

基于 Tent 混沌序列的灰狼优化算法

张贾奎, 崔利杰, 郭 庆, 陈浩然

(空军工程大学 装备管理与安全工程学院, 陕西 西安 710051)

摘 要: 针对灰狼优化(GWO)算法在优化求解过程中存在的局部搜索能力差、易陷入局部最优等问题, 提出了基于 Tent 混沌序列的局部搜索策略以及多样性维持策略. 由此构建了基于 Tent 混沌序列的灰狼优化算法(TCGWO). 所提出的策略分别用以提高算法的局部搜索能力、避免算法陷入局部最优. 通过对 23 个测试函数的仿真计算, 并与 GWO 以及 FPA、ABC 等进行综合对比, 结果验证了 TCGWO 算法的有效性及其优越性.

关键词: 灰狼优化算法; Tent 混沌序列; 局部搜索策略; 多样性维持策略

中图分类号: TP18

文献标识码: A

文章编号: 1000-7180(2018)06-0011-06

Grey Wolf Optimizer Based on Tent Chaotic Sequence

ZHANG Jia-kui, CUI Li-jie, GUO Qing, CHEN Hao-ran

(Material Management and Safety Engineering College, Air Force Engineering
University, Xi'an 710051, China)

Abstract: Focusing on the disadvantages that Grey Wolf Optimizer (GWO) was easy to fall into local optimum and had bad local searching ability on optimization, local search strategy and diversity maintenance strategy were proposed. On the basis of that, the Grey Wolf Optimizer Based on Tent Chaotic Sequence (TCGWO) was established. The local searching ability could be improved by local search strategy; falling into local optimum could be avoided by diversity maintenance strategy. 23 benchmark test functions were used to experiment and analysis, and comparative analysis with GWO, Flower Pollination Algorithm (FPA) and Artificial Bee Colony Algorithm (ABC). According to the result, TCGWO is effective and superior.

Key words: grey wolf optimizer; tent chaotic sequence; local search strategy; diversity maintenance strateg

1 引言

群智能算法寻优的有效性,使群智能算法受到了广泛的关注,成为寻优算法设计和优化分析的热点之一,并广泛应用于经济和社会等各个方面.

2014 年, Mirjalili 等^[1]人通过对灰狼的捕食行为的模拟,提出了一种新型的群智能算法——灰狼优化算法(Grey Wolf Optimizer, GWO)算法. 由于 GWO 算法原理简单、参数设置少、寻优效果好等特点,自提出以来, GWO 算法就受到广泛的关注,并在无人机三维航路规划^[2]、UCAV 协同攻击决策^[2]、图像分割^[3]等问题上得到了应用. 虽然 GWO

算法已被证明具有良好的寻优性能,但该算法还存在初始化不均匀、局部搜索能力差和易陷入局部最优等缺陷. 为了提高算法的性能, Zhu 等^[4], 基于 DE 算法良好的全局搜索能力提出了一种混合 GWO 算法; Muangkote 等^[5], 提出了一种通过系数调整来调整算法能局部搜索能力的改进 GWO; 魏政磊等^[6], 提出了控制参数值非线性调整策略来解决算法陷入局部最优的问题. 本文通过对 α 、 β 、 δ 个体执行基于 Tent 混沌序列的局部搜索策略, 以提高算法的局部搜索能力; 通过对种群执行多样性维持策略, 以避免算法陷入局部最优, 进而构建了基于 Tent 混沌序列的灰狼算法(TCGWO), 以改善算法

的寻优性能.

2 基本灰狼优化算法

2.1 GWO 的生物学原理

灰狼是犬科群居动物,一般群体中由 5~12 头狼组成.灰狼群在捕猎的过程中有着严格的社会等级,从顶端至底端依次是 α 狼、 β 狼、 δ 狼和 ω 狼 ω 狼,如图 1 所示.

