

文章编号:1671-6833(2019)04-0025-07

引入迷失探索与集群分裂机制的改进鸽群优化算法

尚志刚^{1,2,3},王 力^{1,2},李蒙蒙^{1,2},李志辉^{1,2,3}

(1.郑州大学 电气工程学院,河南 郑州 450001; 2.郑州大学 产业技术研究院,河南 郑州 450001; 3.河南省脑科学与脑机接口技术重点实验室,河南 郑州 450001)

摘 要: 鸽群优化算法(pigeon-inspired optimization,PIO)作为一种新兴的优化技术,具有收敛速度快、精度高等优点,但其对于一些具有局部最优值的问题的求解效果并不理想.将自然界中鸽群飞行时的迷失探索和集群分裂机制引入原始鸽群优化算法,提出了一种迷失探索与集群分裂鸽群优化算法(lost and split pigeon-inspired optimization,LSPIO),迷失探索机制的引入加强了算法的全局搜索性能,而集群分裂机制增加了种群多样性.选取 9 个标准测试函数进行算法性能评估,并与标准鸽群算法和粒子群算法进行对比.结果表明,LSPIO 算法在保持良好收敛性质的同时可以有效地避免早熟问题,且提高了种群多样性.

关键词: 鸽群优化; 迷失探索; 集群分裂; 全局搜索; 种群多样性

中图分类号: TP301.6 **文献标志码:** A **doi:** 10.13705/j.issn.1671-6833.2019.04.017

0 引言

生物种群中单个个体的智能水平往往有限,但整个生物群体却表现出处理复杂问题的能力.群体智能就是受自然生物群体行为的启发而产生的一类重要的人工智能方法.Colormi 等从生物进化的机理中受到启发,通过模拟蚂蚁的寻径行为,提出了蚁群优化算法^[1],但其收敛时间过长,易陷入局部最优^[2]; Kennedy 等通过观察鸟群觅食行为提出了粒子群优化算法^[3](particle swarm optimization,PSO),其收敛速度较快,但易陷入局部最优^[4].鸽子具有特殊的归巢能力,它们被认为使用太阳、地球磁场和地标的组合来寻找巢穴. Roberts 等认为,鸽子可能在旅程的不同阶段使用不同的导航工具^[5], Guilford 和他的同事开发了一个数学模型,可以预测鸽子何时从一种导航方式转换到另一种方式.当鸽子开始它们的旅程时,它们可能会更多地依赖类似指南针的工具^[6];在旅途中,当它们找到熟悉的地形或者标志性建筑时,它们转而使用地标^[7-8].受这种鸽群归巢行为

机制的启发,Duan 等提出了一种鸽群优化算法^[9](pigeon-inspired optimization,PIO),其简单、有效的特点^[10]促使它得到了众多学者的重视和研究^[11-12].

鸽群优化算法收敛速度快,精度高,但是与其他一些全局最优算法一样,它也有早熟收敛的现象.针对算法的这一缺陷,现在也有一些改进方法,例如 Sun 等将捕食者机制引入鸽群算法^[13], Li 等引入了模拟退火法^[14], Yang 等将柯西变异引入鸽群算法^[15],这些改进虽然增加了种群规模或种群活跃性,但是对具有局部最优值的函数优化依然不太理想.笔者对自然界鸽子群体飞行行为进行了总结,提出了一种生物行为启发式的引入迷失探索与集群分裂机制的改进鸽群优化算法(lost and split pigeon-inspired optimization,LSPIO).新的算法以当前全局最优位置作为罗盘方向,受鸽子自然飞行过程中可能会迷失方向(感应不到罗盘),从而采用有限时间感知探索机制^[16]的启发,赋予算法中个体一个迷失概率,在迷失期间鸽子会自由探索,增加算法的全局搜索能力.同时借鉴

收稿日期:2018-12-17;修订日期:2019-03-12
基金项目:国家自然科学基金资助项目(61673353)
作者简介:尚志刚(1975—),男,甘肃兰州人,郑州大学教授,博士,主要从事生物医学信息与模式识别研究, E-mail: zhigang_shang@zzu.edu.cn.
通信作者:李志辉(1978—),女,河南濮阳人,郑州大学讲师,博士生,主要从事信号处理与动物强化学习研究, E-mail: lizhrain@zzu.edu.cn.

