

基于认知多样性变异的鸡群算法协同优化异步实现

肖 亮¹ 刘思彤²

(东北石油大学地球科学学院 大庆 163318)¹ (东北大学秦皇岛分校资源与材料学院 秦皇岛 066004)²

摘 要 从小鸡更新公式、优化方式和基于认知多样性变异三方面改进鸡群算法。在小鸡位置更新过程中加入自我学习系数,并向所在种群公鸡学习,同时对未知空间进行探索;采用逆序协同优化异步实现策略提高算法解决更高维度问题的能力;充分利用个体认知多样性,使个体最优以一定概率发生变异,从而带领群体逃离局部最优,收敛到全局最优。Benchmark function 测试表明,改进的鸡群算法优于其他优化算法。模型数据反演结果表明,该算法具有很强的全局搜索能力,反演精度较高,同时抗噪能力很强。

关键词 群体智能,鸡群算法,协同优化,波阻抗反演

中图分类号 TP315 文献标识码 A DOI 10.11896/j.issn.1002-137X.2017.06.001

Asynchronous Collaborative Chicken Swarm Optimization with Mutation Based on Cognitive Diversity

XIAO Liang¹ LIU Si-tong²

(College of Geoscience, Northeast Petroleum University, Daqing 163318, China)¹

(School of Resources and Materials, Northeastern University at Qinhuangdao, Qinhuangdao 066004, China)²

Abstract The standard chicken swarm optimization is improved from the following three aspects: chick-update formula, optimization method and mutation based on cognitive diversity. Self-learning factor is added to chick-update formula. It is assumed that chicks learn from their own roosters respectively, and meanwhile the unknown space is explored. Asynchronous collaborative optimization strategy is adopted with inverted order to improve capacity of solving higher-dimensions problems. Self-cognitive diversity is taken full advantage to make sure the pbests mutate at a certain probability to lead the swarm to escape from the local optimum to converge to the global optimum. Benchmark function test indicates ICSO is better than other optimization algorithms. Model seismic data inversion shows strong global search ability, high precision and strong antinoise ability as well.

Keywords Swarm intelligence, CSO, Co-optimization, Impedance inversion

1 引言

仿生元启发式优化算法在解决许多优化应用问题时展现出了独特的优势^[1-2]。如基于基因选择的遗传算法^[3]、基于鸟群觅食的粒子群算法^[4-5]、基于蚂蚁寻路的蚁群算法^[6]、鱼群觅食的鱼群算法^[7]、蝙蝠算法^[1,8]以及差分演化算法^[2]等^[9-12]。它们利用优化问题对精度和不确定性的容忍性以较低的计算成本获得可接受的解。新的算法不断涌现,比如磷虾群优化算法^[13]、蜘蛛优化算法^[14]等。文献^[15]提出了一种全新的基于鸡群搜索觅食的随机优化算法,它模拟鸡群的等级制度和鸡群的觅食行为。整个鸡群由若干个子群组成,而每个子群有一只公鸡、若干只母鸡和小鸡。各等级的鸡有不同的移动规则,而且鸡群之间还存在竞争。相比其他优化算法,该算法有较快的收敛速度和较高的收敛精度。文献^[16-19]对其进行了改进,但它们都只是解决了维度较低的问题。本文也对其进行了改进,并通过测试集函数表明该改进鸡群算法能够解决更高维度的问题,同时避免了早熟收敛和陷入局部极值。

2 标准鸡群算法

2.1 鸡群算法的等级关系

1) 鸡群中存在若干子群,每个子群包括一只公鸡、若干只母鸡和小鸡。

2) 子群的划分和公鸡、母鸡、小鸡的定义取决于它们适应度的大小,鸡群中适应度最好的一些个体作为公鸡,适应度最差的一些个体作为小鸡,其余的个体作为母鸡。母鸡随机选择子群,小鸡随机选择母鸡。

3) 子群中母子关系和领头关系一经建立则保持不变,直到 G 次迭代后再由适应度重新确定。母鸡围绕公鸡寻找食物,小鸡在母鸡周围寻找食物。

2.2 鸡群算法的更新

假设 N_r, N_h, N_c, N_m 分别代表公鸡、母鸡、小鸡及与小鸡对应的母鸡的数目(注: N_m 可以不等于 N_h , 即一只母鸡可能拥有不止一只小鸡)。所有个体 ($N = N_r + N_h + N_c$) 都由它们的位置 $x_{i,j}^t$ ($i \in [1, \dots, N], j \in [1, \dots, D]$) 所描述,表示 t 次迭代第 i 个个体在 D 维空间中第 j 维所处的位置。

本文受东北石油大学研究生创新科研项目(YJSCX2016-005NEPU)资助。

肖 亮(1992—),男,硕士生,主要研究方向为地震资料数字处理,E-mail: xiaoliang851475324@163.com(通信作者)。

2.2.1 公鸡更新

$$x_{i,j}(t+1) = x_{i,j}(t) \cdot (1 + \text{Randn}(0, \sigma^2)) \quad (1)$$

$$\sigma^2 = \begin{cases} 1, & \text{if } f_i \leq f_k \\ \exp(\frac{f_i - f_k}{f_i + \epsilon}), & \text{其他} \end{cases} \quad (2)$$

其中, $\text{Randn}(0, \sigma^2)$ 为均值为 0、标准差为 σ^2 的高斯随机函数; ϵ 为一个很小的常数; $k \in [1, Nr]$, $k \neq i$, 为除自身外随机选择的公鸡的索引; f 为对应个体的适应度。

