; x_1 和 x_2 为黄金分割系数, 其表达式为

$$\begin{cases} x_1 = a\tau + b(1-\tau); \\ x_2 = a(1-\tau) + b\tau. \end{cases} \quad (14)$$

式(14)中: $\tau = (\sqrt{5} - 1)/2$ 为黄金分割数; a 和 b 的初始默认值分别为 $-\pi$ 和 π 。

然后, 对式(10)、式(11)和式(13)中所得到的两个更新位置预备解 $\mathbf{X}_i(t+1)$ 和 $\mathbf{X}_{iG}(t+1)$ 进行比较。

$$\mathbf{X}(t+1) = \begin{cases} \mathbf{X}_i(t+1), f(\mathbf{X}_i(t+1)) < f(\mathbf{X}_{iG}(t+1)); \\ \mathbf{X}_{iG}(t+1), f(\mathbf{X}_i(t+1)) \geq f(\mathbf{X}_{iG}(t+1)). \end{cases} \quad (15)$$

最后, 将式(15)得到的下一个解的位置 $\mathbf{X}(t+1)$ 同 $\mathbf{X}(t)$ 进行比较, 若新位置的适应度值比 $\mathbf{X}(t)$ 优, 则按照 $\mathbf{X}(t+1)$ 进行位置更新; 反之则保持不动。

2.4 算法步骤

综上所述, 将收敛因子非线性调整策略、比例权重动态调整策略和黄金正弦策略融合, 形成融合多策略的黄金正弦灰狼优化算法(G-GWO), 算法步骤总结如下:

- 1) 设置种群规模数 N , 最大迭代次数 t_{\max} , 搜索维度 D ; 生成 a 、 \mathbf{A} 和 \mathbf{C} 等参数, 随机初始化灰狼种群。
- 2) 计算种群中所有灰狼个体的适应度值, 并进行排序, 记录排列前三的适应度并记录位置为 \mathbf{X}_α 、 \mathbf{X}_β 和 \mathbf{X}_δ 。
- 3) 通过式(9)更新非线性收敛因子 a , 式(3)和式(4)更新系数向量 \mathbf{A} 和 \mathbf{C} 的值。
- 4) 通过式(6)和式(7)及步长欧氏距离动态调整策略式(10)和式(11)建立第一个预备解 $\mathbf{X}_i(t+1)$ 。
- 5) 通过黄金正弦策略的式(13)建立另一个预备解 $\mathbf{X}_{iG}(t+1)$, 通过式(15)比较两个预备解, 选择较优的值与 $\mathbf{X}(t)$ 比较, 择优作为灰狼的位置更新。
- 6) 判断是否满足结束条件, 若达到预定的最大迭代次数 t_{\max} , 则停止计算; 否则重复循环跳转至 2)。
- 7) 输出全局最优解。

3 试验仿真结果与分析

3.1 试验环境及参数设置

为测试 G-GWO 算法的性能, 选择文献[1]中的单峰、多峰和固定维度多峰三类基准测试函数进行仿真试验, 三类基准测试函数见表 1。试验运行环境为 Intel(R)Core(TM)i7-7770HQ CPU, 主频 2.80 GHz, 内存 8 GB, Win10 64 位操作系统; 运行软件为 MatLabR2018b。

为保证试验的无偏性, 灰狼总个数设定为 30, 总迭代次数设置为 500 次, 所有试验独立运行 30 次, 通过试验比较得到的平均值和标准差值。

表 1 测试函数

Table 1 Test functions

类型	函数	维数	函数表达式	最优值
单峰	f_1	30	$f(x) = \sum_{i=1}^n x_i^2$	0
	f_2	30	$f(x) = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	0

表 1(续)

类型	函数	维数	函数表达式	最优值
多峰	f_3	30	$f(x) = \sum_{i=1}^n -x_i \sin(\sqrt{ x_i })$	-12 569.487
	f_4	30	$f(x) = \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10]$	0
	f_5	30	$f(x) = -20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	0
固定维 度多峰	f_6	4	$f(x) = \sum_{i=1}^{11} \left[a_i - \frac{x_i(b_i^2 + b_i x_2)}{b_i^2 + b_i x_3 + x_4} \right]^2$	0.000 30
	f_7	2	$f(x) = \left(x_2 - \frac{5.1}{4\pi^2}x_1^2 + \frac{5}{\pi}x_1 - 6\right)^2 + 10\left(1 - \frac{1}{8\pi}\right)\cos x_1 + 10$	0.398
	f_8	2	$f(x) = [1 + (x_1 + x_2 + 1)^2(19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2)] \times [30 + (2x_1 + 3x_2)^2(18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2)]$	3
	f_9	3	$f(x) = -\sum_{i=1}^4 c_i \exp\left(-\sum_{j=1}^3 a_{ij}(x_j - p_{ij})^2\right)$	-3.86