鸽子飞行过程中发生的集群分裂机制,随机分裂出若干子群进行地标探索,这样既保证了全局搜索性能,也兼顾了局部搜索性能.笔者采用 9 个标准测试函数测试 LSPIO 算法的性能,并与标准鸽群算法和粒子群算法结果进行对比分析,在多个测试函数上的结果表明,笔者提出的算法不仅具有更高的收敛性能,可以有效地避免早熟问题,并且有效提高了种群多样性.

1 鸽群优化算法介绍

在鸽群优化算法中,分别基于鸽子的太阳、地磁导航机制和地标导航机制提出了两种算子:地图罗盘算子和地标算子.在迭代优化过程中,前期使用地图罗盘算子,后期使用地标算子.

(1)地图罗盘算子:鸽子可以通过使用磁感应在大脑中塑造地图.它们将太阳的高度视为指南针来调整方向.

(2)地标算子:当鸽子靠近目的地飞行时,它们将依赖邻近的地标指示方向.

PIO 模型中,每个优化问题的解都是搜索空间中一只虚拟鸽子.每只鸽子都有一个速度矢量和位置矢量来决定它们的当前位置以及飞翔方向和距离,每只鸽子的位置所对应的目标函数值作为该鸽子的适应度(fitness value).所有鸽子个体按照一定的规则在解空间中寻找最优解,即适应度的最大值或最小值,笔者以最小值优化为例进行讨论.

在鸽群优化算法中,首先运行地图和罗盘算子,鸽子*i*的位置和速度分别用 X_i 和 V_i 表示,如下公式为地图和罗盘算子位置速度更新公式:

$$V_i(t) = rand \cdot (X_g(t) - X_i(t - 1)) + V_i(t - 1) \cdot e^{-Rt}; \quad (1)$$

$$X_i(t) = X_i(t - 1) + V_i(t), \quad (2)$$

式中: R 是地图和罗盘因子; $rand$ 是一个随机数; X_g 是当前全局最佳位置,可以通过比较所有鸽子中的所有位置来获得; t 表示当前迭代次数.

地图和罗盘算子运行一段时间后,转为地标算子运行,其中远离目的地的鸽子对地标不熟悉,它们将不再有分辨路径的能力,因而被舍去,然后将剩余鸽子的中心位置 X_c 当作地标,所有看到地标的鸽子直接飞向地标.地标算子运行一段时间后输出最优解.地标算子的位置更新公式如下所示:

$$N_p(t) = \frac{N_p(t - 1) - 1}{2}; \quad (3)$$

$$X_c(t) = \frac{\sum X_i(t) \cdot fitness(X_i(t))}{N_p(t) \cdot \sum fitness(X_i(t))}; \quad (4)$$

$$X_i(t) = rand \cdot (X_c(t) - X_i(t - 1)) + X_i(t - 1), \quad (5)$$

式中: $fitness$ 表示鸽子个体的适应度; N_p 为种群数量.

2 改进的鸽群优化算法

在鸽群优化算法运行过程中后期,如果某只鸽子发现一个当前种群最优位置,其他鸽子将迅速向其靠拢.如果该最优位置为一局部最优点,鸽群就无法在解空间内重新搜索,因此,算法陷入局部最优,出现了所谓的早熟收敛现象.

笔者结合现有研究资料和鸽群放飞实验观察发现,鸽群飞行时存在以下两个特别的行为机制:

(1)鸽子会在归巢导航过程中发生察觉不到罗盘信息而自由探索的情况,而在自由探索过程中,鸽子存在一个有限时间感知探索机制^[16]:鸽子在飞行探索时不会不停转向,而是会沿一个方向飞行一段时间,如果在该时段内不能获得目标方向信息,则会选择反向或根据自身探索到的一些信息而调整方向.

(2)由于某些外界因素(如障碍物、天敌等)的干扰,鸽群归巢过程中会发生集群分裂的情况,分裂后的子群通常会脱离主群飞行,子群飞行中可能会探索到熟悉的建筑或感兴趣的地标.