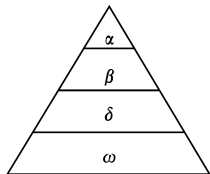


图 1 灰狼群体的社会等级

严格的等级制度使灰狼群在狩猎时的效率大大提高.通常狩猎的过程是由 α 狼带领完成的.其狩猎行为主要包括搜索追踪、包围袭击和进攻三个阶段.

2.2 GWO 的数学描述

在 GWO 算法中将当前的最优个体记为 α 、将次优个体记为 β 、将第三优个体记为 δ ,除却 α 、 β 、 δ 外,其余个体均记为 ω .针对 D 维寻优求解问题,设初始灰狼数量为 N ,则对第 i 只灰狼的包围行为描述如下.

$$\vec{D}_i = |\vec{C} \cdot \vec{X}_p(t) - \vec{X}_i(t)| \quad (1)$$

$$\vec{X}_i(t+1) = \vec{X}_p(t) - \vec{A} \cdot \vec{D}_i \quad (2)$$

式中, t 表示当前的迭代次数 ($t \leq T_{max}$); $\vec{D}_i$ 表示当前第 i 只灰狼与猎物之间的距离; $\vec{X}_p(t)$ 表示第 t 代时全局最优解的位置(猎物的位置); $\vec{X}_i(t)$ 表示第 t 代时的解向量(灰狼个体的位置); $\vec{A}$ 与 $\vec{C}$ 为向量系数,由如下公式计算得到.

$$\vec{A} = 2a \cdot \vec{r}_1 - a \quad (3)$$

$$\vec{C} = 2 \cdot \vec{r}_2 \quad (4)$$

式中, $\vec{r}_1$ 与 $\vec{r}_2$ 是 $[0,1]$ 区间上的随机向量; a 为随迭代次数的增加从 2 线性递减到 0.

经过包围阶段,第 i 只灰狼会通过 α 、 β 、 δ 对的位置更新位置.

$$\begin{cases} \vec{D}_\alpha = |\vec{C}_1 \cdot \vec{X}_\alpha(t) - \vec{X}_i(t)| \\ \vec{D}_\beta = |\vec{C}_2 \cdot \vec{X}_\beta(t) - \vec{X}_i(t)| \\ \vec{D}_\delta = |\vec{C}_3 \cdot \vec{X}_\delta(t) - \vec{X}_i(t)| \end{cases} \quad (5)$$

$$\begin{cases} \vec{X}_1(t+1) = \vec{X}_\alpha(t) - \vec{A}_1 \cdot \vec{D}_\alpha \\ \vec{X}_2(t+1) = \vec{X}_\beta(t) - \vec{A}_2 \cdot \vec{D}_\beta \\ \vec{X}_3(t+1) = \vec{X}_\delta(t) - \vec{A}_3 \cdot \vec{D}_\delta \end{cases} \quad (6)$$

$$\vec{X}_i(t+1) = \frac{\vec{X}_1(t+1) + \vec{X}_2(t+1) + \vec{X}_3(t+1)}{3} \quad (7)$$

3 基于 Tent 混沌序列的灰狼优化算法

混沌序列具有内随机性、遍历性、有界性等特点,但不同的映射产生的混沌序列遍历的均匀性不同,对算法的寻优速度会产生不同的影响.目前文献中多采用 Logistic 映射产生混沌序列,但是 Logistic 映射产生的混沌序列的均匀性较差,多在 $[0,0.1]$ 和 $[0.9,1]$ 两个区间取值.单梁等^[7]指出 Tent 映射产生的混沌序列在 $[0,1]$ 区间内分布比 Logistic 映射产生的混沌序列更加均匀.因此,本文选用 Tent 映射产生的混沌序列来改进 GWO 算法. Tent 映射其数学表达式为:

$$y_{n+1} = \begin{cases} 2y_n, & 0 \leq y_n \leq \frac{1}{2} \\ 2(1-y_n), & \frac{1}{2} < y_n \leq 1 \end{cases} \quad (8)$$