3.2 不同改进策略寻优性能对比

为验证改进策略的有效性,将 GWO 算法与仅采用非线性收敛因子调整策略的 GWO1 算法、仅采用动态比例权重调整策略的 GWO2 算法、仅引入黄金正弦策略的 GWO3 算法和 G-GWO 算法进行比较。测试本研究提出的三种不同的改进策略单独作用时的情况,使用三类基准测试函数进行仿真试验,结果见表 2,其中平均值和标准差的较优值加粗展示(下同)。

表 2 不同改进策略对算法的影响比较

Table 2 Comparison of influences among different improvement strategies on algorithms

函数	GWO		GWO1		GWO2	
	平均值	标准差	平均值	标准差	平均值	标准差
f_1	1.038 6e-27	1.330 1e-27	1.830 2e-43	2.076 6e-43	5.817 3e-304	0.000 0e+00
f_2	1.122 5e-16	1.486 0e-16	7.427 6e-26	8.383 8e-26	1.289 3e-159	1.136 0e-159
f_3	-6.008 0e+03	9.957 2e+02	-5.276 4e+03	1.148 8e+03	-2.271 7e+03	4.902 8e+02
f_4	2.630 4e+00	4.006 5e+00	1.894 8e-15	1.037 8e-14	0.000 0e+00	0.000 0e+00
f_5	1.015 5e-13	1.753 8e-14	1.320 4e-14	2.757 2e-15	3.967 2e-15	1.228 3e-15
f_6	3.756 0e-03	7.562 3e-03	5.801 0e-03	8.939 8e-03	6.420 3e-04	1.355 6e-04
f_7	3.978 9e-01	2.301 9e-04	3.981 0e-01	1.967 2e-04	3.979 0e-01	1.200 0e-04
f_8	3.000 0e+00	3.102 9e-05	3.000 0e+00	3.675 5e-05	3.000 0e+00	4.293 3e-07
f_9	-3.861 0e+00	2.862 3e-03	-3.860 9e+00	2.538 9e-03	-3.842 6e+00	5.141 9e-02
函数	GWO3		G-GWO			
	平均值	标准差	平均值	标准差		
f_1	4.848 5e-267	0.000 0e-00	0.000 0e-00	0.000 0e+00		
f_2	2.971 3e-137	1.184 7e-137	1.948 9e-217	0.000 0e+00		
f_3	-1.255 4e+04	7.690 3e+02	-1.256 7e+04	5.123 1e+00		
f_4	0.000 0e+00	0.000 0e+00	0.000 0e+00	0.000 0e+00		
f_5	8.881 8e-16	0.000 0e+00	8.881 8e-16	0.000 0e+00		
f_6	3.158 8e-04	1.214 0e-05	3.335 8e-04	2.406 3e-05		
f_7	3.979 0e-01	9.346 4e-05	3.980 4e-01	1.494 7e-04		
f_8	3.000 0e+00	9.965 9e-06	3.000 0e+00	7.483 6e-06		
f_9	-3.862 7e+00	7.282 5e-05	-3.860 3e+00	1.892 9e-03		

由表 2 可知,对于单峰函数 f_1 和 f_2 ,GWO1、GWO2 和 GWO3 算法的平均值和标准差均比 GWO 算法更接近于理论最优值 0。对于多峰函数 f_3 、 f_4 和 f_5 ,GWO1 和 GWO2 算法在 f_4 和 f_5 的平均值和标

准差值比 GWO 算法更接近于理论最优值, GWO3 算法的平均值和标准差均比 GWO 算法更接近于理论最优值(f_3 的最优平均值为 $-12\,569.487$, f_4 与 f_5 的最优平均值为 0)。对于固定维度多峰函数 f_6 、 f_7 、 f_8 和 f_9 , GWO3 算法在 f_6 的平均值和标准差结果最优(f_6 的最优平均值为 $0.000\,30$), GWO、GWO1、GWO2 和 GWO3 算法在 f_7 和 f_9 平均值和标准差值较为接近, 在 f_8 中均收敛到理论最优值 3。融合三种策略的 G-GWO 算法除了在多峰函数 f_6 的平均值仅次于 GWO3 算法外(优于其他算法), 对其他 8 个测试函数的收敛精度均为最优, 且在单峰函数 f_1 和多峰函数 f_4 收敛到理论最优值 0, 在固定维度多峰函数 f_8 收敛到理论最优值 3, 相较于 GWO 算法而言, G-GWO 算法的标准差值更小, 对三类基准测试函数的鲁棒性均有明显提升。由此可见: 在对提升单峰函数、多峰函数与固定维度多峰函数的收敛精度和鲁棒性上, 三种改进策略均是有效的。

3.3 与其他智能优化算法对比

将 G-GWO 算法分别与黄金正弦算法(golden sine algorithm, Golden-SA)^[18]、WOA 算法^[19] 和粒子群算法(particle swarm optimization, PSO)^[20] 进行比较, 其中设置 PSO 参数为 $w_{\max}=0.9$ 、 $w_{\min}=0.2$ 和 $c_1=c_2=2$, 其余的参数设置则参照上述相关文献, 试验结果见表 3。