2.2.2 母鸡更新

$$x_{i,j}(t+1) = x_{i,j}(t) + C_1 \cdot \text{Rand} \cdot (x_{r1,j}(t) - x_{i,j}(t)) + C_2 \cdot \text{Rand} \cdot (x_{r2,j}(t) - x_{i,j}(t)) \quad (3)$$

$$C_1 = \exp((f_1 - f_{r1}) / \text{abs}(f_i + \epsilon)) \quad (4)$$

$$C_2 = \exp(f_{r2} - f_i) \quad (5)$$

其中, Rand 为 $[0, 1]$ 均匀随机分布函数, $r1$ 为母鸡所在种群的公鸡的索引, $r2$ 为整个种群中公鸡或母鸡中随机选择的一个索引, 且 $r1 \neq r2$ 。

2.2.3 小鸡更新

$$x_{i,j}(t+1) = x_{i,j}(t) + F \cdot (x_{m,j}(t) - x_{i,j}(t)) \quad (6)$$

其中, m 为小鸡对应的母鸡; F 为 $(0, 2)$ 的随机数, 表示小鸡跟随母鸡学习的系数, 且每个小鸡的学习系数不同。

3 改进的鸡群算法

3.1 标准算法分析

根据对 10 个标准测试集函数的分析, 鸡群算法中严格的等级制度使得当公鸡陷入局部最优后, 母鸡和小鸡随之也陷入局部最优, 而且在没有其他不可行解加入的情况下无法跳出。鸡群算法在低维度情况下能够快速收敛到全局最优, 但在高维度情况下会出现早熟收敛和陷入局部最优的情况, 特别是 Rosenbrock 函数, 在迭代几次后会迅速收敛到局部最优且不能跳出。

3.2 改进的鸡群算法

通过对标准鸡群算法的分析, 同时结合从其他演化算法吸取的经验, 本文对标准鸡群算法进行如下改进。

3.2.1 小鸡更新公式的改进

小鸡除了跟随母鸡寻找食物之外, 也要向种群公鸡学习, 同时对未知世界充满兴趣, 且这个趋势随着迭代次数的增加而增大。改进后的位置更新公式如下:

$$x_{i,j}(t+1) = w \cdot x_{i,j} + C_1 \cdot (x_{m,j}(t) - x_{i,j}(t)) + C_2 \cdot (x_{r,j}(t) - x_{i,j}(t)) + Q \cdot (\text{Random} - x_{i,j}(t)) \quad (7)$$

其中, w 是自我学习系数, 类似于 PSO 中的惯性权重; $C_1 = 0.5 \times (1 - w)$ 为母鸡跟随系数; $C_2 = 0.3 \times (1 - w)$ 为公鸡跟随系数; Random 为解空间随机位置; Q 为对未知世界的好奇度, $Q = 0.2 \times (1 - w)$, w 随着迭代次数的增加而减小。适当增加种群中小鸡的数目, 能在迭代后期产生不可行解而跳出局部最优。

3.2.2 优化方式的改进

采用逆序的协同优化异步实现策略来提高算法解决更高维度问题的能力。本文从问题解的尾部向首部进行优化。算法中存在多个鸡群, 每个鸡群优化问题解的维度大小 (D_c) 和协同次数 (C_o) 由用户自定义。协同优化过程中, 除待优化维度外, 其他维度被对应群体中适应度最差的个体的维度所替换。迭代过程中, 异步实现使得每个个体能及时与当前最好

个体进行信息共享交流, 且在寻优过程中具有不同步性。

3.2.3 基于认知多样性的变异

定义当前群体位置多样性为暂态多样性或瞬态多样性, 个体目标多样性为认知多样性。当群体处于高暂态多样性、低认知多样性时, 表明算法正处于全局搜索向局部搜索转换的过程, 即正处于收敛状态。如果处于收敛状态而又找不到需要的解, 则意味着已陷入局部最优, 若此时瞬态多样性高, 则在位置上加噪作用很小, 应将噪声加在目标上。因此针对算法早熟收敛且易陷入局部最优的问题, 引入遗传变异的思想, 即在个体最优上以一定概率 (Mu) 发生变异。

3.3 仿真算法流程

1) 定义相关参数 D_c, Nr, Nh, N_c, N_m 等, 初始化鸡群, 计算每个个体的适应度, 并按适应度排序, 确定分组与个体的关系;

2) 提取各群体最差适应度维度信息, 除待优化维度外, 其他维度被对应群体中适应度最差的个体对应的维度替换, 更新适应度;

3) 使用式 (1)、式 (3)、式 (7) 更新对应个体的位置并计算个体适应度, 更新群体中个体的当前最优和全局最优, 并以一定概率变异个体最优;

4) 如果 $t \% G = 0$, 则重新建立等级关系, 重复步骤 1) 一步骤 4), 当满足迭代终止条件时, 输出解。

4 仿真实验

4.1 参数设置

表 1 列出了 10 个标准测试集函数。为了对本文改进的鸡群算法进行有效性验证, 将其与标准粒子群优化算法、标准鸡群优化算法和采用了本文相同思想的改进粒子群算法 IP-SO 进行了仿真对比。为了达到客观性, 各算法对每个测试集函数独立运行 50 次, 种群大小都为 200 个个体, 最大迭代次数都为 1000, 分别在 $D=20$ 和 $D=200$ 两种情况下对所有的测试集函数进行求解。4 种算法的其他相关参数如表 2 所列。

