

文章编号: 1001-0920(2010)11-1619-06

基于 Lotka-Volterra 模型的双群协同竞争粒子群优化算法

吴宪祥¹, 郭宝龙¹, 王娟²

(1. 西安电子科技大学 机电工程学院, 西安 710071; 2. 空军工程大学 电讯工程学院, 西安 710077)

摘要: 针对粒子群优化算法易出现早熟收敛的问题, 提出了基于 Lotka-Volterra 模型的双群协同竞争粒子群优化算法(LVPSO). LVPSO 算法借鉴种群生态学中著名的 Lotka-Volterra 双群协同竞争模型, 讨论了两种种群协同竞争方案, 通过群内和群间竞争增加粒子的多样性, 提高了种群摆脱局部极值的能力. 对 5 个典型基准测试函数进行优化实验表明, LVPSO 在收敛速度和优化精度方面均有良好的表现.

关键词: 粒子群优化; 早熟收敛; Lotka-Volterra 模型; 多样性

中图分类号: TP18

文献标识码: A

Lotka-Volterra model based particle swarm optimization

WU Xian-xiang¹, GUO Bao-long¹, WANG Juan²

(1. School of Electromechanical Engineering, Xidian University, Xi'an 710071, China; 2. Telecommunication Engineering Institute, Air Force Engineering University, Xi'an 710077, China. Correspondent: WU Xian-xiang, E-mail: wuxianxiang@163.com)

Abstract: A particle swarm optimization algorithm inspired by Lotka-Volterra model(LVPSO) is proposed to avoid the premature convergence problem. The famous Lotka-Volterra model in ecology is introduced into basic particle swarm optimization algorithm, and two different cooperative-competitive schemes are discussed. The diversity of particles are increased by intraspecific and interspecific competition, and the ability of particles breaking away from the local extremum is improved remarkably. The experimental results show that the proposed LVPSO algorithm can converge in higher speed and higher precision by optimizing five typical benchmark functions.

Key words: Particle swarm optimization(PSO); Premature convergence problem; Lotka-Volterra model; Diversity

1 引言

受鸟群觅食行为的启发, Kennedy 和 Eberhart 于 1995 年提出了粒子群优化(PSO)算法^[1]. 该算法从诞生起, 就引起了国内外学者的广泛关注. 目前, PSO 算法已发展为群智能的一个重要分支, 大量改进型粒子群算法被提出并广泛地应用于科学研究、工程实践和管理等(如: 函数优化、神经网络训练、参数优化、路径规划、资源配置等)诸多领域. PSO 简单而又具有深刻的智能背景, 具有全局优化能力, 但对于多维复杂问题, 往往会出现早熟收敛现象, 容易陷入局部极优^[2-5].

粒子群算法的改进主要包括提高收敛速度和增加多样性, 而收敛性和多样性往往存在着矛盾. 惯性权重 w 和收缩因子 χ 的引入, 大大提高了标准粒子群

优化算法的收敛速度, 但在实际应用粒子群进行优化时, 往往效果不够理想, 其原因多在于种群多样性的丧失^[6-8]. 对于粒子群优化而言, 一方面, 接近收敛时的聚集恰恰意味着粒子趋同, 即多样性的丧失. 如果最优点位于拟聚集种群覆盖范围之内, 则重新寻优是有效的. 然而, 如果最优点明显远离粒子群, 粒子所具有的较低移动速度则会抑制寻优, 甚至粒子群会在某个伪最优处附近震荡^[9]. 另一方面, 粒子速度更新公式的后两项对种群当前最优粒子 g_{best} 没有任何贡献, 搜索中 g_{best} 总是沿着前一代速度方向飞行, 这导致 g_{best} 极易陷入局部极优. 以上两方面因素导致了标准粒子群优化算法容易早熟收敛.

为了平衡多样性丧失带来的影响而提出的解决方法可以分为三大类: 1) 在优化过程中通过随机化

收稿日期: 2009-09-20; 修回日期: 2009-12-25.

基金项目: 国家863计划项目(2006AA01Z127); 国家自然科学基金项目(60802077, 61003196); 中央高校基本科研业务费资助项目(K50510040007).