表 3 G-GWO 与其他优化算法对比结果

Table 3 Comparison of results between G-GWO and other optimization algorithms

函数	Golden-SA		WOA	
	平均值	标准差	平均值	标准差
f_1	$8.052\,7\text{e}-227$	0.000 0e-00	$3.303\,5\text{e}-72$	$1.717\,1\text{e}-71$
f_2	$1.222\,0\text{e}+128$	$6.693\,4\text{e}-127$	$3.767\,1\text{e}-50$	$1.536\,6\text{e}-49$
f_3	-1.256 9e+04	4.567 2e+01	$-1.077\,7\text{e}+04$	$1.516\,5\text{e}+03$
f_4	0.000 0e+00	0.000 0e+00	0.000 0e+00	0.000 0e+00
f_5	8.881 8e-16	0.000 0e+00	$3.493\,5\text{e}-15$	$2.272\,6\text{e}-15$
f_6	$4.576\,1\text{e}-04$	$3.349\,2\text{e}-04$	$6.908\,6\text{e}-04$	$3.958\,9\text{e}-04$
f_7	$3.981\,8\text{e}-01$	$2.634\,8\text{e}-03$	$3.979\,0\text{e}-01$	$2.144\,4\text{e}-05$
f_8	$8.597\,1\text{e}+00$	$1.117\,7\text{e}+01$	3.000 0e+00	$7.372\,8\text{e}-05$
f_9	$-3.817\,4\text{e}+00$	$2.481\,8\text{e}-03$	$-3.856\,6\text{e}+00$	$7.693\,0\text{e}-03$
函数	PSO		G-GWO	
	平均值	标准差	平均值	标准差
f_1	$1.748\,1\text{e}-04$	$2.551\,7\text{e}-04$	0.000 0e+00	0.000 0e+00
f_2	$2.869\,4\text{e}-02$	$3.306\,3\text{e}-02$	1.948 9e-217	0.000 0e+00
f_3	$-4.852\,9\text{e}+03$	$1.291\,4\text{e}+03$	$-1.256\,7\text{e}+04$	$5.123\,1\text{e}+00$
f_4	$5.679\,1\text{e}+01$	$1.806\,9\text{e}+01$	0.000 0e+00	0.000 0e+00
f_5	$1.004\,1\text{e}-01$	$2.541\,3\text{e}-01$	8.881 8e-16	0.000 0e+00
f_6	$8.356\,6\text{e}-04$	$1.688\,8\text{e}-04$	3.335 8e-04	2.406 3e-05
f_7	$3.978\,9\text{e}-01$	0.000 0e+00	3.980 4e-01	$1.494\,7\text{e}-04$
f_8	3.000 0e+00	1.531 7e-15	3.000 0e+00	$7.483\,6\text{e}-06$
f_9	$-3.862\,8\text{e}+00$	2.654 3e-15	-3.860 3e+00	$1.892\,9\text{e}-03$

由表 3 可知, 对于单峰函数 f_1 和 f_2 , G-GWO 算法的平均值分别为 0 和 $1.948\,9\text{e}-217$, 标准差值均为 0, 明显小于 Golden-SA、WOA 和 PSO 算法的平均值和标准差值, G-GWO 算法在单峰函数的收敛精度和鲁棒性较优。对于多峰函数 f_3 、 f_4 和 f_5 , G-GWO 算法的平均值分别为 $-1.256\,7\text{e}+04$ 、0 和 $8.881\,8\text{e}-16$, 且在 f_4 和 f_5 的标准差值为 0, 与 Golden-SA 算法相比, 两者在 f_3 的平均值结果相近, 在 f_4 和 f_5 中测试结果一致; 与 WOA 算法相比, 两者在 f_4 均收敛到理论最优值 0, 但在 f_3 和 f_5 中, G-GWO 算法的平均值更接近理论最优值, 具有更高的收敛精度, 同时, G-GWO 算法的标准差值在 f_3 和 f_5 均小于 WOA 算法, 具有较优的鲁棒性; 与 PSO 算法相比, G-GWO 算法的平均值和标准差值明显较优, 其收敛精度和鲁棒性均有显著优势。对于固定维度多峰函数 f_6 、 f_7 、 f_8 和 f_9 , G-GWO 算法的平均值分别为 $3.335\,8\text{e}-04$ 、 $3.980\,4\text{e}-01$ 、 $3.000\,0\text{e}+00$ 、 $-3.860\,3\text{e}+00$, 均收敛到理论最优值附近(f_6 的最优平均值为 $0.000\,30$),

f_7 的最优平均值为 0.398, f_8 的最优平均值为 3, f_9 的最优平均值为 -3.86), 较 Golden-SA 算法均有明显优势; 与 WOA 和 PSO 算法相比, G-GWO 算法在 f_6 、 f_7 和 f_9 收敛精度更高, 在 f_8 中则均收敛到理论最优值。