表 1 10 个标准测试集函数

标准测试集函数	代号	定义域范围	最优解
Sphere	F1	$[-5, 12, 5, 12]$	0
Weighted sphere	F2	$[-5, 12, 5, 12]$	0
Rotated hyper-ellipsoid	F3	$[-65, 536, 65, 536]$	0
Moved axis parallel hyper-ellipsoid	F4	$[-5, 12, 5, 12]$	0
Rosenbrock	F5	$[-2, 048, 2, 048]$	0
Rastrigin	F6	$[-5, 12, 5, 12]$	0
Schwefel	F7	$[-500, 500]$	$-n \times 418, 9829$
Griewangk	F8	$[-600, 600]$	0
Sum of different power	F9	$[-1, 1]$	0
Ackley	F10	$[-1, 1]$	0

表 2 算法的相关参数设置

算法	参数
PSO	$c1 = c2 = 1, 49445, w = 0.729$
IPSO	$c1 = c2 = 2, 0, w \in [0, 4, 0, 9], Mu \in [0, 001, 0, 009], D_c = 1, C_o = 10$
CSO	$Nr = 0, 05N, Nh = 0, 5N, N_c = N - Nr - Nh, G = 5, F \in \text{rand}(0, 2)$
ICSO	$Nr = 0, 05N, Nh = 0, 5N, N_c = N - Nr - Nh, G = 5, C1 = 0, 5 \times (1 - w), C2 = 0, 3 \times (1 - w), Q = 0, 2 \times (1 - w), w \in [0, 4, 0, 9], Mu \in [0, 001, 0, 009], D_c = 1, C_o = 10$

4.2 仿真分析

50 次独立运行的结果如表 3、表 4 所列。

表 3 改进算法和其他算法优化结果的对比($D=20$)

代号	算法	最好值	最差值	平均值	标准差
F1	PSO	7.30884×10^{-24}	2.73761×10^{-21}	6.66691×10^{-22}	3.15971×10^{-21}
	IPSO	3.07373×10^{-164}	3.56057×10^{-160}	$4.631186 \times 10^{-161}$	$3.294335 \times 10^{-160}$
	CSO	5.45399×10^{-93}	3.83147×10^{-78}	3.83160×10^{-79}	3.63484×10^{-78}
	ICSO	5.19519×10^{-161}	3.40450×10^{-147}	5.26510×10^{-148}	3.50433×10^{-147}
F2	PSO	2.32091×10^{-22}	5.87995×10^{-19}	9.21804×10^{-20}	5.70177×10^{-19}
	IPSO	2.00196×10^{-149}	3.04395×10^{-140}	3.04399×10^{-141}	2.88774×10^{-140}
	CSO	4.17123×10^{-99}	6.73061×10^{-81}	6.96163×10^{-82}	6.36458×10^{-81}
	ICSO	2.20978×10^{-161}	3.75340×10^{-144}	3.75340×10^{-145}	3.56079×10^{-144}
F3	PSO	1.16956×10^{-12}	1.71173×10^{-09}	4.13946×10^{-10}	1.77890×10^{-09}
	IPSO	2.79917×10^{-48}	7.72086×10^{-43}	1.73732×10^{-43}	8.75843×10^{-43}
	CSO	4.70869×10^{-61}	3.53437×10^{-35}	3.53498×10^{-36}	3.35293×10^{-35}
	ICSO	$0.00000 \times 10^{+00}$	$0.00000 \times 10^{+00}$	$0.00000 \times 10^{+00}$	$0.00000 \times 10^{+00}$
F4	PSO	2.95535×10^{-21}	1.26578×10^{-18}	1.54050×10^{-19}	1.17593×10^{-18}
	IPSO	2.91668×10^{-155}	2.72960×10^{-144}	4.54343×10^{-145}	2.56309×10^{-144}
	CSO	3.06061×10^{-91}	1.24926×10^{-80}	1.25100×10^{-81}	1.18497×10^{-80}
	ICSO	6.34201×10^{-161}	9.06355×10^{-158}	2.29733×10^{-158}	1.03314×10^{-157}
F5	PSO	$4.21706 \times 10^{+00}$	$7.12886 \times 10^{+00}$	$6.58541 \times 10^{+00}$	$2.88017 \times 10^{+00}$
	IPSO	1.23714×10^{-03}	$4.27886 \times 10^{+00}$	$1.54576 \times 10^{+00}$	$5.36161 \times 10^{+00}$
	CSO	$7.14290 \times 10^{+00}$	$7.46331 \times 10^{+00}$	$7.27336 \times 10^{+00}$	2.90407×10^{-01}
	ICSO	2.46550×10^{-03}	$5.07142 \times 10^{+00}$	$1.18950 \times 10^{+00}$	$4.57752 \times 10^{+00}$
F6	PSO	$5.96975 \times 10^{+00}$	$2.28840 \times 10^{+01}$	$1.44269 \times 10^{+01}$	$1.65445 \times 10^{+01}$
	IPSO	$0.00000 \times 10^{+00}$	$0.00000 \times 10^{+00}$	$0.00000 \times 10^{+00}$	$0.00000 \times 10^{+00}$
	CSO	$0.00000 \times 10^{+00}$	$0.00000 \times 10^{+00}$	$0.00000 \times 10^{+00}$	$0.00000 \times 10^{+00}$
	ICSO	$0.00000 \times 10^{+00}$	$0.00000 \times 10^{+00}$	$0.00000 \times 10^{+00}$	$0.00000 \times 10^{+00}$
F7	PSO	$-3.01470 \times 10^{+03}$	$-2.30554 \times 10^{+03}$	$-2.60096 \times 10^{+03}$	6.30605×10^{-02}
	IPSO	$-8.37966 \times 10^{+03}$	$-8.37966 \times 10^{+03}$	$-8.37966 \times 10^{+03}$	5.75215×10^{-12}
	CSO	$-6.36298 \times 10^{+03}$	$-5.01739 \times 10^{+03}$	$-5.71437 \times 10^{+03}$	$1.24272 \times 10^{+03}$
	ICSO	$-8.37966 \times 10^{+03}$	$-8.37966 \times 10^{+03}$	$-8.37966 \times 10^{+03}$	5.75215×10^{-12}
F8	PSO	$0.00000 \times 10^{+00}$	4.68400×10^{-01}	7.58195×10^{-02}	4.20334×10^{-01}
	IPSO	$0.00000 \times 10^{+00}$	$0.00000 \times 10^{+00}$	$0.00000 \times 10^{+00}$	$0.00000 \times 10^{+00}$
	CSO	$0.00000 \times 10^{+00}$	$0.00000 \times 10^{+00}$	$0.00000 \times 10^{+00}$	$0.00000 \times 10^{+00}$
	ICSO	$0.00000 \times 10^{+00}$	$0.00000 \times 10^{+00}$	$0.00000 \times 10^{+00}$	$0.00000 \times 10^{+00}$
F9	PSO	2.89140×10^{-82}	1.49630×10^{-78}	3.58117×10^{-79}	1.41761×10^{-78}
	IPSO	1.27171×10^{-183}	5.89826×10^{-170}	5.89857×10^{-171}	$0.00000 \times 10^{+00}$
	CSO	2.07063×10^{-242}	1.57252×10^{-219}	1.57454×10^{-220}	$0.00000 \times 10^{+00}$
	ICSO	$0.00000 \times 10^{+00}$	$0.00000 \times 10^{+00}$	$0.00000 \times 10^{+00}$	$0.00000 \times 10^{+00}$
F10	PSO	3.99680×10^{-15}	$2.01332 \times 10^{+00}$	3.65954×10^{-01}	$2.32901 \times 10^{+00}$
	IPSO	7.54952×10^{-15}	7.54952×10^{-15}	7.54952×10^{-15}	$0.00000 \times 10^{+00}$
	CSO	4.44089×10^{-16}	4.44089×10^{-16}	4.44089×10^{-16}	$0.00000 \times 10^{+00}$
	ICSO	4.44089×10^{-16}	4.44089×10^{-16}	4.44089×10^{-16}	$0.00000 \times 10^{+00}$