作者简介: 吴宪祥(1980-), 男, 山东临沂人, 讲师, 博士生, 从事计算机视觉、智能控制的研究; 郭宝龙(1962-), 男, 西安人, 教授, 博士生导师, 从事图像工程与智能控制等研究.

引入多样性^[10-14]; 2) 限定种群邻域拓扑结构以保持种群多样性, 控制算法的探测和开发能力^[15-17]; 3) 采用多个种群以实现在不同波峰上维持若干不同的粒子群. 人们期望这项技术能够对一类包含若干波峰的动态函数具有良好的性能, 已有证据表明, 这些动态函数可以很好地表现现实世界^[18]. 为了让每个子群能够跟踪自己所在的波峰, 文献[7]把多群的概念与量子粒子概念相结合, 以便维持群内的多样性; 文献[19]把生态选择中的 r -选择 和 K -选择的 概念引入粒子群, 以维持种群的多样性.

本文将生态学中著名的 Lotka-Volterra 双群协同竞争模型引入粒子群优化算法, 提出一种基于 Lotka-Volterra 模型的双群协同竞争粒子群优化算法 (LVPSO), 并讨论了两种种群协同竞争方案. LVPSO 属于第 3 类方法, 其核心思想来自于生态学中双群协同竞争的 Lotka-Volterra 模型. 通过两个子群的协同竞争增加种群粒子的多样性, 提高种群摆脱局部极值的能力, 克服单群的早熟收敛问题, 达到优化的目的. 该方法具有良好的生态学基础和进化论支撑. 对 5 个典型基准测试函数进行优化实验, 结果表明, LVPSO 在收敛速度和优化精度方面均有良好的表现.

2 Lotka-Volterra 模型

生态学理论认为, 生物个体在自身进化过程中受个体适应度、所处生存环境以及与其他个体之间的相互竞争等因素的影响. 在一定生态环境中的种群, 其种群进化不仅受到自身适应度的影响, 同时还受到环境和与其他种群相互之间的竞争协同的影响.

最著名的单种群动力学模型为逻辑斯蒂 (Logistic) 模型, 其动力学模型^[5]为

$$\frac{dN}{dt} = rN \left(1 - \frac{N}{K} \right). \quad (1)$$

其中: N 为任何时刻 t 种群个体的数量, r 为种群内禀增长率, K 为环境最大容纳量.

Logistic 模型仅仅关注了单个种群的数量变化情况. 事实上任何生物种群都是生物链中的一环, 存在着不可割裂的联系, 为此有必要考虑多个种群的情况. 用 $N_1(t)$ 和 $N_2(t)$ 分别表示两种群在 t 时刻的数量, 考虑到种群内自身发展规律和种群间相互作用的影响, 两种群相互作用的 Lotka-Volterra 模型常用的形式为

$$\begin{cases} \frac{dN_1}{dt} = r_1 N_1 \left(1 - \frac{N_1}{K_1} - \frac{a_{1,2} N_2}{K_1} \right), \\ \frac{dN_2}{dt} = r_2 N_2 \left(1 - \frac{N_2}{K_2} - \frac{a_{2,1} N_1}{K_2} \right). \end{cases} \quad (2)$$

其中: K_i 为不发生竞争的情况下种群 P_i 的环境负荷量; r_i 为种群 P_i 的内禀增长率; N_i 为种群 P_i 的个体数量; $a_{i,j}$ 为竞争系数, 反映了种群 P_j 对种群 P_i 的竞

争抑制作用; $i, j = 1, 2$.

根据种群间的相互作用, 一般分为以下 3 种情况:

1) 互利共生型: 即每一种群的存在都对对方有利, 对对方的增长起促进作用, 这时 $a_{1,2} \leq 0, a_{2,1} \leq 0$;

2) 捕食与被捕食型: 即种群 P_2 以种群 P_1 为食物来源 (或相反), 这时种群 P_1 的存在对种群 P_2 的数量增长有利, 而种群 P_2 的存在则对种群 P_1 不利, 此时 $a_{1,2} \geq 0, a_{2,1} \leq 0$;

3) 相互竞争型: 两个种群或者相互残杀, 或者竞争同一食物资源, 各自的存在对对方的增长都是不利的, 此时 $a_{1,2} \geq 0, a_{2,1} \geq 0$.