经伯努利位移变换之后的表达式为

$$y_{n+1} = (2y_n) \bmod 1 \quad (9)$$

3.1 局部搜索策略

GWO 在迭代后期存在局部搜索能力差的缺点.因此,本文提出了基于 Tent 混沌序列进行局部搜索的策略.该策略主要是对当前最优的三个解 α 、 β 、 δ ,进行局部的混沌搜索.以 α 的局部搜索为例,步骤如下:

Step1: 记 $\text{Pos} = (x_1, x_2, \dots, x_n)$ 为当前 α 的位置, fitness 为当前 α 的值,设置最大混沌搜索次数 M ,以及混沌搜索计数器 $m = 1$;

Step2: 利用 Tent 映射产生一个混沌序列,序列中的混沌变量记为 t_i ,其中 $i = 1, 2, \dots, n$, n 为变量个数;

Step3: 对于当前的 α 位置 $(x_1, x_2, \dots, x_n)$ 根据下式进行改变:

$$x_i^m = x_i + \rho \times f \times t_i \quad (10)$$

Step4: 计算 $(x_1^m, x_2^m, \dots, x_n^m)$ 的适应度值 fitness_m , 若 $\text{fitness}_m < \text{fitness}$ 则 $\text{fitness} = \text{fitness}_m$, $\text{Pos} = (x_1^m, x_2^m, \dots, x_n^m)$, 并令计数器 $m = m + 1$;

Step5:若 $m \leq M$,则跳转至 Step 2,若 $m > M$,则跳转至 Step6;

Step6:输出混沌搜索后 α 的位置 Pos,以及适应度值 fitness.

在式(10)中, f 为随机方向因子,取值范围为 $\{-1, 1\}$,这样可以使得搜索随机向正、负方向进行,保证混沌局部搜索的成功率; r 为搜索半径,其设置如下.

$$r = \frac{ub_i - lb_i}{10}$$

(11)

式中, ρ 为衰减因子,其值将会随着迭代次数的增加而线性衰减,即混沌搜索的范围随着迭代次数的增加不断减小,逐渐实现精细搜索.其计算公式为:

$$\rho = 1 - \frac{t-1}{T_{\max}}$$

(12)

式中, $T_{\max}$ 为算法最大迭代次数; t 为当前迭代次数.

3.2 多样性维持策略

混沌局部搜索策略保证了 GWO 的局部搜索能力,但算法后期仍然容易陷入局部最优,导致结果质量不佳.为了克服该问题,本文提出了一种多样性维持策略.首先,判断种群的多样性是否需要调整,判

断依据主要有两个:

判据一:任意个体位置 $(x_{i1}, x_{i2}, \dots, x_{in})$ 与最优个体 α 的位置 $(x_1, x_2, \dots, x_n)$ 之间的欧氏距离小于 r , r 为混沌局部搜索的半径.即:

$$\sqrt{\sum_{j=1}^n (x_j - x_{ij})^2} \leq r$$

(13)

判据二:算法全局最优位置连续不变的迭代次数超过 10.

当满足上述两个判据中的任意一个时,则对种群的多样性进行调整.调整策略为:保留最优的三个解 α, β, δ ;然后对种群中随机的 30% 的个体重新进行初始化.

4 数值仿真及分析

4.1 测试函数

为了验证 TCGWO 算法的性能,选取了 23 个基准测试函数进行数值仿真并与原始 GWO 和 FPA、ABC,进行综合比较.表 1 和表 2 分别给出了固定维度基准测试函数与高维基准测试函数的信息.