表 4 改进算法和其他算法优化结果的对比($D=200$)

代号	算法	最好值	最差值	平均值	标准差
F1	PSO	6.14591×10^{-04}	1.22849×10^{-03}	8.964023×10^{-04}	5.82253×10^{-04}
	IPSO	7.81187×10^{-149}	3.27642×10^{-148}	$1.290115 \times 10^{-148}$	2.31206×10^{-148}
	CSO	$5.36973 \times 10^{+00}$	$6.89027 \times 10^{+01}$	$3.48801 \times 10^{+01}$	$6.95999 \times 10^{+01}$
	ICSO	2.56252×10^{-147}	9.41237×10^{-147}	5.62817×10^{-147}	5.67599×10^{-147}
F2	PSO	4.43149×10^{-01}	$1.90360 \times 10^{+00}$	9.65929×10^{-01}	$1.34884 \times 10^{+00}$
	IPSO	2.29014×10^{-139}	4.83038×10^{-130}	9.94938×10^{-131}	5.15431×10^{-130}
	CSO	$4.23800 \times 10^{+01}$	$1.54588 \times 10^{+03}$	$8.02359 \times 10^{+02}$	$1.38543 \times 10^{+03}$
	ICSO	7.21613×10^{-146}	4.05396×10^{-145}	1.50476×10^{-145}	3.18834×10^{-145}
F3	PSO	7.29997×10^{-03}	$7.16392 \times 10^{+01}$	$1.557740 \times 10^{+01}$	$8.32050 \times 10^{+01}$
	IPSO	2.10454×10^{-19}	8.70265×10^{-18}	2.14270×10^{-18}	7.85154×10^{-18}
	CSO	2.59556×10^{-03}	$1.93208 \times 10^{+01}$	$5.78862 \times 10^{+00}$	$2.18717 \times 10^{+01}$
	ICSO	3.03969×10^{-302}	3.11850×10^{-158}	3.11850×10^{-159}	2.95847×10^{-158}
F4	PSO	$1.56303 \times 10^{+00}$	$5.76579 \times 10^{+00}$	$3.10200 \times 10^{+00}$	$3.56208 \times 10^{+00}$
	IPSO	3.78916×10^{-134}	4.39822×10^{-123}	4.59393×10^{-124}	4.15599×10^{-123}
	CSO	$8.97761 \times 10^{+02}$	$2.84172 \times 10^{+04}$	$1.28287 \times 10^{+04}$	$3.06779 \times 10^{+04}$
	ICSO	2.60172×10^{-145}	9.06355×10^{-158}	1.87723×10^{-144}	8.04644×10^{-144}
F5	PSO	$9.805530 \times 10^{+01}$	$9.95016 \times 10^{+01}$	$9.84899 \times 10^{+01}$	$1.16060 \times 10^{+00}$
	IPSO	5.75800×10^{-03}	$4.27886 \times 10^{+00}$	1.47317×10^{-01}	4.52672×10^{-01}
	CSO	$9.85682 \times 10^{+01}$	$2.29066 \times 10^{+02}$	$1.19782 \times 10^{+02}$	$1.21685 \times 10^{+02}$
	ICSO	4.36000×10^{-04}	6.56167×10^{-01}	1.83297×10^{-01}	7.00266×10^{-01}
F6	PSO	$7.49929 \times 10^{+01}$	$1.55563 \times 10^{+02}$	$1.12685 \times 10^{+02}$	$7.22915 \times 10^{+01}$
	IPSO	$0.00000 \times 10^{+00}$	$0.00000 \times 10^{+00}$	$0.00000 \times 10^{+00}$	$0.00000 \times 10^{+00}$
	CSO	$0.00000 \times 10^{+00}$	$4.13874 \times 10^{+01}$	$1.22212 \times 10^{+01}$	$4.52215 \times 10^{+01}$
	ICSO	$0.00000 \times 10^{+00}$	$0.00000 \times 10^{+00}$	$0.00000 \times 10^{+00}$	$0.00000 \times 10^{+00}$