基于上述机制,笔者对鸽群算法进行改进,提出一种新的改进 LSPIO 算法.首先引入迷失探索机制改进地图罗盘算子以增强算法的全局搜索能力.在原有的算子基础上加入了一个迷失因子 m_1 ,表示鸽子的迷失概率.算法每次迭代都会生成一个随机数,如果该随机数小于 m_1 则进行迷失探索操作.鸽子进入迷失状态时,会进行自由探索,首先随机产生下一时刻的速度方向,经过一定时间的探索飞行后,如果自身适应度没有向希望的方向变化,则改变飞行方向,然后重复迭代一段时间后,恢复可以探测到罗盘的状态.迷失状态时位置速度更新公式:

$$diffitness_i(t) = fitness_i(t) - fitness_i(t - 1). \quad (6)$$

$$b_i = \begin{cases} 0, & diffitness_i(t) < 0. \\ b_i + 1, & diffitness_i(t) \geq 0. \end{cases} \quad (7)$$

$$V_i(t) = \begin{cases} rand \cdot V_i(t - 1), & b_i \leq 2. \\ 2 \cdot (rand - 1) \cdot V_{max}, & b_i > 2. \end{cases} \quad (8)$$

式中: $V_i(t)$ 为当前时刻速度; $V_i(t - 1)$ 为上一时

刻速度; V_{\max} 为个体最大允许速度; $diffitness$ 为适应度差值; b_i 为适应度变化标记. 每次更新位置后, 如果适应度没有减少, 则标记 b_i 增加 1 点, 连续 3 次适应度没有减少 (即 $b_i \geq 2$), 则更换飞行方向进行探索, 如果适应度减少, 则保持飞行方向探索且标记 b_i 置零.

在此基础上, 引入集群分裂机制提高种群个体多样性. 在算法中定义了一个分裂因子 m_2 , 表示鸽群分裂概率. 算法迭代过程中, 为每个个体生成一个随机数, 如果该随机数小于 m_2 , 则进行集群分裂操作, 并将子群分裂标志 s 记为 1, 总体子群个数 $L \leq 3$. 对于第 j 个子群, 其适应度最优个体为 X_g^j . 该子群内其他个体会跟随 X_g^j 去寻找一定范围内的局部最优解, 并与主群中的最优解进行比较, 更优者作为新的全局最优解, 子群在分裂迭代次数 $r > 10$ 后重新与主群建立联系, 并入主群飞行. 其速度更新公式:

$$V_i = \begin{cases} V_i(t-1)e^{-Rt} + rand \cdot (X_g^j - X_i(t-1)), & rand < m_2; \\ V_i(t-1)e^{-Rt} + rand \cdot (X_g - X_i(t-1)), & rand \geq m_2. \end{cases} \quad (9)$$

综合的鸽群优化算法, 算法流程如下:

(1) 初始化种群和参数, 随机赋予每只虚拟鸽子的位置矢量和速度矢量并计算其适应度值, 确定全局最优.

(2) 判断如果迭代次数 $p = gen_{\max}$, 则 $p = 0$, 转入步骤(7), 否则 $p = p + 1, i = 1$ 并转入步骤(3).

(3) 判断鸽子 i 的分裂状态, 若处于分裂状态, 则根据式(9)更新其速度和位置并转入步骤(6); 否则转入步骤(4).

(4) 生成随机数 $rand$, 判断是否 $rand < m_1$, 若是, 对该鸽子进行迷失操作, 根据式(6)、(7)、(8)更新其速度和位置; 否则根据式(1)、(2)更新该鸽子的速度和位置.

(5) 生成随机数 $rand$, 判断是否 $rand < m_2$ 且总子群数小于 3, 如果是, 该鸽子及其 k 邻域的鸽子设置为分裂状态.

(6) 计算鸽子适应度, $i = i + 1$, 判断是否 $i > N_{\max}$, 若是, 转入步骤(2); 否则转入步骤(3).

(7) 根据式(3)、(4)、(5)更新位置, 计算鸽子适应度, 确定最优解, 迭代更新直至 $p = gen_{\max}$, 输出结果.