在粒子群优化中, 各种群竞争同一个“食物源”全局最优点, 因此各种群之间存在着相互竞争.

3 LVPSO 算法

生态学理论认为, 生存于一定自然环境资源约束中的种群, 通过相互之间的竞争协同, 互相驱使对方提高性能和复杂性, 从而实现种群之间的协同进化^[20]. 基于这一思想, 本文将 Lotka-Volterra 竞争模型引入粒子群优化算法, 提出了基于 Lotka-Volterra 模型的双群协同竞争粒子群优化算法 (LVPSO). 该算法充分考虑了环境和粒子群之间相互竞争的关系, 通过两个种群的 Lotka-Volterra 竞争方程提高粒子的多样性, 以克服单群的早熟收敛问题, 获得更高精度的结果.

3.1 LVPSO 的参数选择

设有种群 A 和种群 B , 其种群规模分别为 N_A 和 N_B . 在初始化时, 种群中的粒子随机产生. 种群 A 和种群 B 的速度和位置更新公式采用下面两种不同的方案引入群间竞争项.

3.1.1 方案 1 (LVPSO-1)

经过大量实验, 种群 A 和种群 B 均采用带收缩因子的粒子群优化算法. 在方案 1 (LVPSO-1) 中, 种群 A 和种群 B 的速度和位置更新公式如下:

$$\begin{aligned} v_i^A(t+1) = \chi^A(v_i^A(t) + c_1^A r_{1i}^A(p_i^A(t) - x_i^A(t)) + \\ c_2^A r_{2i}^A(p_g^A(t) - x_i^A(t)) + \\ c_3^A r_{3i}^A(p_g^B(t) - x_i^A(t))), \end{aligned} \quad (3)$$

$$x_i^A(t+1) = x_i^A(t) + v_i^A(t+1); \quad (4)$$

$$\begin{aligned} v_j^B(t+1) = \chi^B(v_j^B(t) + c_1^B r_{1j}^B(p_j^B(t) - x_j^B(t)) + \\ c_2^B r_{2j}^B(p_g^B(t) - x_j^B(t)) + \\ c_3^B r_{3j}^B(p_g^A(t) - x_j^B(t))), \end{aligned} \quad (5)$$

$$x_j^B(t+1) = x_j^B(t) + v_j^B(t+1). \quad (6)$$

其中: i 为种群 A 粒子序号, $i \in (1, 2, \dots, N_A)$; j 为种群 B 粒子序号, $j \in (1, 2, \dots, N_B)$; v_i^A 为种群 A 中第 i

个粒子的速度; v_j^B 为种群 B 中第 j 个粒子的速度; x_i^A 为种群 A 中第 i 个粒子的位置; x_j^B 为种群 B 中第 j 个粒子的位置; 种群 A 和种群 B 的收缩因子 $\chi^A = \chi^B = 0.729$; 种群 A 的加速因子 $c_1^A = 2.05$, $c_2^A = c_3^A = 1.025$; 种群 B 的加速因子 $c_1^B = 2.05$, $c_2^B = c_3^B = 1.025$; r_1, r_2 和 r_3 是区间 $[0,1]$ 之间的随机数; p_i^A 为种群 A 中第 i 个粒子的个人历史最优位置; p_j^B 为种群 B 中第 j 个粒子的个人历史最优位置.