(续表)

代号	算法	最好值	最差值	平均值	标准差
F7	PSO	$-1.04218 \times 10^{+04}$	$-7.26957 \times 10^{+03}$	$-8.49226 \times 10^{+03}$	$2.84581 \times 10^{+03}$
	IPSO	$-8.37966 \times 10^{+04}$	$-8.37966 \times 10^{+04}$	$-8.37966 \times 10^{+04}$	4.60172×10^{-11}
	CSO	$-4.12207 \times 10^{+04}$	$-1.65352 \times 10^{+04}$	$-2.93928 \times 10^{+04}$	$2.39148 \times 10^{+04}$
	ICSO	$-8.37966 \times 10^{+04}$	$-8.37966 \times 10^{+04}$	$-8.37966 \times 10^{+04}$	4.60172×10^{-11}
F8	PSO	$1.97261 \times 10^{+02}$	$2.13152 \times 10^{+02}$	$2.06106 \times 10^{+02}$	$1.35447 \times 10^{+01}$
	IPSO	5.14700×10^{-05}	1.97405×10^{-02}	7.20296×10^{-03}	2.42216×10^{-02}
	CSO	8.28508×10^{-06}	9.56598×10^{-01}	3.08497×10^{-01}	9.44751×10^{-01}
	ICSO	$0.00000 \times 10^{+00}$	$0.00000 \times 10^{+00}$	$0.00000 \times 10^{+00}$	$0.00000 \times 10^{+00}$
F9	PSO	1.21361×10^{-54}	1.22666×10^{-51}	2.67346×10^{-52}	1.25341×10^{-51}
	IPSO	6.90112×10^{-107}	2.19348×10^{-91}	2.19348×10^{-92}	2.08092×10^{-91}
	CSO	3.75433×10^{-21}	8.84403×10^{-14}	1.75869×10^{-14}	9.20726×10^{-14}
	ICSO	7.72172×10^{-179}	4.10992×10^{-153}	4.10992×10^{-154}	3.89901×10^{-153}
F10	PSO	$3.70299 \times 10^{+00}$	$5.05133 \times 10^{+00}$	$4.34074 \times 10^{+00}$	$1.45303 \times 10^{+00}$
	IPSO	6.79456×10^{-14}	1.14131×10^{-13}	9.24593×10^{-14}	3.97046×10^{-14}
	CSO	$6.40962 \times 10^{+00}$	$1.05382 \times 10^{+01}$	$8.74266 \times 10^{+00}$	$3.82784 \times 10^{+00}$
	ICSO	2.88658×10^{-14}	3.95239×10^{-14}	3.20632×10^{-14}	9.33218×10^{-15}

由表 3 可知,20 维时,对于绝大部分测试函数来说,ICSO 和 IPSO 的优化结果都要远远好于 CSO 和 PSO 的优化结果,且对于大部分测试函数,ICSO 又好于 IPSO。表 3 验证了本文从小鸡更新公式、优化方式和基于认知多样性的变异三方面对算法的改进是有效的。

由表 4 可以看出,在 $D=200$ 时,除对于测试函数 F1,IPSO 略胜于 ICSO,以及对于测试函数 F6 和 F7,ICSO 与 IPSO 有相同的寻优结果外,ICSO 的求解结果都要好于其他算法的求解结果;尤其对于测试函数 F3,F8,F9,ICSO 远远好于 IPSO 的优化结果。表 4 验证了本文改进的鸡群算法解决更高维度复杂问题的能力。