表 1 固定维度基准测试函数

函数	函数表达式	维度	解空间	全局极小值
F_1	$f(x) = \sum_{j=1}^n (\sum_{i=1}^n x_i - \beta_j)^2$	$D = 2$	$[0, 2]^2$	3.299 3
F_2	$f(x) = \sum_{j=1}^n \{ \sum_{i=1}^n (i + \beta) \cdot \lfloor (x_i / i)^j - 1 \rfloor \}$	$D = 2$	$[-2, 2]^2$	0
F_3	$f(x) = \lfloor 1 + (x_1 + x_2 + 1)^2 (19 - 14x_1 + 3x_1^2 - 14x_2 + 6x_1x_2 + 3x_2^2) \rfloor \times \lfloor 30 + (2x_1 - 3x_2)^2 \times (18 - 32x_1 + 12x_1^2 + 48x_2 - 36x_1x_2 + 27x_2^2) \rfloor$	$D = 2$	$[-2, 2]^2$	3
F_4	$f(x) = -0.5 - (\cos(\sin x_1^2 - x_2^2) - 0.5) / \lfloor 1 + 0.001(x_1^2 + x_2^2) \rfloor^2$	$D = 2$	$[-100, 100]^2$	-1
F_5	$f(x) = 0.5 + (\sin^2\sqrt{x_1^2 + x_2^2} - 0.5) / \lfloor 1 + 0.001(x_1^2 + x_2^2) \rfloor^2$	$D = 2$	$[-100, 100]^2$	0
F_6	$f(x) = -(x_2 + 47) \cdot \sin \sqrt{ x_2 + x_1/2 + 47 } - x_1 \cdot \sin \sqrt{ x_1 - x_2 - 47 }$	$D = 2$	$[-512, 512]^2$	-959.640 7
F_7	$f(x) = \frac{-1}{10\,000} \lfloor \sin(x_1) \cdot \sin(x_2) \cdot \exp(100 - \sqrt{x_1^2 + x_2^2}/\pi) + 1 \rfloor^{0.1}$	$D = 2$	$[-10, 10]^2$	-2.062 6
F_8	$f(x) = \{0.002 + \sum_{i=-2j}^2 \sum_{j=-2}^2 \lfloor 5(i+2) + j + 3 + (x_1 - 16j)^6 + (x_2 - 16i)^6 \rfloor^{-1} \}^{-1}$	$D = 2$	$[-65.53, 65.53]^2$	0.998
F_9	$f(x) = -1 + \cos(12\sqrt{x_1^2 + x_2^2}) / (0.5(x_1^2 + x_2^2) + 2)$	$D = 2$	$[-5.12, 5.12]^2$	-1
F_{10}	$f(x) = -\sin \sqrt{x_1^2 + x_2^2} / \sqrt{x_1^2 + x_2^2} - \exp \lfloor (\cos 2\pi x_1 + \cos 2\pi x_2) / 2 \rfloor + 0.712\,9$	$D = 2$	$[-10, 10]^2$	-3.005 4
F_{11}	$f(x) = (1/500 + \sum_{j=1}^{25} 1/\lfloor j + \sum_{i=1}^j (x_i + a_{ij})^6 \rfloor)^{-1}$	$D = 2$	$[-65, 65]^2$	1
F_{12}	$f(x) = 4x_1^2 - 2.1x_1^4 + \frac{1}{3}x_1^6 + x_1x_2 - 4x_2^2 + 4x_2^4$	$D = 2$	$[-5, 5]^2$	-1.031 6
F_{13}	$f(x) = \sum_{j=1}^{11} \lfloor a_i - x_1(b_i^2 + b_ix_2) / b_i^2 + b_ix_3 + x_4 \rfloor^2$	$D = 4$	$[-5, 5]^4$	0.000 3
F_{14}	$f(x) = -\sum_{i=1}^4 c_i \exp(-\sum_{j=1}^3 a_{ij} (x_j - p_{ij})^2)$	$D = 3$	$[1, 3]^3$	-3.86
F_{15}	$f(x) = -\sum_{i=1}^4 c_i \exp(-\sum_{j=1}^6 a_{ij} (x_j - p_{ij})^2)$	$D = 6$	$[0, 1]^6$	-3.32