算法流程图如图 1 所示. 迷失因子 m_1 和分裂因子 m_2 的选取直接影响算法的收敛性能, 如果选

取的因子过大, 则种群会有更多的机会进行探索, 对全局的搜索能力更强, 但收敛速度会变慢; 反之收敛速度加快, 但可能陷入局部最优解. 笔者建议 m_1 与 m_2 的取值分别为 0.2 和 0.1.

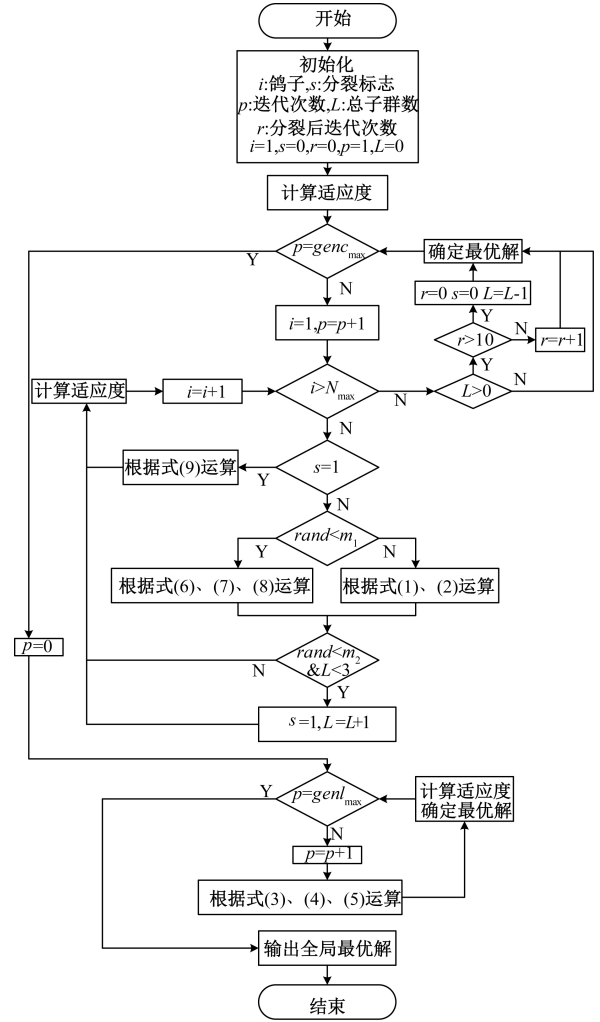


图 1 LSPIO 算法流程图

Fig.1 Flow chart of LSPIO algorithm

3 改进优化算法仿真实验

为测试笔者提出的 LSPIO 算法的有效性, 下面将通过 9 个典型的标准测试函数^[17]展开实验, 同时与 PIO 算法和 PSO 算法进行比较, 3 种算法种群规模和总迭代次数相同, 并且在每次迭代过程中, 每个个体都只评价一次个体适应度. 9 个测试函数公式如下, 图 2 为测试函数三维示例图.

(1) Sphere 函数:

$$F1 = \sum_{i=1}^{10} x_i^2, \quad -600 \leq x_i \leq 600. \quad (10)$$

(2) Three-hump camel 函数:

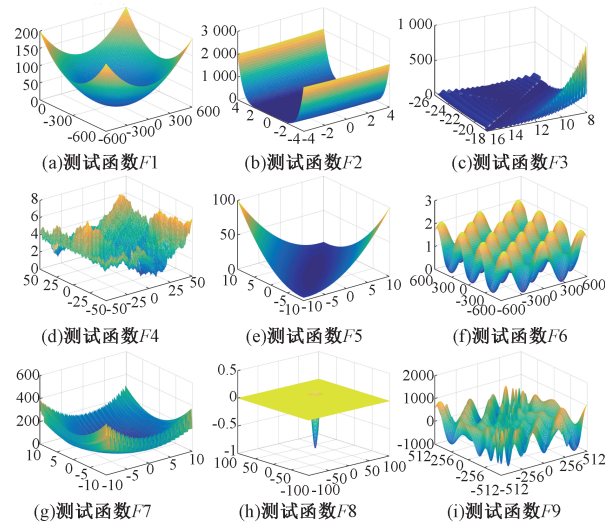


图 2 测试函数三维示例图

Fig.2 3D example of test function

$$F2 = 2x_1^2 - 1.05x_1^4 + \frac{x_1^6}{6} + x_1x_2 + x_2^2, \\ -5 \leq x_1, x_2 \leq 5. \quad (11)$$