4 种优化算法分别对 10 个 200 维的测试集函数独立运行 50 次,并求取平均值收敛,曲线如图 1—图 10 所示。

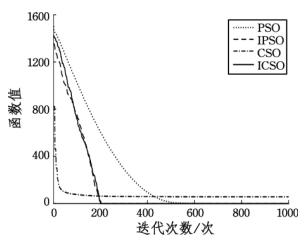


图 1 4 种算法在 $D=200$ 对 F1 函数的平均收敛曲线

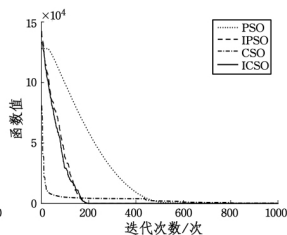


图 2 4 种算法在 $D=200$ 对 F2 函数的平均收敛曲线

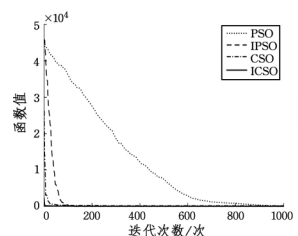


图 3 4 种算法在 $D=200$ 对 F3 函数的平均收敛曲线

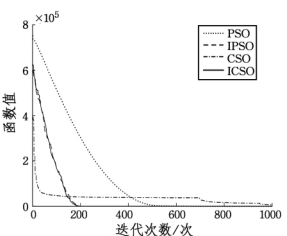


图 4 4 种算法在 $D=200$ 对 F4 函数的平均收敛曲线

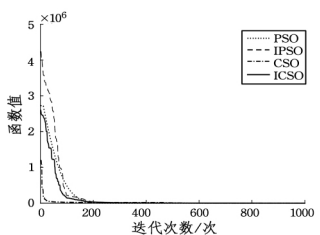


图 5 4 种算法在 $D=200$ 对 F5 函数的平均收敛曲线

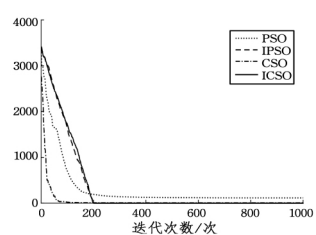


图 6 4 种算法在 $D=200$ 对 F6 函数的平均收敛曲线

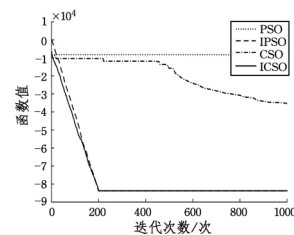


图 7 4 种算法在 $D=200$ 对 F7 函数的平均收敛曲线

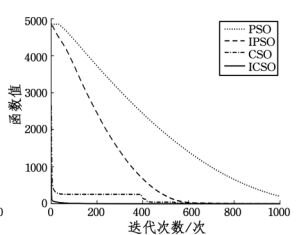


图 8 4 种算法在 $D=200$ 对 F8 函数的平均收敛曲线

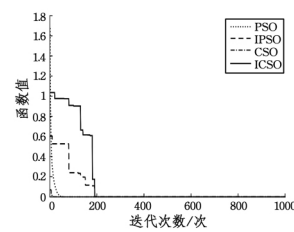


图 9 4 种算法在 $D=200$ 对 F9 函数的平均收敛曲线

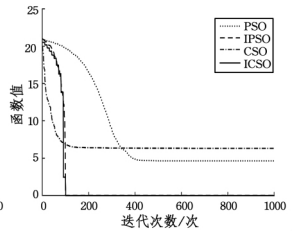


图 10 4 种算法在 $D=200$ 对 F10 函数的平均收敛曲线

分析收敛曲线,200 维时,就 10 个测试集函数而言,ICSO 的收敛精度均高于 IPSO,CSO 和 PSO,且远远高于 CSO 和 PSO。在收敛速度方面,因为不存在速度的概念,鸡群算法较粒子群算法更快收敛到了全局最优。同时,根据本文算法相关参数设置发现,ICSO 和 IPSO 算法最晚收敛到全局最优时为 200 次迭代,刚好是一轮协同优化结束,算法的高效性得到验证。

5 理论模型试算及分析

基于地震勘探褶积模型^[20](见图 11),设计一个模型地层(见图 12),选取 60Hz 的零相位雷克子波制作合成地震记录(见图 13),然后采用本文改进的鸡群算法进行波阻抗反演,最后用反演结果制作反演地震记录(见图 14)。计算得,反演前后地震记录能量相对误差不足 0.00002%,相关系数高达 100%。

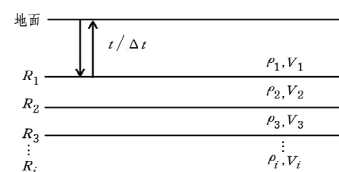


图 11 波阻抗与反射系数示意图

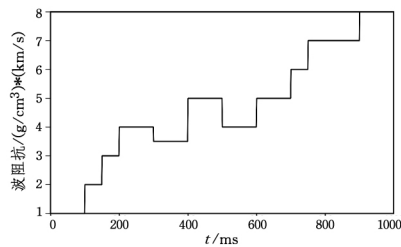


图 12 地层模型波阻抗剖面

为了检测本文算法对随机噪声的抗干扰能力^[21],在图 13 中加入 5%,10%,30% 的随机噪声,然后再进行反演,从而得到反演地震记录(见图 15)和波阻抗(见图 16)。反演前后的结果如表 5 所列。