表 2 高维基准测试函数

函数	函数表达式测试函数	维数	解空间	全局极小值
F_{16}	$f(x)=\sum_{i=1}^n(x_i+0.5)^2$	$D=30$	$[-100,100]^{30}$	0
F_{17}	$f(x)=\sum_{i=1}^n[(x_0-x_i)^2+(1-x_i)^2]$	$D=30$	$[-1,1]^{30}$	0
F_{18}	$f(x)=\sum_{i=1}^n x_i \cdot\prod_{i=1}^n x_i $	$D=30$	$[-10,10]^{30}$	0
F_{19}	$f(x)=\frac{\pi}{n}\{10\sin(\pi y_1)+\sum_{i=1}^{n-1}(y_i-1)^2[1+100\sin^2(\pi y_i)]+(y_n-1)^2\}+\sum_{i=1}^nu(x_i,10,100,4)$	$D=30$	$[-50,50]^{30}$	0
F_{20}	$f(x)=0.1\{\sin^2(3\pi x_1)+\sum_{i=1}^n(x_i-1)^2[1+\sin^2(3\pi x_1+1)]+(x_n-1)2[1+\sin^2(2\pi x_n)]+\sum_{i=1}^nu(x_i,5,100,4)\}$	$D=30$	$[-50,50]^{30}$	0
F_{21}	$f(x)=\sin^2(\pi z_1)+\sum_{i=1}^{n-1}(z_i-1)^2\cdot[1+10\sin^2(\pi z_i+1)]+(z_n-1)^2\cdot[1+\sin^2(2\pi z_n)]$ $z_i=1+(x_i-1)/4\quad i=1,2,\cdots,n$	$D=30$	$[-10,10]^{30}$	0
F_{22}	$f(x)=\{\exp[-\sum_{i=1}^n(x_i/\beta)^{2m}]-2\exp(-\sum_{i=1}^nx_i^2)\}\cdot\prod_{i=1}^n\cos^2(x_i),\beta=15,m=5$	$D=30$	$[-20,20]^{30}$	-1
F_{23}	$f(x)=\sum_{i=1}^{n/4}[(x_{4i-3}+10x_{4i-2})^2+5(x_{4i-1}-x_{4i})^2+(x_{4i-2}-2x_{4i-1})^4+10(x_{4i-3}-x_{4i})^4]$	$D=30$	$[-4,5]^{30}$	0

式中, F_{18} 和 F_{19} 中的 y_i 和 $u(x_i,a,k,m)$ 的计算公式如下.

$$y_i=1+\frac{x_i+1}{4}$$
$$u(x_i,a,k,m)=\begin{cases}k(x_i-a)^m,&x_i>a\\0,&-a<x_i<a\\k(-x_i-a)^m,&x_i<-a\end{cases}$$

4.2 数值仿真及结果分析

为了比较结果更加公平合理,仿真实验统一在平台 Matlab 2012a 上进行. 分别用 TCGWO 算法、原始 GWO 算法、FPA 算法、ABC 算法对 23 个基准测试函数进行寻优求解. 几种算法的参数设置见表

3. 采用表 3 中的参数设置,所有算法对每个函数进行 50 次独立的求解,以避免算法的随机性所带来的误差. 为了更好的比较算法的性能,本文用 50 次寻优结果的平均值、标准差等统计学指标对算法进行评价. 平均值主要反映算法的求解精度,标准差主要反映算法的寻优稳定性.