在方案 1 中, 种群 A 和种群 B 共享全局最优信息

$$p_g(t) = \min(p_g^A(t), p_g^B(t)). \quad (7)$$

3.1.2 方案 2 (LVPSO-2)

种群 A 和种群 B 的速度和位置更新公式如下:

$$v_i^A(t+1) = \chi^A(v_i^A(t) + c_1^A r_{1i}^A(p_i^A(t) - x_i^A(t)) + c_2^A r_{2i}^A(p_g^A(t) - x_i^A(t)) + c_3^A r_{3i}^A(p_g^B(t) - x_i^A(t))), \quad (8)$$

$$x_i^A(t+1) = x_i^A(t) + v_i^A(t+1); \quad (9)$$

$$v_j^B(t+1) = \chi^B(v_j^B(t) + c_1^B r_{1j}^B(p_j^B(t) - x_j^B(t)) + c_2^B r_{2j}^B(p_g^B(t) - x_j^B(t)) + c_3^B r_{3j}^B(p_g^A(t) - x_j^B(t))), \quad (10)$$

$$x_j^B(t+1) = x_j^B(t) + v_j^B(t+1). \quad (11)$$

方案 2 与方案 1 的主要区别是速度更新公式中的最后一项. 在方案 1 中, 种群 A 和种群 B 每次进化时总有一个子群以标准收缩型 PSO 迭代, 而另一个子群粒子则受本子群全局最优和双群全局最优的共同吸引, 尽管增加了粒子多样性, 但仍无法有效避免全局最优粒子速度迅速衰减; 在方案 2 中, 种群 A 和种群 B 每次进化时各种群粒子均受本种群全局最优和另一种群全局最优的共同吸引, 通过两个随机数 r_2 与 r_3 的共同作用, 种群 A 和种群 B 的后代中一部分会偏向种群 g_{bestA} , 另一部分会偏向 g_{bestB} , 种群 A 和种群 B 之间有一个动态变化, 从而大大增加了粒子的多样性, 有效避免了粒子群早熟收敛问题.

3.2 LVPSO 的算法描述

LVPSO 的算法流程如下:

Step 1: 初始化两个种群. 设定两个种群的参数, 并为两个种群中的每个粒子随机赋予初始位置和初始速度.

Step 2: 评价粒子. 计算两个种群中每个粒子的适应度函数.

Step 3: 更新两个种群的粒子个人最好位置和全局最好位置. 确定两个种群中每个粒子个人历史最好位置 p_{bestA} 和 p_{bestB} , 确定两个种群迄今为止所经历的最好位置 g_{bestA} 和 g_{bestB} .

Step 4: 更新粒子. 根据粒子速度更新公式更新粒子飞行速度; 根据粒子位置更新公式更新粒子位置.

Step 5: 根据式 (7) 更新整个种群的全局最优位置.

Step 6: 检验终止条件. 如果当前进化次数达到预设的最大进化代数或优化结果达到最小误差, 则寻优结束, 输出最优解及最优值; 否则返回 Step 2, 继续进行搜索.

4 LVPSO 函数优化实验及分析

为了比较 LVPSO 的优化性能, 本文选择优化问题中最为典型的函数极值优化问题进行性能测试实验^[3,4]. 实验中采用的粒子群优化算法有权重线性递减粒子群优化算法 (LDWPSO) 和收缩型粒子群算法 (CFMPSO).

4.1 粒子群参数设置

表 1 给出了实验中涉及到的粒子群算法的参数设置. 最大迭代次数均为 1000. 为了对各种算法的性能进行定量比较, LVPSO 的两个种群规模均为其余算法种群规模的 1/2, 且两个子群的参数相同. 需要说明的是, 并非所有粒子群算法中都存在表中列出的参数, 对于未定义的参数在表中用“/”表示.

表 1 几种粒子群的参数设置

Items	LDWPSO	CFMPSO	LVPSO-1	LVPSO-2
T_{max}	1000	1000	1000	1000
Swarm size	40	40	20	20
c_1	2.05	2.05	2.05	2.05
c_2	2.05	2.05	1.025	1.025
c_3	/	/	1.025	1.025
χ	/	0.729	0.729	0.729
w_{start}	0.9	/	/	/
w_{end}	0.4	/	/	/

4.2 基准测试函数及其特性

在实验中, 选择 5 个著名的基准测试函数进行性能测试, 分别为 Sphere, Rosenbrock, Griewank, Rastrigin 和 Schaffer f6. 其中: Sphere, Rosenbrock 为单模态函数, 在给定的定义域内只有全局极小值点, 没有局部极小值点, 是评价进化算法全局收敛性的经典函数; Griewank, Rastrigin 和 Schaffer f6 为多模态函数, 在给定的定义域内有众多的局部极小值点, 难以优化, 常用来评价算法的搜索能力和开发性能.