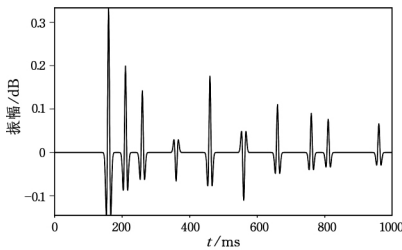


图 13 合成地震记录

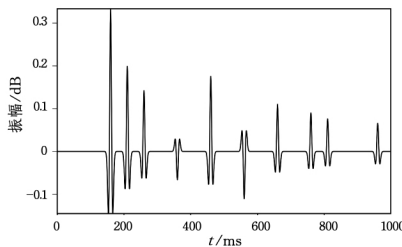


图 14 反演地震记录

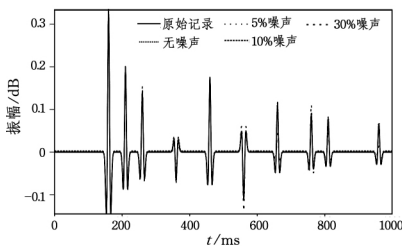


图 15 反演前后地震记录对比

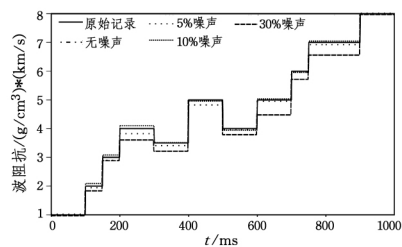


图 16 反演前后波阻抗对比

表 5 反演结果对比/%

噪声	反演前后能量相对误差	反演前后波形相关系数
无	0.00002	100.00000
5	0.15709	99.93610
10	0.24737	99.88040
30	4.37960	97.79080

从表 5 中可以看出,本文算法对随机噪声有很强的抗干

扰能力,特别是当噪声含量较少时,反演结果了相当令人满意;在加入 30% 的随机噪声后,反演结果误差也很小,这充分验证了该算法的实用性。

结束语 (1)从小鸡更新公式、优化方式和基于认知多样性变异三方面改进鸡群算法,避免在求解高维度复杂优化问题时出现早熟收敛和易陷入局部最优的问题。Benchmark function 测试表明,改进的鸡群算法比其他优化算法更易跳出局部最优,避免早熟收敛,尤其是处理高维问题时更容易找到全局最优解。

(2)一维模型反演结果表明,改进的鸡群算法具有很强的全局搜索能力,反演获得了较高精度,从而证实了该算法用于地震道反演的可行性,同时证明了改进的鸡群算法具有很强的抗噪能力。

参考文献

- [1] YANG X S. Bat algorithm: Literature review and applications [J]. International Journal of Bio-inspired Computation, 2013, 5 (3): 141-149.
- [2] DAS S, SUGANTHAN P N. Differential evolution: A survey of the state-of-the-art [J]. IEEE Transactions on Evolutionary Computation, 2011, 15(1): 4-31.
- [3] HOLLAND J H. Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence (2nd ed) [M]. Cambridge: MIT press, 1992.
- [4] KENNEDY J, EBERHART R. Particle swarm optimization [C] // Proceedings of IEEE International Conference on Neural Networks, Perth Australia: IEEE Press, 1995: 1942-1948.
- [5] JORDEHI A R, JASNI J. Parameter sselection in particle swarm optimization: A survey [J]. Journal of Experimental & Theoretical Artificial Intelligence, 2013, 25(4): 527-542.
- [6] DORIGO M, MANIEZZO V, COLORNI A. Ant system: optimization by a colony of cooperating agents [J]. IEEE Transactions on SystemsMan and Cybernetics, Part B: Cybernetics, 1996, 26 (1): 29-41.
- [7] LI X L, SHAO Z J, QIAN J X. An optimizing method based on autonomous animats: fish-swarm algorithm [J]. Systems Engineering Theory and Practice, 2002, 22(11): 32-38.
- [8] YANG X S. A new metaheuristic bat-inspired algorithm [C] // Nature Inspired Cooperative Strategies for Optimization (NIC-SO2010). Berlin Heidelberg: Springer, 2010.
- [9] KARABOGA D. An idea based on honey bee swarm for numerical optimization [R]. Kayseri: Erciyes University, 2005.
- [10] PASSINO K. Biomimicry of bacterial foraging fir distributed optimization and control [J]. IEEE Control Systems Magazine, 2002, 22(3): 52-57.
- [11] EUSUFF M M, LANSEY K E. Optimization of water distribution network design using the shuffled frog leaping algorithm [J]. Journal of Water Resources Planning and Management, 2003, 129(3): 201-225.
- [12] JORGE A, OCOTLAN D P, FELIPE C, et al. Meta-heuristics algorithms based on the grouping of animals by social behavior for the traveling salesman problem [J]. International Journal of Combinatorial Optimization Problems and Infirematics, 2012, 3 (3): 104-123.

- [13] GANDOMI A H, ALAVI A H. Krill herd: A new bio-inspired optimization algorithm. *Communications in Nonlinear Science and Numerical Simulation*, 2012, 17: 4831-4845.
- [14] CUEVAS E, CIENFUEGOS M, ZALDIVAR D, et al. A swarm optimization algorithm inspired in the behavior of the social-spider[J]. *Expert Systems with Applications*, 2013, 40: 6374-6384.
- [15] MENG X B, LIU Y, GAO X Z, et al. A new bio-inspired algorithm: chicken swarm optimization[C]//5th International Conference on Swarm Intelligence. Hefei: Springer International Publishing, 2014: 86-94.
- [16] 孔飞, 吴定会. 一种改进的鸡群算法[J]. *江南大学学报(自然科学版)*, 2015, 14(6): 681-688.

- [17] 王兴成, 胡汉梅, 刘林. 基于鸡群优化算法的配电网重构[J]. *电气工程*, 2016(3): 20-24.
- [18] 崔东文. 鸡群优化算法—投影寻踪洪旱灾害评估模型[J]. *水利水电科技进展*, 2016, 36(2): 16-23.
- [19] 洪杨, 于凤芹. 改进的鸡群算法并用于多分类器系数优化[J]. *计算机工程与应用*, 2017, 53(9): 158-161.
- [20] 易远元, 袁三一, 黄凯, 等. 地震波阻抗反演的粒子群算法实现[J]. *石油天然气学报*, 2007, 29(3): 79-81.
- [21] 成双全, 王尚旭, 季敏, 等. 地震波阻抗反演的蚁群算法实现[J]. *石油物探*, 2005, 44(6): 551-553.