表 3 算法参数设置

算法	参数设置
公共参数	种群规模 $N=30$;最大迭代次数 $T_{\text{ma}}=500$;
FPA	转换概率 $p=0.8$;Lévy 系数 $\lambda=1/7$;
ABC	侦察蜂数量为 10;

表 4 高维基准测试函数的寻优结果对比数据

F	TCGWO		GWO		FPA		ABC	
	平均值	标准差	平均值	标准差	平均值	标准差	平均值	标准差
F_{16}	3.34E-04	8.68E-05	6.64E-01	3.48E-01	2.41E+03	5.25E+02	2.03E+03	1.43E+03
F_{17}	7.23E-04	2.19E-04	3.67E+00	1.82E+00	6.08E+01	1.41E+01	6.22E+01	1.94E+01
F_{18}	0	0	0	0	8.78E-07	1.74E-06	2.79E-17	2.61E-17
F_{19}	6.41E-05	2.36E-05	4.07E-02	1.94E-02	2.85E+01	8.91E+00	4.24E+02	1.55E+02
F_{20}	5.69E-03	9.02E-03	7.13E-01	2.17E-01	1.55E+05	7.93E+04	6.50E+05	1.23E+06
F_{21}	3.39E-01	6.89E-02	1.29E+00	2.78E-01	1.92E+01	3.72E+00	9.71E+00	5.87E+00
F_{22}	1.17E-165	0	3.17E-144	6.34E-144	2.25E-134	4.51E-134	5.38E-66	1.07E-65
F_{23}	9.95E-06	1.01E-05	3.23E-05	2.35E-05	2.30E+02	7.81E+01	9.82E+02	4.11E+02

高维基准测试函数中 $F_{16} \sim F_{18}$ 为单峰函数,它们主要测试算法在高维情况下的全局搜索能力,

$F_{19}-F_{34}$ 为具有更多局部最小值的多峰函数,它们主要测试算法在高维情况下的局部搜索能力与跳出局部最优的能力.从表 5 给出的 4 种算法对高维基准测试函数的寻优结果对比数据,TCGWO 与 GWO 虽然都能逼近函数的理论最优值,但 TCGWO 的平

均值优于 GWO 的平均值 0~21 个数量级,并且 TCGWO 的标准差比 GWO 至少小 0~144 个数量级.而且,TCGWO 的求解精度与求解稳定性明显优于 FPA 与 ABC.

表 5 固定维度基准测试函数的寻优结果对比数据

F	TCGWO		GWO		FPA		ABC	
	平均值	标准差	平均值	标准差	平均值	标准差	平均值	标准差
F_1	3.299 3	0	3.299 3	0	3.299 3	0	3.299 36	4.90E-05
F_2	6.21E-12	9.17E-12	7.83E-10	6.02E-10	7.63E-12	5.04E-12	2.67E-04	0.000 116
F_3	3	0	3.000 06	8.00E-05	3	0	3.012 1	1.35E-02
F_4	-1	0	-1	0	-1	0	-9.99E-01	1.59E-03
F_5	1.24E-03	2.48E-03	5.83E-03	4.76E-03	7.39E-03	3.72E-03	1.07E-02	6.96E-04
F_6	-959.640	1.20E-03	-912.002	6.76E+01	-956.682	1.77E+00	-943.244	6.19E+00
F_7	-2.062 6	0	-2.062 6	0	-2.062 6	0	-2.062 6	4.00E-05
F_8	0.998	0	4.324 94	4.237 120 5	0.998	0	1.032	6.02E-02
F_9	-1	0	-1	0	-0.999 992	1.17E-05	-0.962 55	2.35E-02
F_{10}	-3.005 4	0	-3.005 4	0	-3.005 4	0	-2.996 84	5.46E-03
F_{11}	0.998	0	3.93E+00	4.43E+00	0.998	3.99E-01	1.00E+00	4.26E-03
F_{12}	-1.031 6	0	-1.031 6	0	-1.031 6	0	-1.031 14	7.36E-04
F_{13}	6.79E-04	2.89E-04	4.40E-03	7.98E-03	5.76E-04	7.14E-05	1.92E-03	8.03E-04
F_{14}	-3.862 78	4.00E-05	-3.862 56	1.96E-04	-3.862 8	0	-3.862 8	0
F_{15}	-3.322	0	-3.322	0	-1.976 48	2.63E+00	-3.316 72	9.57E-03