3.4 与其他改进 GWO 算法对比

选取具有变权重的改进灰狼优化算法(variable weights grey wolf optimization algorithm, VM-GWO)^[9]、自适应正态云模型的灰狼优化算法(adaptive normal cloud model grey wolf optimization algorithm, CGWO)^[21]随机对立学习的灰狼优化算法(random opposition-based learning grey wolf optimization algorithm, ROLGWO)^[22]三种改进 GWO 算法进行对比, 其中设置 CGWO 算法的参数为 $\omega=0.3$ 、 $\tau=1$ 和 $\xi=2$, 其余的参数设置则参照上述相关文献, 试验结果见表 4。

表 4 G-GWO 与其他改进 GWO 算法对比结果

Table 4 Comparison of results between G-GWO and other improved GWO algorithms

函数	VM-GWO		CGWO	
	平均值	标准差	平均值	标准差
f_1	1.459 7e-36	1.990 1e-36	1.938 6e-27	2.442 4e-27
f_2	3.557 9e-22	3.644 4e-22	1.078 2e-16	9.687 3e-17
f_3	-4.559 3e+03	3.485 6e+02	-5.905 4e+03	8.762 9e+02
f_4	3.160 2e+00	4.515 4e+00	2.791 2e+00	3.285 9e+00
f_5	3.452 1e-14	4.038 2e-15	9.894 3e-14	2.093 7e-14
f_6	7.158 0e-03	7.488 7e-03	2.923 2e-03	7.150 3e-05
f_7	3.981 3e-01	3.046 8e-04	3.978 9e-01	2.233 2e-06
f_8	3.000 0e+00	6.981 2e-05	3.000 0e+00	1.905 9e-05
f_9	-3.861 0e+00	2.758 3e-03	-3.862 6e+00	3.288 6e-04

函数	ROLGWO		G-GWO	
	平均值	标准差	平均值	标准差
f_1	7.245 1e-225	0.000 0e+00	0.000 0e+00	0.000 0e+00
f_2	1.674 2e-119	7.067 3e-118	1.948 9e-217	0.000 0e+00
f_3	-4.417 1e+03	1.254 9e+03	-1.256 7e+04	5.123 1e+00
f_4	0.000 0e+00	0.000 0e+00	0.000 0e+00	0.000 0e+00
f_5	8.881 8e-16	0.000 0e+00	8.881 8e-16	0.000 0e+00
f_6	6.337 3e-04	8.775 8e-04	3.335 8e-04	2.406 3e-05
f_7	3.979 4e-01	6.365 4e-05	3.980 4e-01	1.494 7e-04
f_8	3.000 0e+00	5.440 8e-05	3.000 0e+00	7.483 6e-06
f_9	-3.860 6e+00	2.951 4e-03	-3.860 3e+00	1.892 9e-03

由表 4 可知, 对于单峰函数 f_1 和 f_2 , 与 VM-GWO、CGWO 和 ROLGWO 算法相比, G-GWO 算法的平均值更接近理论最优值 0, 标准差值更小, G-GWO 算法的收敛精度和鲁棒性均有明显的优势。对于多峰函数 f_3 、 f_4 和 f_5 , 与 VM-GWO 和 CGWO 算法相比, G-GWO 算法的平均值更接近于理论最优值, 标准差更小; 与 ROLGWO 算法相比, G-GWO 算法在 f_3 的平均值和标准差均比 ROLGWO 算法接近于理论最优值, 而在 f_4 和 f_5 中的测试结果相同。对于固定维度多峰函数 f_6 、 f_7 、 f_8 和 f_9 , 与 VM-GWO 算法和 ROLGWO 算法相比, 三者仅在 f_8 中得到相同的平均值, 其余则 G-GWO 算法的平均值和标准差更优; 与 CGWO 算法相比, G-GWO 算法在 f_6 、 f_7 和 f_9 中的平均值明显更接近理论最优值, 在 f_8 中两者均收敛到理论最优值。

3.5 Wilcoxon 秩和检验分析

为了评估 G-GWO 算法的性能, 将 G-GWO 算法与其他 7 种算法在三类测试函数的运行结果进行 Wilcoxon 秩和检验, 并计算 p 值, 其中选择显著性水平 0.05, 使用 $+/\approx/-$ 来表明算法优于、相似或者差于比较的算法, N/A 表示算法结果接近, 无法进行显著性判断。测试函数的 Wilcoxon 秩和检验结果见表 5。

表 5 测试函数的 Wilcoxon 秩和检验结果

Table 5 Results of Wilcoxon rank sum test of test functions

函数	GWO		Golden-SA		WOA		PSO	
	p		p		p		p	
f_1	1.211 8e-12	+	5.772 0e-11	+	1.211 8e-12	+	1.211 8e-12	+
f_2	3.019 9e-11	+	3.019 9e-11	+	3.019 9e-11	+	3.019 9e-11	+
f_3	3.019 9e-11	+	4.856 0e-03	-	5.494 1e-11	+	3.019 9e-11	+
f_4	1.295 9e-11	+	N/A	≈	N/A	≈	1.211 8e-12	+
f_5	1.151 8e-12	+	N/A	≈	2.346 7e-07	+	1.211 8e-12	+
f_6	9.068 8e-03	+	5.873 7e-04	+	1.202 3e-08	+	3.019 9e-11	+
f_7	3.689 7e-11	+	3.402 9e-01	≈	3.019 9e-11	-	1.211 8e-12	+
f_8	1.174 7e-04	+	3.019 9e-11	+	4.553 0e-01	≈	2.655 4e-11	-
f_9	7.043 0e 07	+	9.513 6e 06	+	1.501 4e 02	+	7.574 1e 12	+
(+ / ≈ / -)	9/0/0		5/3/1		6/2/1		8/0/1	