(上接第 69 页)

协调靠操作者腰部动作完成。

3 控制算法的仿真验证

依据图 4 所示的流程图在 Matlab 上对末端随动控制算法进行仿真验证。首先, 设定人体下肢一侧末端运动轨迹为半圆, 其圆心坐标为 (5, -15), 半径为 5, 其曲线如图 5 所示。

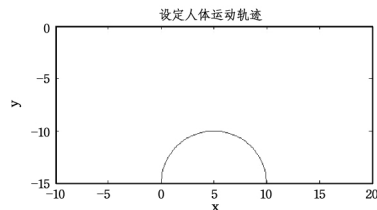


图 5 单侧人体下肢末端曲线

假设, 作用力 F 的大小与两点之间的距离成正比, 如式 (4) 所示。

$$\|F\| = KP \star \|P - P'\| \quad (4)$$

其中, KP 为作用系数; 作用力 F 的方向主要由机器人踝关节角度 γ' 确定, 而 γ' 可以用 P 和 P' 两点之间的斜率得到, 计算公式如式 (5) 所示。

$$\gamma' = \tan^{-1} K_L = \tan^{-1} \left(\frac{Y_{P'} - Y_P}{X_{P'} - X_P} \right) \quad (5)$$

其中, K_L 为两点直线的斜率。 $\|F\|$ 和 γ' 合成 F , 将其代入式 (2) 和式 (3) 中, 并设定人体的大腿和小腿长度均为 9; 令机器人大腿长度 $L_1 = 10$, 机器人小腿长度 $L_2 = 9$, $KP = 0.05$, $K = 12$, 最终得到下一时刻机器人髋关节和膝关节的角度值 θ_1' 和 θ_2' 。图 6 和图 7 为末端随动控制算法的 Matlab 仿真结果。

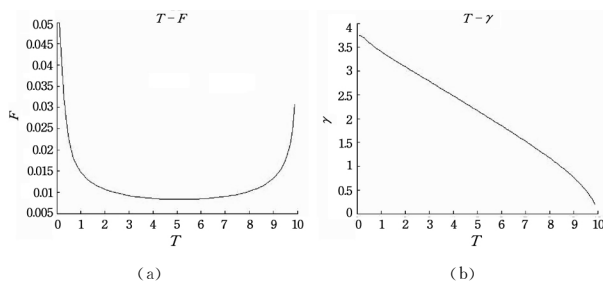


图 6 人体足底压力和机器人踝关节变化曲线

从图 6 可以看出, F 随时间 T 由大变小然后再增大, 在设

定半圆轨迹的起始点和终点处值偏大, 而在中间位置值最小, 符合实际情况; γ 随时间 T 由大逐渐减小直至为零, 其变化情况符合 γ 设定。

在图 7 中, 半圆为机器人单侧下肢末端模拟运动轨迹, 虚折线代表人体下肢, 实折线为机器人下肢, X 轴代表时间, Y 轴代表位置。从图中可以看出, 机器人下肢末端与人体下肢末端位置始终保持一致, 很好地实现了位置跟随。

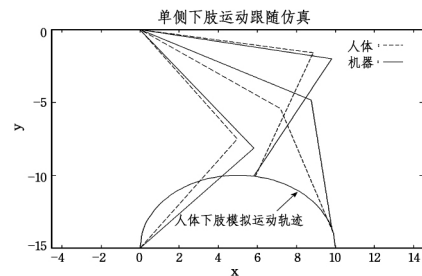


图 7 机器人下肢末端随动控制效果图

结束语 针对下肢外骨骼机器人的随动控制问题, 本文结合外骨骼机器人的特点, 提出了末端随动控制方法以及推导过程, 给出了具体的控制流程图。根据控制流程图, 结合 Matlab 软件, 给出了算法的仿真实验结果。从仿真实验结果可以看出: 机器人与人体下肢末端能实现良好随动, 适用于下肢外骨骼机器人的运动控制。

参 考 文 献

- [1] 蔡兆云, 肖湘江. 外骨骼机器人技术研究综述[J]. *国防科技*, 2007, 12: 6-8.
- [2] 张佳帆, 陈鹰, 杨灿军. 柔性外骨骼人机智能系统[M]. 北京: 科学出版社, 2011: 11-20.
- [3] 李会营, 王惠源, 张鹏军, 等. 外骨骼机器人发展趋势研究[J]. *机械工程师*, 2011(8): 9-10.
- [4] 曹恒, 孟宪伟, 凌正阳, 等. 两足外骨骼机器人足底压力测量系统[J]. *传感技术学报*, 2010(3): 326-330.
- [5] KUSHIDAD, NAKAMURAM, GOTOS, et al. Human direct teaching of industrial articulated robot arms based on forceless control[J]. *Artificial Life and Robotics*, 2001, 5(1): 26-32.
- [6] 胡建元, 黄心汉, 陈锦江. Movemaster-EX 机器人零力跟踪控制[J]. *华中理工大学学报*, 1993, 21(6): 135-139.
- [7] 吴威, 赵杰, 王晓东, 等. 焊接机器人在线示教[J]. *焊接学报*, 1995, 16(1): 16-19.