固定维度基准测试函数中, $F_1\sim F_3$ 为单峰函数,它们的全局极小值处在比较平坦的区域,寻优难度较大,因此它们能有效的测试算法的全局搜索能力.从表 4 可以看出,对于 $F_1\sim F_3$,4 种算法的寻优结果都能接近理论值.但是在求解 F_2 时,TCGWO 的求解精度与求解稳定性明显优于原始 GWO,并且优于 FPA、ABC. $F_4\sim F_{15}$ 为多峰函数,大量的局部最优值使得算法很容易陷入局部最优,因此它们能很好的测试算法的局部搜索能力和跳出局部最优的能力.虽然 4 种算法的寻优结果都能接近理论值,但在寻优精度方面,TCGWO 算法的寻优平均值对比算法更接近理论最优值,其中对 F_4 、 F_7 、 F_8 、 F_9 、 F_{10} 、 F_{12} 、 F_{15} 等几个函数,TCGWO 算法均能够收敛到理论最优值;在寻优稳定性方面,TCGWO 的标准差远小于对比算法.

为了进一步的对比四种算法的寻优性能,画出了部分函数寻优的迭代曲线,如图 2.在这些函数的求解过程中,对比算法的求解精度与 TCGWO 的求解精度相近.从图 2 可以看出,在求解精度相近时,TCGWO 相对 GWO 具有更快的收敛速度,并且优

于 FPA 与 ABC.

5 结束语

针对灰狼优化(GWO)算法在优化求解过程中存在的局部搜索能力差、易陷入局部最优等问题,本文提出了基于 Tent 混沌序列的灰狼优化算法,并与原始 GWO 以及 FPA、ABC 两种群智能算法进行了比较.仿真结果表明,本文所提出的算法具有更高的寻优精度,更快的收敛速度,和更好的寻优稳定性,在一定程度上解决了原始 GWO 算法存在的局部搜索能力差、易陷入局部最优等问题. TCGWO 算法的可行性与优越性,为下一步的具体应用研究提供很好的参考与借鉴^[8].

参考文献:

[1] Mirjalili S, Mirjalili S M, Lewis A. Grey wolf optimizer[J]. Advances in Engineering Software, 2014, 69 (3): 46-61.
[2] 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.