(3) Rotated Schaffers 函数:

$$F3 = \left(\frac{1}{4} \sum_{i=1}^4 (\sqrt{z_i} + \sqrt{z_i} \sin^2(50z_i^{0.2})) \right)^2,$$

$$\text{其中, } z_i = \sqrt{y_i^2 + t_{i+1}^2}, i = 1, \dots, 5, \\ y = \mathbf{A}^{10} \mathbf{M}_2 \mathbf{T}_{\text{asy}}^{0.5} (\mathbf{M}_1 (\mathbf{x} - \mathbf{o})). \quad (12)$$

(4) Rotated Weierstrass 函数:

$$F4 = \sum_{i=1}^5 \left(\sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k (z_i + 0.5))] \right) - \\ 5 \sum_{k=0}^{k_{\max}} [a^k \cos(2\pi b^k \cdot 0.5)],$$

其中, $a = 0.5; b = 3; k_{\max} = 20;$

$$z = \mathbf{A}^{10} \mathbf{M}_2 \mathbf{T}_{\text{asy}}^{0.5} \left(\mathbf{M}_1 \frac{0.5(\mathbf{x} - \mathbf{o})}{100} \right). \quad (13)$$

(5) Matyas 函数:

$$F5 = 0.26(x_1^2 + x_2^2) - 0.48x_1x_2, \\ -10 \leq x_i \leq 10. \quad (14)$$

(6) Griewank 函数:

$$F6 = \sum_{i=1}^5 \frac{x_i^2}{4000} - \prod_{i=1}^5 \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1, \\ -600 \leq x_i \leq 600. \quad (15)$$

(7) Levy 函数:

$$F7 = 2(x_2 - 1)^2(1 + \sin^2(2\pi x_2)) + \\ (x_1 - 1)^2(1 + \sin^2(3\pi x_2)) + \\ \sin^2(3\pi x_1), -10 \leq x_1, x_2 \leq 10. \quad (16)$$

(8) Easom 函数:

$$F8 = \cos x_1 \cos x_2 \cdot \\ \exp(-(x_1 - \pi)^2 + (x_2 - \pi)^2), \\ -100 \leq x_1, x_2 \leq 100. \quad (17)$$

(9) Eggholder 函数:

$$F9 = -(x_2 + 47) \sin\left(\sqrt{\left|x_2 + \frac{x_1}{2} + 47\right|}\right) - \\ x \cdot \sin(\sqrt{|x_1 - (x_2 + 47)|}), \\ -512 \leq x, y \leq 512. \quad (18)$$

其中, $\mathbf{o} = [o_1, o_2, \dots, o_D]^T$ 表示随机分布在 $[-80, 80]^D$ 的数, D 表示维数; \mathbf{A} 表示以第 i 个对角线元素为 $\lambda_{ii} = a^{\frac{i-1}{D-1}}$ 的 D 维对角线矩阵.

其中函数 $F1$ 、 $F2$ 、 $F5$ 是较为简单的单模态函数, 函数 $F3$ 和 $F4$ 主要考虑了平移和旋转对优化算法的影响, $F6$ 、 $F7$ 、 $F9$ 是具有多个局部最优的函数, $F8$ 的全局最小值较搜索空间小.

实验设置参数如下: 3 种算法种群大小都相同, 粒子群算法 (PSO) 学习因子 $c_1 = 2, c_2 = 2$; PIO 算法地图罗盘因子 $R = -0.02$; LSPIO 算法迷失因子和分裂因子 $m_1 = 0.2, m_2 = 0.1$, 分别将 9 个测试函数当作目标函数, 进行测试.