表 2 给出了这些函数的表达式及其寻优空间. 其中: Schaffer f6 函数取 2 维, 其余函数维数均为 30. 除 Rosenbrock 的最优点为 $x^* = (1, 1, \dots, 1)^T$ 外, 其余最优点均为 $x^* = (0, 0, \dots, 0)^T$. 所有函数的全局极小值均为 $f(x^*) = 0$.

表 2 基准测试函数及其参数设置

函数名称	函数形式及搜索范围
Sphere	$f_1(x) = \sum_{i=1}^n x_i^2$ [-100, 100] ³⁰
Rosenbrock	$f_2(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$ [-30, 30] ³⁰
Griewank	$f_3(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$ [-600, 600] ³⁰
Rastrigin	$f_4(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$ [-5.12, 5.12] ³⁰
Schaffer f6	$f_5(x) = 0.5 + \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{(1 + 0.001(x_1^2 + x_2^2))^2}$ [-100, 100]

4.3 收敛精度分析

本文对几种优化算法和每个标准测试函数进行了 200 次优化实验, 每次进化代数 1 000, 种群规模 40. 各算法及函数的参数设置见表 1 和表 2. 在总共进行的 200 次优化中, 对某个函数的某一次优化, 定义误差为

$$E = |f(x) - f(x^*)|_{\min}. \quad (12)$$

其中: $f(x)$ 表示某个测试函数; x 表示优化过程中的位置向量; $f(x^*)$ 表示全局最小值, 对于所有基准测试函数, $f(x^*) = 0$. 可见, E 等于寻优过程中的函数值.

表 3 给出了 200 次优化最佳结果的统计数据. 其中相关指标的最小值用黑体表示. 在多次优化的情况下, 所有最佳优化结果 (即最小误差) 的平均值是评价

表 3 不同粒子群算法的函数极值优化实验误差结果比较

函数	f_1	f_2	f_3	f_4	f_5
LDWPSO					
最大值	9.8e-05	6.4e+02	0.108	85.567	0.0097
最小值	6.6e-08	2.491	2.8e-07	19.901	0
标准方差	1.2e-05	69.755	0.015	12.334	0.0025
平均值	7.6e-06	77.621	0.012	45.055	6.8e-04
CFMPSO					
最大值	0.0185	3.7e+02	0.975	111.435	0.0097
最小值	5.7e-08	5.145	9.1e-08	24.874	0
标准方差	1.3e-03	62.544	0.077	16.258	0.0035
平均值	1.5e-04	66.532	0.029	54.402	0.0015
LVPSO-1					
最大值	2.4e-08	165.589	0.135	99.495	0.0097
最小值	6.1e-13	0.060	3.5e-12	15.919	0
标准方差	2.1e-09	35.610	0.019	12.609	0.003
平均值	4.8e-10	44.534	0.016	40.196	1.0e-03
LVPSO-2					
最大值	4.64e-06	2.2e+02	0.044	93.754	0.0097
最小值	5.07e-09	14.515	5.6e-09	9.949	0
标准方差	5.35e-07	29.943	0.008	8.063	0.0014
平均值	3.41e-07	40.825	0.005	22.631	2.1e-04

优化算法性能的重要指标之一. 误差最小值和最大值也很重要, 但往往具有一定的偶然性.

依据表 3 所示的结果, 就平均误差而言, LVPSO-1 在 Sphere 函数中表现最优秀, LVPSO-2 在其余 4 个测试函数中是较优秀的; 就误差最大值而言, 除 LDWPSO 在 Rastrigin 函数中表现最好之外, 表现较好的是 LVPSO-1 或 LVPSO-2; 在误差最小值、标准方差等方面表现最佳的是 LVPSO-1 或 LVPSO-2. 总之, LVPSO-1 在单模态函数优化中性能表现更为突出; LVPSO-2 在多模态函数优化中性能表现最佳.