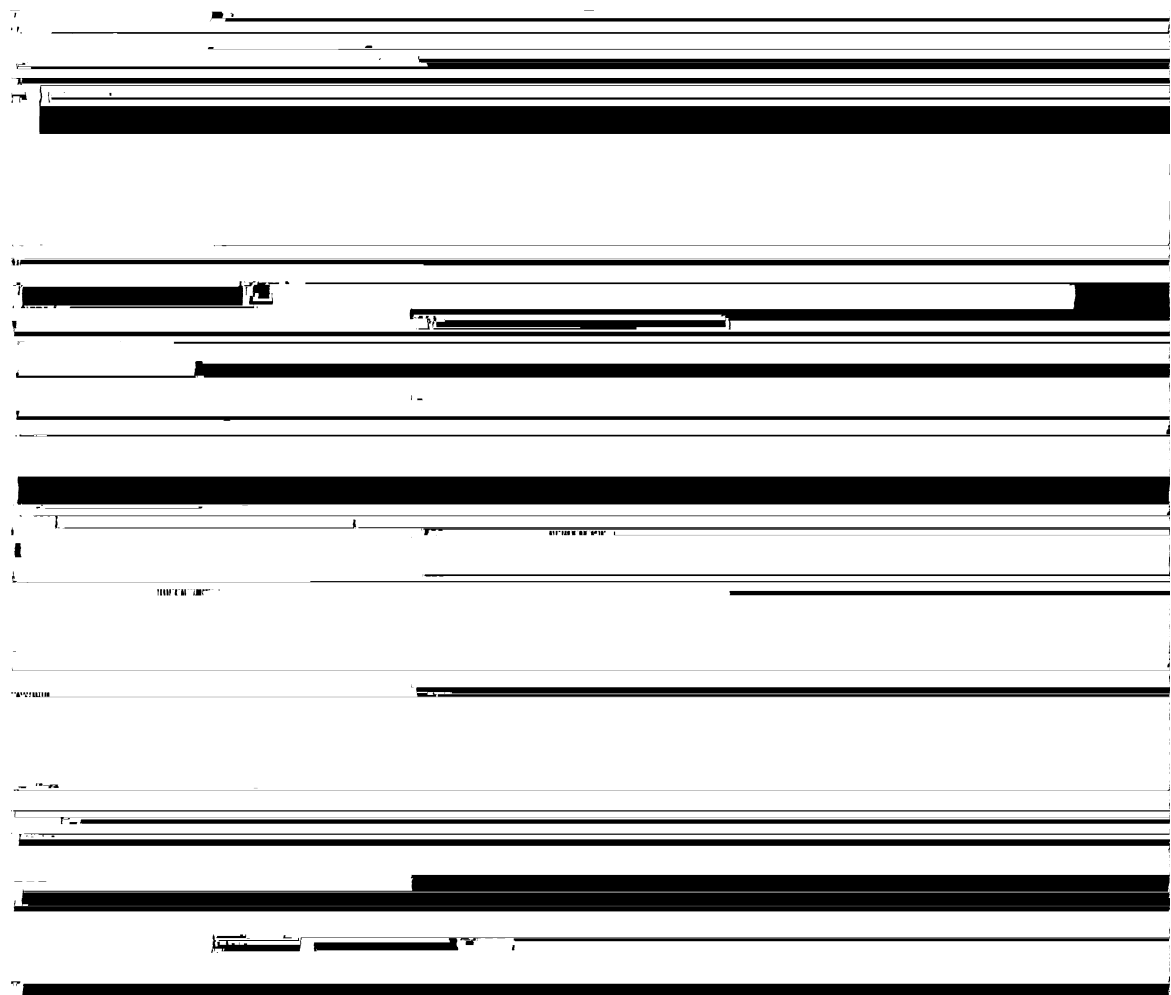


图 2 部分函数寻优进化曲线

[3] 魏政磊,赵辉,韩邦杰,等. 基于自适应 GWO 的多 UCAV 协同攻击目标决策[J]. 计算机工程与应用, 2016 , 52(18):257-261.

[4] Zhu A, Xu C, Li Z, et al. Hybridizing grey wolf optimization with differential evolution for global optimization and test scheduling for 3D stacked SoC[J]. Journal of Systems Engineering and Electronics, 2015, 26 (2):317-328.

[5] Muangkote N, Sunat K, Chiewchanwattana S. An improved grey wolf optimizer for training q-Gaussian radial basis functional-link nets[C]//Proceedings of the 2014 International Computer Science and Engineering Conference. Piscataway, NJ: IEEE,2014:209-214.

[6] 魏政磊,赵辉,李牧东,等. 控制参数值非线性调整策略的灰狼优化算法[J]. 空军工程大学学报(自然科学版), 2016,17(3):68-72.

[7] 单梁,强浩,李军,等 基于 Tent 映射的混沌优化算法 [J]. 控制与决策,2005,20(2):179-182.

[8] Haupt R, Haupt S. Practical genetic algorithm[M]. New York: John Wiley and Sons, 2004;38-39.

作者简介:

张贾奎 男,(1993-),硕士研究生. 研究方向为群智能算法. E-mail:794692508@qq. com.

崔利杰 男,(1979-),副教授. 研究方向为为航空安全、系统可靠性优化.

郭 庆 男,(1994-),硕士研究生. 研究方向为群智能算法.

陈浩然 男,(1994-),硕士研究生. 研究方向为航空安全.