表 1 列出了用 3 种算法求解上述优化问题运行 10 次后得到的平均函数最优解. 从表 1 可以看出, 对于简单函数和具有局部最优陷阱函数, LSPIO 算法的优化结果优于其他两种算法, 而对于旋转函数, LSPIO 的表现和其他两种算法相当.

图 3 为 3 种算法对函数优化 10 次后的平均最佳适应度演化曲线. 为了验证上述结果差异是不是显著性的, 笔者将上述所得数据进行假设检验. 由于秩和检验^[18]具有易于理解、易于计算的优点, 笔者选用秩和检验进行验证. 秩和检验的基本思想: 若检验假设成立, 则差值的总体分布应是对称的, 故正负秩和不应悬殊. 分别对 LSPIO、PIO 和 LSPIO、PSO 进行秩和检验, 当前者优于后者并且检测 P 值小于 0.05, 则标“+”号, 表示前者优于后者是显著性的. 当后者优于前者并且 P 值小于 0.05, 则标“-”号, 表示后者优于前者是显著性的. 若检测 $P \geq 0.05$, 则认为它们差异不显著. 表 2 是统计检验结果, 从表 2 中可以看出, LSPIO 在其中 7 个函数上的表现优于 PIO, 证明 LSPIO 比 PIO 收敛性能更强; 而与 PSO 相比, LSPIO 也在 6 个函数上表现出更优的性能. 综上所述, 笔者提出的 LSPIO 算法表现出较好的全局搜索能力, 可以有效避免早熟收敛问题.

表 1 3 种算法运行 10 次平均最优解

Tab.1 The average optimal solution of three algorithms with 10 runs

函数	最优解	PIO	PSO	LSPIO
$F1$	0	5.16E-14±5.2E-29	4.58E-15±3E-30	0±2.7E-50
$F2$	0	7.5E-05±2.5E-10	1.67E-12±1.3E-25	1.1E-16±6.6E-35
$F3$	0	7.3E-07±3.7E-11	2.2E-07±1.03E-12	8.8E-08±1.2E-12
$F4$	0	0.7E+00±0.5E+00	1.04E+00±0.9E+00	0.93E+00±0.1E+00
$F5$	0	2.1E-11±1.7E-22	4.9E-14±2.1E-29	1.2E-17±2.2E-36
$F6$	0	0.26E-01±0.7E-02	0.25E-01±0.3E-02	0.2E-01±9.2E-02
$F7$	0	0.1E-03±4.4E-08	1.1E-11±2.6E-24	2.4E-15±1.9E-31
$F8$	-1	-0.98E+00±1.7E-09	-0.99E+00±1.65E-24	-1E+00±1.6E-24
$F9$	-959.640	-9.24E+02±7.3E+02	-9.20E+02±5.1E+2	-9.51E+02±0.8E+0

注:加黑字体表示最接近最优解的优化结果.

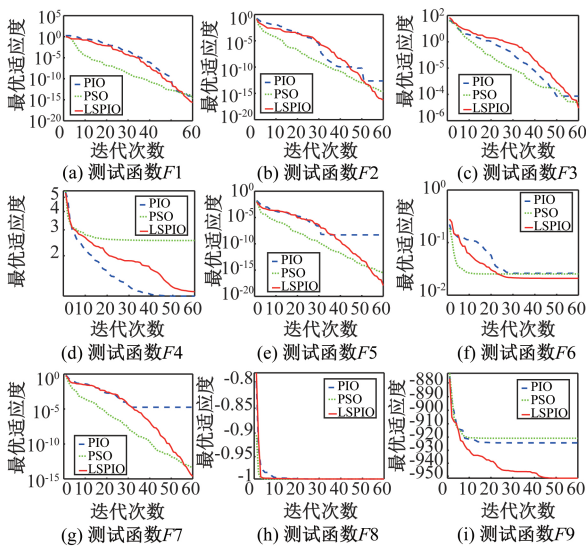


图 3 最佳适应度演化曲线

Fig.3 Best fitness evolution curve

表 2 3 种算法运行 10 次结果的统计检验

Tab.2 Statistical test of 10 results of three algorithms				
函数	LSPIO vs PIO		LSPIO vs PSO	
	P 值	优劣性比较	P 值	优劣性比较
$F1$	1.8E-04	+	1.8E-04	+
$F2$	1.8E-04	+	1.6E-04	+
$F3$	7.6E-06	+	3.1E-04	+
$F4$	0.3E+00	=	0.9E+00	=
$F5$	2.4E-04	+	1.8E-04	+
$F6$	1.8E-01	=	3.8E-01	=
$F7$	1.5E-04	+	1.2E-04	+
$F8$	1.8E-04	+	0.8E+0	=
$F9$	0.1E-04	+	4.1E-04	+
-	0		0	
+	7		6	
=	2		3	