4.4 收敛速度分析

在函数极值优化中除了精度这一重要指标外, 还有一个重要指标就是收敛速度. 图 1~图 5 给出了基准测试函数在不同算法下 200 次优化的平均收敛曲线 (误差用以 10 为底的对数表示). 为了能较为完整地描述各算法的收敛速度, 实验中最大进化代数统一设为 1 000 代. 对于每个函数, 各种优化算法的误差区间设置相等.

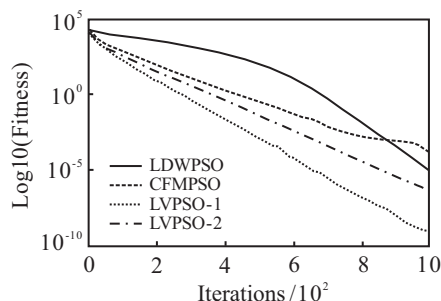


图 1 Sphere 函数的平均收敛曲线

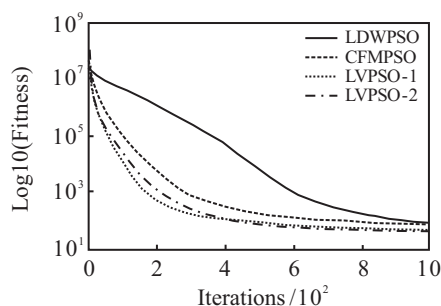


图 2 Rosenbrock 函数的平均收敛曲线

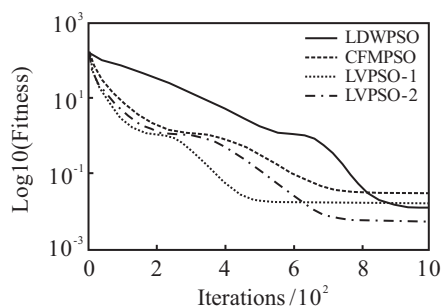


图 3 Griewank 函数的平均收敛曲线

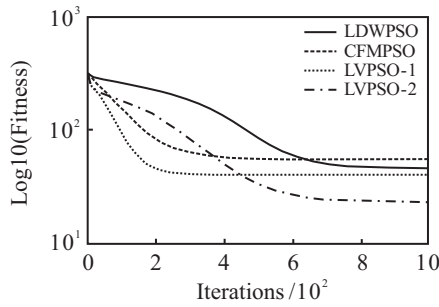


图 4 Rastrigrin 函数的平均收敛曲线

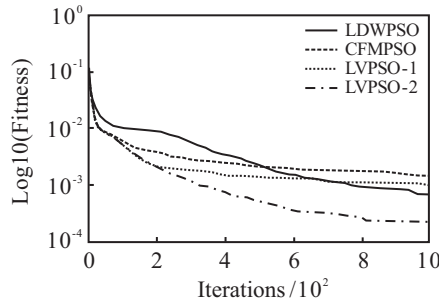


图 5 Schaffer f6 函数的平均收敛曲线

从图 1~图 5 可以看出, LVPSO-1 算法在收敛速度方面表现优异, 而 LVPSO-2 在收敛精度方面表现更为突出. 分析原因, LVPSO-2 粒子多样性的增加, 保证了算法可以获得更高的收敛精度, 但对收敛速度有一定的影响. 大量数据表明, LVPSO-2 的综合性能明显优于其余几种算法.

4.5 优化成功率分析

在实际优化应用中, 往往是达到设定的精度要求便可停止优化迭代过程, 而不必找到理论最优值. 本小节主要分析不同的优化算法在这类允许达到设定的误差阈值便可提前终止的情况下其性能表现.

实验中各函数的设定误差阈值见表 4. 为便于比较, 本文设定的误差阈值比一些文献里的数据偏小. 在优化过程中, 如果达到了该设定阈值, 则认为优化获得成功; 如果在给定的最大进化代数之内未能达到设定的误差阈值, 则判定优化失败. 表 4 还给出了进行 200 次优化过程中, 优化失败的次数. 其中: 每次进化代数 1000, 种群规模 40.