函数	VM-GWO		CGWO		ROLGWO	
	p		p		p	
f_1	1.211 8e-12	+	1.211 8e-12	+	1.211 8e-12	+
f_2	3.019 9e-11	+	3.019 9e-11	+	3.019 9e-11	+
f_3	3.019 9e-11	+	3.019 9e-11	+	3.019 9e-11	+
f_4	3.641 6e-11	+	1.205 9e-12	+	N/A	≈
f_5	7.869 8e-13	+	1.119 2e-12	+	N/A	≈
f_6	1.729 4e-07	+	2.510 1e-02	+	4.744 5e-06	+
f_7	1.119 9e-01	≈	3.019 9e-11	+	2.871 6e-10	+
f_8	2.530 6e-04	+	3.501 2e-03	+	5.792 9e-02	≈
f_9	6.145 2e-02	≈	7.389 1e-11	+	9.883 4e-03	+
(+ / ≈ / -)	7/2/0		9/0/0		6/3/0	

由表 5 可知,G-GWO 算法的 Wilcoxon 秩和检验结果 p 值大部分小于 0.05,统计结果表明:相比其他 7 种常用优化算法,G-GWO 算法对三类测试函数的寻优性能具有显著优势。

3.6 收敛性分析

为了直观地反映 G-GWO 算法的收敛性能,选取 GWO、Golden-SA、WOA、PSO、VM-GWO、CGWO、ROLGWO 和 G-GWO 等 8 种算法对单峰函数 f_1 、 f_2 ,多峰函数 f_4 、 f_5 和固定维度多峰函数 f_6 、 f_7 作为典型函数进行收敛性分析,得到适应度均值随迭代次数的变化曲线,各算法寻优曲线对比如图 2 所示。

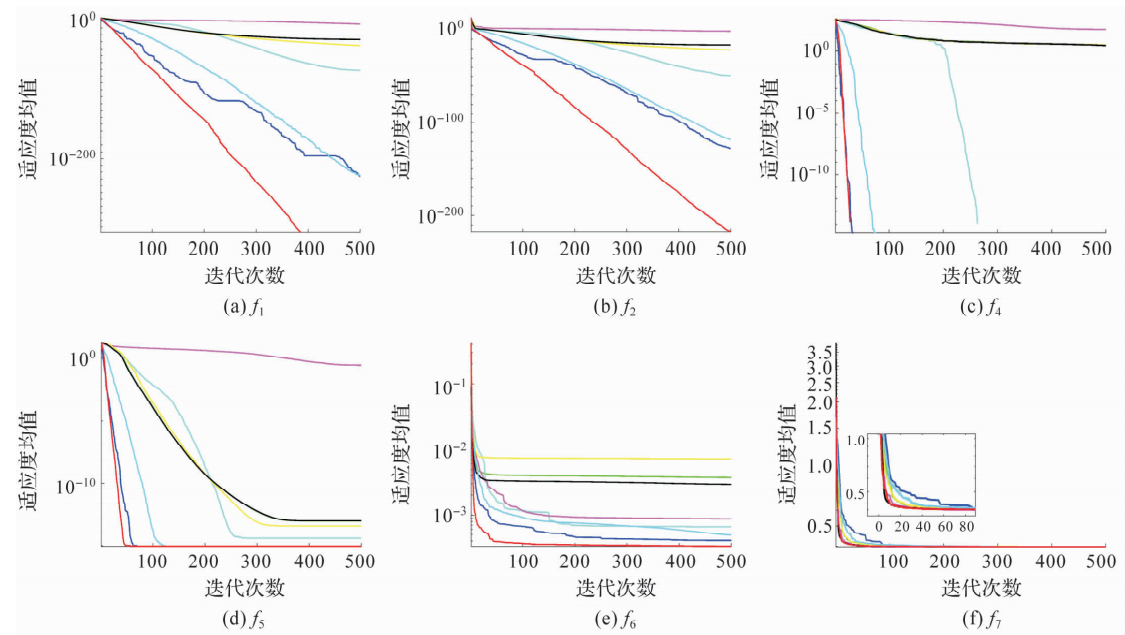


图 2 各算法寻优曲线对比

Fig. 2 Comparison of optimization curves for each algorithm

由图 2 可知,G-GWO 算法在单峰函数 f_1 和多峰函数 f_4 中能够快速地收敛到理论最优值,且不存在明显的停滞,并在单峰函数 f_2 、多峰函数 f_5 及固定维度多峰函数 f_6 和 f_7 中快速收敛至较优值并保持稳定,与其他算法相比具有明显优势。GWO、PSO、VM-GWO 和 CGWO 算法在三类基准测试函数中存在不同程度的停滞,收敛精度低;Golden-SA、WOA 和 ROLGWO 算法除了在多峰函数 f_4 外,其他基准测试函数中均陷入局部最优值。综合图 2 结果分析,G-GWO 算法在收敛速度和收敛精度上较其他算法具有较大的优势。