注:“+”号表示 LSPIO 显著优越;“-”号表示 PIO/PSO 显著优越;“=”表示两者优劣性无显著差异.

为了评估算法的种群多样性,定义了每一代的种群分布散度,其公式为:

$$PD = \frac{1}{D} \sum_{i=1}^D \left(\frac{1}{N} \sum_{j=1}^N (x_{ij} - \bar{x}_i)^2 \right), \quad (19)$$

式中: D 为个体总维数; N 为种群个体数.

笔者计算了迭代过程中每代种群的分布散度,如图 4 所示为 9 个测试函数分别运行 10 次后取平均得到的种群分布散度-迭代次数曲线.可以看出在迭代过程中,LSPIO 算法的种群分布散度显著高于 PSO 和 PIO,从而说明 LSPIO 算法保证了较好的种群多样性.

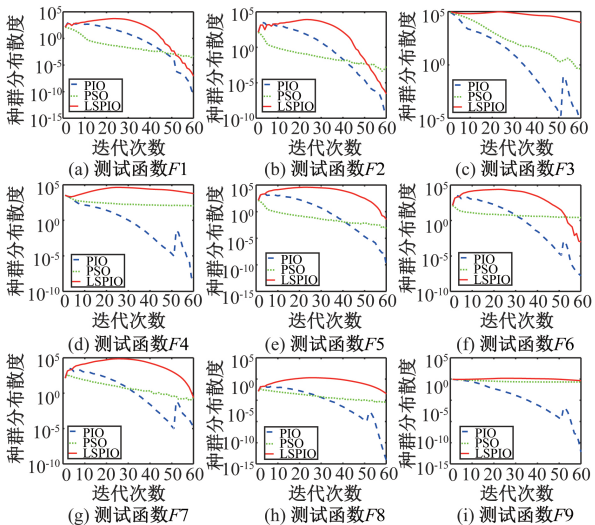


图 4 种群分布散度演化曲线

Fig.4 Evolution curve of divergence of population distribution

4 结论

笔者针对鸽群优化算法的早熟收敛问题,提

出了一种迷失探索与集群分裂机制的改进鸽群优化算法.这种优化算法的灵感来源于实际中鸽群飞行的迷失探索和集群分裂现象.为了测试和对比算法性能,笔者选取 9 个标准测试函数进行函数优化实验,并与标准鸽群算法和粒子群算法进行了对比分析,在多个测试函数上的结果表明:笔者提出的 LSPIO 算法具有良好的全局搜索能力,有效避免了早熟收敛问题,且较好地保持了迭代过程中的种群多样性.

笔者提出的算法虽然可以有效避免早熟收敛问题,但是它的收敛速度并不理想,并且只考虑了低维问题,接下来笔者将针对这两方面的问题进行深入研究,进一步进行算法改进以提高收敛速度和优化性能.

参考文献:

[1] COLORNI A, DORIGO M, MANIEZZO V. Distributed optimization by ant colonies[C] // Proceedings of the First European Conference on Artificial Life. Paris: Elsevier, 1991:134-142.

[2] 毛晓波,张勇杰,陈铁军.基于蚁群及空间邻域信息的 FCM 图像分割方法[J].郑州大学学报(工学版),2014,35(1):1-4.

[3] KENNEDY J, EBERHART R. Particle swarm optimization [C]// Proceedings of ICNN' 95-International Conference on Neural Networks. Perth, WA: IEEE, 1995:1942-1948.

[4] 梁静,宋慧,瞿博阳,等.基于改进粒子群算法的路径优化问题研究[J]. 郑州大学学报(工学版), 2014, 35(1): 34-38.