表 4 寻优失败次数

函数	f_1	f_2	f_3	f_4	f_5
LDWPSO	24	40	80	20	14
CFMPSO	72	29	107	65	31
LVPSO-1	0	10	97	11	23
LVPSO-2	0	5	37	1	7
误差阈值	1e-5	100	0.01	60	1e-5

5 结 论

本文分析了 PSO 的早熟收敛原因, 在此基础上, 结合种群生态学中著名的 Lotka-Volterra 模型, 提出

了基于 Lotka-Volterra 模型的双群协同竞争粒子群优化算法 (LVPSO), 讨论了两种种群协同竞争方案, 采用 5 个基准测试函数进行了仿真实验. 大量实验结果表明, 本文提出的 LVPSO 算法有效改善了 PSO 算法的性能. 综合收敛精度、收敛速度和优化成功率等各方面性能, LVPSO-2 算法性能更加均衡. LVPSO 算法通过群内和群间的竞争与协作, 有效增加了粒子的多样性, 对避免早熟收敛和陷入局部极值有积极作用.

参考文献(References)

- [1] Kennedy J, Eberhart R C. Particle swarm optimization[C]. Proc of IEEE Int Conf on Neural Networks. New York: IEEE Press, 1995: 1942-1948.
- [2] Shi Y, Eberhart R C. A modified particle swarm optimizer[C]. Proc of the IEEE Congress on Evolutionary Computation. Piscataway: IEEE Press, 1998: 69-73.
- [3] Eberhart R C, Shi Y. Comparing inertia weights and constriction factors in particle swarm optimization[C]. Proc of the Congress on Evolutionary Computation. San Diego: IEEE Press, 2000: 84-88.
- [4] 吴敏, 丁雷, 曹卫华, 等. 一种克服粒子群早熟的混合优化算法[J]. 控制与决策, 2008, 23(5): 511-514. (Wu M, Ding L, Cao W H, et al. A kind of hybrid optimization algorithm with prevention of premature convergence of particle swarm[J]. Control and Decision, 2008, 23(5): 511-514.)
- [5] 高鹰, 姚振坚, 谢胜利. 基于种群密度的粒子群优化算法[J]. 系统工程与电子技术, 2006, 28(6): 922-924. (Gao Y, Yao Z J, Xie S L. Particle swarm optimization algorithm based on population density[J]. Systems Engineering and Electronics, 2006, 28(6): 922-924.)
- [6] Carlisle A, Dozier G. Adapting particle swarm optimization to dynamic environments[C]. Proc of Int Conf on Artificial Intelligence. Las Vegas: IEEE Press, 2000: 429-434.
- [7] Blackwell T M. Particle swarms and population diversity[J]. Soft Computing, 2005, 9(11): 793-802.
- [8] Blackwell T M, Branke J. Multiswarms, exclusion, and anti-convergence in dynamic environments[J]. IEEE Trans on Evolutionary Computation, 2006, 10(4): 459-472.
- [9] Blackwell T M, Bentley P J. Dynamic search with charged swarms[C]. Proc of the Genetic and Evolutionary Computation Conf. New York: Morgan Kaufmann, 2002: 19-26.
- [10] Zhang L P, Yu H J, Hu S X. A new approach to improve particle swarm optimization[C]. Proc of the Genetic and Evolutionary Computation Conf. Chicago, 2003: 134-142.
- [11] 张丽平, 俞欢军, 陈德钊, 等. 粒子群优化算法的分析与改进[J]. 信息与控制, 2004, 33(5): 513-518.