4 G-GWO 算法对板料冲压成形的代理模型精度优化研究

为验证 G-GWO 算法在实际应用中的有效性,对建立的板料冲压成形的 BP 神经网络代理模型精度进行优化,并将 G-GWO 算法优化结果与 GWO、Golden-SA、WOA、PSO、VM-GWO、CGWO 和 ROLGWO 算法的结果进行对比。

4.1 冲压件简介

本文选取异形薄壁不锈钢隔热罩为研究对象,其几何模型见图 3,三维尺寸为 378 mm×227 mm×76 mm,冲压件的厚度为 0.25 mm。

板坯材料所使用的是 SUS304 不锈钢,其材料性能参数如下:屈服强度为 239 MPa,抗拉强度为 520 MPa,弹性模量为 2.1×10^5 MPa,泊松比为 0.28,硬化指数为 0.40。



图 3 异形隔热罩几何模型

Fig. 3 Geometric model of special-shaped heat shield

4.2 成形质量与目标

将冲压件的最大减薄率和最大增厚率两个质量参数作为优化目标,优化目标函数如下:

$$Y_1 = \min y_1; \tag{15}$$

$$Y_2 = \min y_2. \tag{16}$$

式(15)~(16)中: Y_1 和 Y_2 为目标函数; y_1 为冲压件最大减薄率; y_2 为最大增厚率。

在影响冲压成形质量的主要工艺参数中,压边力(F)、冲压速度(v)、模具间隙(t)和摩擦系数(u)的取值范围见表 6。

表 6 参数的取值范围

Table 6 Range of parameter values

参数	取值范围
F/kN	90~170
$v/(\text{mm} \cdot \text{s}^{-1})$	2 000~5 000
t/mm	0.09~0.15
u	0.25~0.31

4.3 试验设计

使用拉丁超立方抽样法(latin hypercube sampling,LHS)抽取样本,在压边力、冲压速度、模具间隙和摩擦系数 4 个优化参数的取值范围内抽取满足数值拟合要求的 90 组试验参数,使用有限元软件 Dynaform 进行仿真试验,得到各组在抽样点条件下的冲压件最大减薄率和最大增厚率,并通过 BP 神经网络采用单输出的拟合方式^[23],即对隔热罩异形件的质量参数最大减薄率和最大增厚率为输出分别建模。将 90 个样本中的 80%作为网络训练数据,余下 20%作为网络预测数据,截取前两组和后三组的抽样仿真数据的结果,见表 7。

表 7 抽样仿真数据的结果

Table 7 Results of sample simulation data

编号	工艺参数				$y_1/\%$	$y_2/\%$
	F/kN	$v/(\text{mm} \cdot \text{s}^{-1})$	t/mm	u		
1	154.39	2 367	0.29	0.11	30.62	8.00
2	147.60	3 279	0.26	0.12	30.96	6.06
88	109.12	3 890	0.27	0.13	29.55	8.74
89	134.89	3 487	0.27	0.12	30.35	7.74
90	108.57	2 066	0.26	0.10	28.58	9.80

4.4 精度分析

分别使用 G-GWO 和其他 7 种算法优化 BP 神经网络的权值和阈值,建立 4 个工艺参数与最大减薄率和最大增厚率的代理模型。通过均方误差和决定系数来检验 BP 神经网络模型的精度。其中,最大减薄率和最大增厚率的预测相对误差比较如图 4 所示,均方误差(mean square error,MSE)和决定系数(coefficient of determination, R^2)比较结果见表 8。