[5] ROBERTS S, GUILFORD T, REZEK L, et al. Positional entropy during pigeon homing I : application of Bayesian latent state modelling [J]. Journal of theoretical biology, 2004, 227(1):39-50.

[6] KIEPENHEUER J. Inversion of the magnetic field during transport: its influence on the homing behavior of pigeons [M]. Berlin: Springer Berlin Heidelberg, 1978.

[7] BALDACCINI N E, BENVENUTI S, FIASCHI V, et al. New data on the influence of olfactory deprivation on the homing behavior of pigeons [J]. Olfaction &

taste symposium, 1975:351-353.

[8] KIEPENHEUER J. The homing behavior of pigeons raised in a reversed magnetic field [M]. Berlin: Springer Berlin Heidelberg, 1984.

[9] DUAN H B, QIAO P X. Pigeon-inspired optimization: a new swarm intelligence optimizer for air robot path planning[J]. International journal of intelligent computing and cybernetics, 2014, 7(1):24-37.

[10] 周子为,段海滨,范彦铭. 仿雁群行为机制的多无人机紧密编队[J]. 中国科学(技术科学), 2017,47(3): 230-238.

[11] DOU R, DUAN H B. Pigeon inspired optimization approach to model prediction control for unmanned air vehicles[J]. Aircraft engineering & aerospace technology, 2015, 88(1):108-116.

[12] 段海滨,杨之元. 基于柯西变异鸽群优化的大型民用飞机滚动时域控制[J]. 中国科学(技术科学), 2018,48(3): 277-288.

[13] SUN H, DUAN H B. PID controller design based on prey-predator pigeon-inspired optimization algorithm [C]// IEEE International Conference on Mechatronics and Automation. Tianjin: IEEE, 2014:1416-1421.

[14] LI C, DUAN H B. Target detection approach for UAVs via improved pigeon-inspired optimization and edge potential function[J]. Aerospace science & technology, 2014, 39:352-360.

[15] YANG Z Y, DUAN H B, FAN Y M, et al. Automatic carrier landing system multilayer parameter design based on cauchy mutation pigeon-inspired optimization [J]. Aerospace science & technology, 2018, 79: 518-530.

[16] MOURITSEN H, HEYERS D, GUNTURKUN O. The neural basis of long-distance navigation in birds[J]. Annual review of physiology, 2015, 78:133-154.

[17] WANG Y, Li B, WEISE T, et al. Self-adaptive learning based particle swarm optimization[J]. Information sciences, 2011, 181(20):4515-4538.

[18] PEROLAT J, COUSO I, LOQUIN K, et al. Generalizing the wilcoxon rank-sum test for interval data [J]. International journal of approximate reasoning, 2015, 56:108-121.

Improved Pigeon Herd Optimization Algorithm with Lost Exploration and Cluster Splitting Mechanism

SHANG Zhigang^{1,2,3}, WANG Li^{1,2}, LI Mengmeng^{1,2}, LI Zhihui^{1,2,3}

(1.School of Electrical Engineering, Zhengzhou University, Zhengzhou 450001, China; 2.Industrial Technology Research Institute, Zhengzhou University, Zhengzhou 450001, China; 3.Henan Key Laboratory of Brain Science and Brain-Computer Interface Technology, Zhengzhou 450001, China)

Abstract: As an emerging optimization technology, pigeon inspired optimization (PIO) algorithm had the advantages of fast convergence and high precision. But it was not ideal for some problems with local optimal values. By introducing lost & exploration and cluster splitting mechanisms of natural flying pigeons, an improved PIO algorithm based on lost & exploration and cluster splitting (LSPIO) was proposed in this paper. The lost & exploration mechanism enhanced the global search performance of the algorithm, and the cluster splitting mechanism increased the diversity of the population. In this paper, 9 standard test functions were selected for algorithm performance evaluation. Compared with standard pigeon group algorithm and particle swarm algorithm, the results showed that the new LSPIO algorithm could effectively avoid premature problems, improve the population diversity while maintaining good convergence properties.

Key words: pigeon-inspired optimization; lost & exploration; cluster splitting; global search; population diversity