- (Zhang L P, Yu H J, Chen D Z, et al. Analysis and improvement of particle swarm optimization algorithm[J]. Information and Control, 2004, 33(5): 513-518.)
- [12] Hu X H, Eberhart R C. Adaptive particle swarm optimization: Detection and response to dynamic systems[C]. Proc of IEEE Congress on Evolutionary Computation. Hawaii: IEEE Press, 2002: 1666-1670.
- [13] 吕振肃, 侯志荣. 自适应变异的粒子群优化算法[J]. 电子学报, 2004, 32(3): 416-420.
(Lv Z S, Hou Z R. Particle swarm optimization with adaptive mutation[J]. Acta Electronica Sinica, 2004, 32(3): 416-420.)
- [14] 高浩, 冷文浩, 须文波. 一种全局收敛的 PSO 算法及其收敛分析[J]. 控制与决策, 2009, 24(2): 196-201.
(Gao H, Leng W H, Xu W B. A global convergence algorithm of particle swarm optimization and its convergence analysis[J]. Control and Decision, 2009, 24(2): 196-201.)
- [15] Kennedy J. Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance[C]. Proc of IEEE Congress on Evolutionary Computation. Piscataway: IEEE Press, 1999: 1931-1938.
- [16] Kennedy J, Mendes R. Population structure and particle swarm performance[C]. Proc of the 2002 World Congress on Computational Intelligence. Hawaii: IEEE Press, 2002: 1671-1676.
- [17] Mendes R, Kennedy J, Neves J. The fully informed particle swarm: Simpler, maybe better[J]. IEEE Trans on Evolutionary Computation, 2004, 8(3): 204-210.
- [18] Branke J. Memory enhanced evolutionary algorithms for changing optimization problems[C]. Proc of Congress on Evolutionary Computation. Washington DC: IEEE Press, 1999, 3: 1875-1882.
- [19] Yan Y Y, Guo B L. Particle swarm optimization inspired by r - and K -selection in ecology[C]. Proc of 2008 IEEE Congress on Evolutionary Computation. Hong Kong: IEEE Press, 2008: 1117-1123.
- [20] 尚玉昌, 蔡晓明. 普通生态学[M]. 北京: 北京大学出版社, 1996.
(Shang Y C, Cai X M. General ecology[M]. Beijing: Beijing University Press, 1996.)

(上接第1618页)

- [10] 郭晨海, 谢俊, 刘军, 等. 连续非线性规划的猴王遗传算法[J]. 江苏大学学报, 2002, 23(4): 87-90.
(Guo C H, Xie J, Liu J, et al. Monkey-king genetic algorithms for continuous nonlinear programming Problems[J]. Jiangsu University of Science and Technology, 2002, 23(4): 87-90.)
- [11] 谢晓锋, 张文俊, 阮骏, 等. 针对带约束非线性规划问题的遗传算法[J]. 计算机工程与应用, 2002, 2(1): 64-67.
(Xie X F, Zhang W J, Ruan J, et al. Character recognition based on feature selection[J]. Computer Engineering and Applications, 2002, 2(1): 64-67.)
- [12] Liu Sheng-song, Hou Zhi-jian. Weighted gradient direction based chaos optimization algorithm for nonlinear programming problem[C]. Proc of the 4th World Congress on Intelligent Control and Automation. Shanghai: IEEE, 2002: 1779-1783.
- [13] 刘冀成, 胡雅毅. 带基因修复策略的自适应遗传算法[J]. 计算机应用, 2006, 26(6): 1041-1045.
(Liu J C, Hu Y Y. Adaptive genetic algorithm with missed gene detection and reparation policy[J]. Computer Applications, 2006, 26(6): 1041-1045.)
- [14] 田方, 谢里阳, 陶柯, 等. 基于惩罚和修复策略的约束优化遗传算法[J]. 机械设计, 2005, 22(11): 7-9.
(Tian F, Xie L Y, Tao K, et al. Constrained optimal genetic algorithm based on strategy of penalty and renovation[J]. Machine Design, 2005, 22(11): 7-9.)
- [15] 郭浩波, 王颖龙, 曾辉, 等. 采用遗传模拟退火算法研究导弹预警卫星传感器调度[J]. 光电与控制, 2006, 13(4): 71-74.
(Guo H B, Wang Y L, Zeng H. Sensor scheduling for missile early-warning satellite based on genetic and simulated annealing algorithm[J]. Electronics Optics and Control, 2006, 13(4): 71-74.)