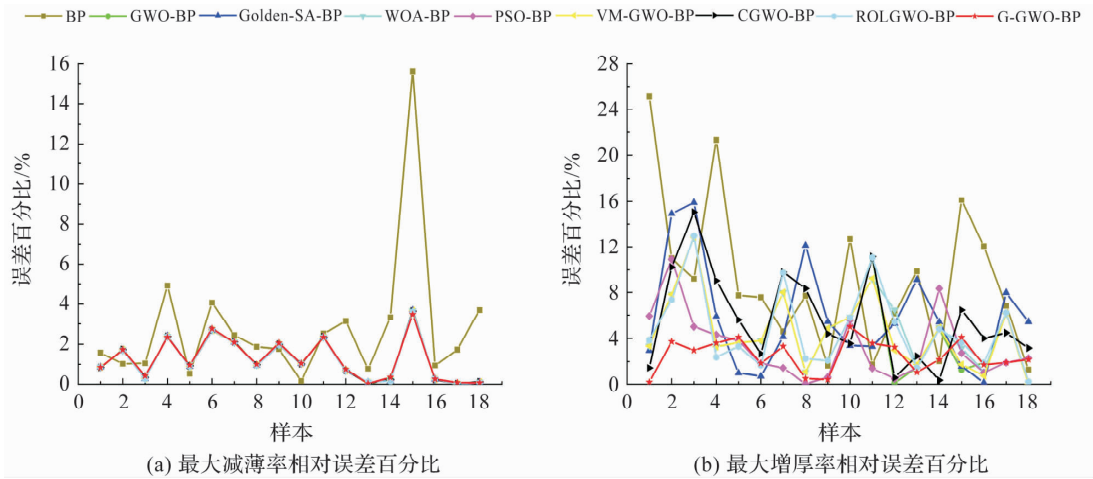


图 4 预测相对误差比较

Fig. 4 Comparison of relative errors of predictions

表 8 均方误差和决定系数比较结果

Table 8 Comparison results of mean square error and coefficient of determination

算法	MSE		R^2	
	y_1	y_2	y_1	y_2
BP	1.65e-04	7.89e-05	0.673 1	0.872 8
GWO-BP	2.75e-05	1.27e-05	0.945 7	0.979 5
Golden-SA-BP	2.74e-05	3.06e-05	0.945 5	0.950 6
WOA-BP	2.74e-05	1.67e-05	0.945 8	0.973 0
PSO-BP	2.73e-05	7.96e-06	0.945 9	0.987 2
VM-GWO-BP	2.74e-05	1.43e-05	0.945 7	0.969 1
CGWO-BP	2.75e-05	2.58e-05	0.945 6	0.958 0
ROLGWO-BP	2.72e-05	1.53e-05	0.945 8	0.975 4
G-GWO-BP	2.72e-05	4.73e-06	0.946 3	0.992 4

由图 4(a)可知,G-GWO 算法优化的代理模型精度较高,最大相对误差为 3.47%;由图 4(b)可知,G-GWO 算法优化的代理模型的最大相对误差为 4.99%,明显优于其他算法。综合图 4 分析,G-GWO 算法优化的代理模型的相对误差值均小于 5%,训练的代理模型精度高。由表 8 可知,G-GWO 算法优化的最大减薄率代理模型的 MSE 和 R^2 分别为 $2.72e-05$ 和 0.946 3,最大增厚率代理模型的 MSE 和 R^2 分别为 $4.73e-06$ 和 0.992 4,G-GWO 优化 BP 神经网络的代理模型精度明显优于其他算法,预测精度高,建立的高精度 G-GWO 算法优化 BP 神经网络代理模型可以满足后续对冲压工艺成形优化设计的要求。综合以上结果,G-GWO 算法在板料冲压成形的代理模型精度优化问题上具有一定的优越性。

5 结 论

本研究在 GWO 算法的基础上,以提高 GWO 算法的收敛精度、收敛速度和跳出局部最优为目的,提出了一种融合多策略的黄金正弦灰狼优化算法(G-GWO),得出如下结论。

1) 引入非线性收敛因子和动态比例权重策略,协调算法全局探索和局部开发的能力,提高算法的收敛速度;结合黄金正弦策略更新灰狼位置,提高算法在搜索后期处理局部最优的能力,采用多种策略融合

可以使算法寻优性能更加良好。通过三类基准测试函数进行寻优测试,从平均值、标准差及收敛曲线的结果表明,G-GWO算法具有较优的收敛精度和鲁棒性,以及快速收敛的能力;并进行 Wilcoxon 秩和检验,从统计学的角度验证了 G-GWO 算法的显著优势。

2) 将 G-GWO 算法与其他 7 种优化算法应用于建立板料冲压成形的 BP 神经网络代理模型中,其中 G-GWO 算法建立的最大减薄率和最大增厚率代理模型误差均小于 5%,具有显著优势,验证了 G-GWO 算法在实际优化问题中的有效性。

未来,通过将不同的灰狼位置更新模型以及结合各种优秀的算法来增强 GWO 算法的搜索能力,同时也将致力研究应用于更多的领域。

参考文献:

- [1] MIRJALILI S, MIRJALILI S M, LEWIS A. Grey wolf optimizer[J]. *Advances in Engineering Software*, 2014, 69: 46.
- [2] MAKHADMEH S N, KHADER A T, AL-BETAR M A, et al. A novel hybrid grey wolf optimizer with min-conflict algorithm for power scheduling problem in a smart home[J]. *Swarm and Evolutionary Computation*, 2021, 60: 100793.
- [3] JIANG T H, ZHANG C, ZHU H Q, et al. Energy-efficient scheduling for a job shop using grey wolf optimization algorithm with double-searching mode[J]. *Mathematical Problems in Engineering*, 2018, 2018: 1.
- [4] LIU Y Y, SUN J H, YU H Y, et al. An improved grey wolf optimizer based on differential evolution and OTSU algorithm[J]. *Applied Sciences*, 2020, 10(18): 6343.
- [5] KUMAR R, SINGH L, TIWARI R. Path planning for the autonomous robots using modified grey wolf optimization approach[J]. *Journal of Intelligent & Fuzzy Systems*, 2021, 40(5): 9453.
- [6] PURUSHOTHAMAN R, RAJAGOPALAN S P, DHANDAPANI G. Hybridizing gray wolf optimization (GWO) with grasshopper optimization algorithm (GOA) for text feature selection and clustering[J]. *Applied Soft Computing*, 2020, 96: 106651.
- [7] YAN F, XU X L, XU J Z. Grey wolf optimizer with a novel weighted distance for global optimization[J]. *IEEE Access*, 2020, 8: 120173.
- [8] LIU X, WANG N. A novel gray wolf optimizer with RNA crossover operation for tackling the non-parametric modeling problem of FCC process[J]. *Knowledge-Based Systems*, 2021, 216: 106751.
- [9] GAO Z M, ZHAO J. An improved grey wolf optimization algorithm with variable weights[J]. *Computational Intelligence and Neuroscience*, 2019, 2019: 2981282.
- [10] 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.
- [11] XU J Z, YAN F, ALA O G, et al. Chaotic dynamic weight grey wolf optimizer for numerical function optimization[J]. *Journal of Intelligent & Fuzzy Systems*, 2019, 37(2): 2367.
- [12] 韩太林, 张延雪, 王啸, 等. 改进型灰狼算法在热电偶动态补偿中的应用[J]. *控制与决策*, 2021, 36(1): 61.
- [13] MENG A B, ZENG C, WANG P, et al. A high-performance crisscross search based grey wolf optimizer for solving optimal power flow problem[J]. *Energy*, 2021, 225: 120211.
- [14] LONG W, WU T B, CAI S H, et al. A novel grey wolf optimizer algorithm with refraction learning[J]. *IEEE Access*, 2019, 7: 57805.
- [15] YU X W, HUANG L P, LIU Y, et al. WSN node location based on beetle antennae search to improve the grey wolf algorithm[J]. *Wireless Networks*, 2022, 28(2): 539.
- [16] MOHAMMED H, RASHID T. A novel hybrid GWO with WOA for global numerical optimization and solving pressure vessel design[J]. *Neural Computing and Applications*, 2020, 32(18): 14701.
- [17] 李全耀, 沈艳霞. 一种基于教与学的混合灰狼优化算法[J]. *控制与决策*, 2022, 37(12): 3190.
- [18] TANYILDIZI E, DEMIR G. Golden sine algorithm; a novel math-inspired algorithm[J]. *Advances in Electrical and Computer Engineering*, 2017, 17(2): 71.

- [19] MIRJALILI S, LEWIS A. The whale optimization algorithm[J]. Advances in Engineering Software, 2016, 95: 51.
- [20] EBERHART R, KENNEDY J. A new optimizer using particle swarm theory[C]//Proceedings of the Sixth International Symposium on Micro Machine and Human Science. Nagoya, Japan: IEEE, 1995: 39.
- [21] 张铸, 饶盛华, 张仕杰. 基于自适应正态云模型的灰狼优化算法[J]. 控制与决策, 2021, 36(10): 2562.
- [22] LONG W, JIAO J J, LIANG X M, et al. A random opposition-based learning grey wolf optimizer[J]. IEEE Access, 2019, 7: 113810.
- [23] 肖展开, 梅益, 罗宁康, 等. 基于神经网络航空发动机曲轴加工工艺设计优化[J]. 锻压技术, 2022, 47(6): 35.

~~~~~  
(上接第 503 页)

- [12] LUO X, FENG H, LIU J, et al. An experimental investigation on thermal contact resistance across metal contact interfaces[C]//12th International Conference on Electronic Packaging Technology and High Density Packaging. Piscataway: IEEE, 2011: 733.
- [13] 毛贺, 陈章位, 黄靖, 等. PCR 仪温度场热模型的研究与验证[J]. 浙江大学学报(工学版), 2013, 47(9): 1619.
- [14] 黄靖, 陈章位, 姚英豪, 等. 定量 PCR 仪热循环系统温度均匀性有限元仿真研究[J]. 仪器仪表学报, 2010, 31(5): 1142.
- [15] 陈世泽, 陈章位, 黄靖, 等. PCR 仪温度控制均匀性研究与仿真分析[J]. 中国生物医学工程学报, 2009, 28(4): 610.
- [16] 朱杰, 高贯虹. 96 孔生化样品处理台的温度均匀设计与优化[J]. 制造业自动化, 2014, 36(9): 145.
- [17] 刘颖, 黄嗣竣, 刘义平, 等. NTC 热敏电阻温度标定方法研究[J]. 中国测试, 2022, 48(增刊 1): 25.
- [18] 王崇杰. 酚醛树脂复合材料导热及力学性能研究[D]. 北京: 北京化工大学, 2016.
- [19] 占成元, 李宁, 文玉华, 等. 基于 ANSYS 模拟的隔热系统设计[J]. 机械, 2009, 36(3): 34.
- [20] 国家药品监督管理局. 全自动发光免疫分析仪: YY/T 1155—2019[S]. 北京: 中国标准出版社, 2019.