

# 生态金字塔粒子群优化算法

刘亚红 张 玮 樊吕彬

(太原理工大学化学化工学院 太原 030024)

**摘要** 为解决粒子群优化算法在处理高维复杂函数时容易陷入局部最优和早熟收敛的问题,提出生态金字塔粒子群优化算法(EP-PSO)。该算法引入生态金字塔系统,使粒子在搜索空间分等级、分子群寻优,有效增加了群体多样性;为增强算法的全局搜索能力,对处于停滞状态的个体极值和全局极值进行动态变异,以达到扩大种群潜在搜索空间的效果。选择15个测试函数验证算法的有效性,结果表明EP-PSO有着良好的寻优性能,能够得到较高精度解,具有较高的效率和可信度。

**关键词** 粒子群优化算法,早熟收敛,生态金字塔系统

中图分类号 TP301.6

文献标识码 A

DOI 10.11896/j.issn.1002-137X.2017.10.043

## Ecological Pyramid Particle Swarm Optimization

LIU Ya-hong ZHANG Wei FAN Lv-bin

(College of Chemistry and Chemical Engineering, Taiyuan University of Technology, Taiyuan 030024, China)

**Abstract** A novel ecological pyramid particle swarm optimization variant was proposed to deal with the high dimensions, complex optimization problems. In the new variant, the ecological pyramid system was introduced to improve the particle's diversity. At the same time, the variation both on the local exemplar and the global exemplar was also employed extending the search space. To verify the effectiveness of the algorithm, fifteen benchmark problems were used to test the performance of EP-PSO. Experimental results validate the outstanding performance of EP-PSO. Compared with other algorithms, EP-PSO not only obtained high accuracy solutions, but also achieved high efficiency and reliability.

**Keywords** Particle swarm optimization algorithm, Premature convergence, Ecological pyramid system

## 1 引言

粒子群优化(Particle Swarm Optimization, PSO)算法最早是由 Eberhart 和 Kennedy 于 1995 年提出的群智能算法<sup>[1]</sup>,用于解决优化问题。在粒子群优化算法中,把鱼群或鸟群的栖息地或食物目标看成优化问题的可能解,利用群体中个体间的信息共享和合作机制,引导整个群体朝着可能解的位置运动,在该过程中逐渐找到更好的解。其中,原来的鸟群或鱼群被抽象为没有质量和体积的“粒子”,协作机制使每个“粒子”的速度受到自己和其他粒子历史状态的影响<sup>[1]</sup>。

PSO 算法由于结构简单且收敛速度快、易于编程实现等因素,得到研究者的广泛关注,在短短二十几年里已经发展成进化算法的一个重要分支,目前已经被广泛应用于函数优化、系统控制和阵列天线方向图优化<sup>[2-3]</sup>等多个领域。

为了提高传统粒子群优化算法中粒子的多样性和避免 PSO 算法早熟收敛,学者们主要在参数选择、学习策略改进、拓扑结构应用及与其他算法融合等方面提出改进。

对于 PSO 算法可调参数的研究,最多的是关于惯性权重  $w$  的取值问题。文献<sup>[4]</sup>首次引入惯性权重概念,为解决粒子群优化问题取得了突破性进展。目前,线性递减惯性权重由

于满足粒子在搜索初期全局开发能力强、搜索后期局部探索能力强的要求,被广泛应用。文献<sup>[5]</sup>提出 Control Strategy PSO(CSPSO),依据控制系统中的震荡特性提出了一种新的参数选择方法,以确定惯性权重  $w$  和加速因子  $c$  的值,从而调节粒子的局部探索和全局开发能力。

粒子群优化算法学习策略的改进能够有效改善粒子寻优性能。Liang 等人<sup>[6]</sup>提出了综合学习粒子群算法(CLPSO),在该算法中当前粒子可以向自己及其他粒子的任意一维学习,有效提高了粒子的搜索范围。Lim 等人<sup>[7]</sup>提出了教学和同行学习的粒子群优化算法(TPL-PSO),将搜索粒子分为两个不同的子群进行寻优,当粒子在教学组得不到有效改进时就以同行组最优粒子作为自己的全局向导,该算法提高了寻优粒子的多样性,具有良好的寻优精度和寻优效率。

粒子群算法是基于种群中粒子相互学习的进化算法。张英杰等人<sup>[8]</sup>提出了一种基于双子群的改进粒子群优化算法(TS-IPSO),通过两组搜索方向相反的主、辅子群之间的相互协同,扩大搜索范围。王维博等人<sup>[9]</sup>提出了基于分层多子群的混沌粒子群优化算法,有效避免了早熟收敛问题。赵新超等人<sup>[10]</sup>提出了一种基于非均匀变异和多阶段扰动的粒子群优化算法,较好地兼顾了群体优化算法的多样性和精英学习

到稿日期:2016-09-18 返修日期:2016-12-12 本文受山西省自然科学基金资助项目(2015011019)资助。

刘亚红(1990-),女,硕士生,主要研究方向为智能优化算法,E-mail:965607067@qq.com;张 玮(1973-),女,博士,副教授,主要研究方向为进化算法、生产过程监测,E-mail:zhangwei01@tyut.edu.cn(通信作者);樊吕彬(1992-),男,硕士生,主要研究方向为智能优化算法。

强度之间的平衡问题。

虽然学者们对粒子群优化算法提出了各种各样的改进并取得了较好效果,但高维复杂函数容易陷入局部最优、早熟以及精度低等问题,仍有待改进和提高。

为解决粒子群优化算法在处理高维复杂函数时容易陷入局部最优和早熟收敛的问题,本文提出生态金字塔粒子群优化算法(EP-PSO)。该算法引入生态金字塔系统,使粒子在搜索空间分等级、分子群寻优,有效提高了群体的多样性,增强了算法的全局搜索能力,对处于停滞状态的个体极值和全局极值进行动态变异,从而达到扩大种群潜在搜索空间的目的。

## 2 粒子群优化算法简介

粒子群优化算法是一种基于种群迭代模式的智能优化算法,每个粒子代表一个潜在可行解,整个群体在解空间中通过合作和竞争来搜寻最优解。在  $D$  维空间中,每个粒子有两个属性:当前位置  $X_i = (x_{i1}, x_{i2}, \dots, x_{iD})$  和当前速度  $V_i = (v_{i1}, v_{i2}, \dots, v_{iD})$ 。粒子  $X_i$  依据其历史最优值  $P_i = (p_{i1}, p_{i2}, \dots, p_{iD})$  和全局最优值  $P_g = (p_{g1}, p_{g2}, \dots, p_{gD})$  迭代更新,具体速度和位置的更新公式如下:

$$v_{id}(t+1) = \omega_i v_{id}(t) + c_1 r_1 (p_{id}(t) - x_{id}(t)) + c_2 r_2 (p_{gd}(t) - x_{id}(t)) \quad (1)$$

$$x_{i,d}(t+1) = x_{i,d}(t) + v_{i,d}(t+1) \quad (2)$$

其中,  $t$  为当前进化的迭代次数,  $\omega_i$  为惯性权重,  $c_1$  和  $c_2$  为两个正的加速度常数,  $r_1$  和  $r_2$  为  $[0, 1]$  之间的两个随机数,  $p_{i,d}(t)$  为第  $i$  个粒子的个体极值,  $p_{g,d}(t)$  为整个粒子群的群体极值。在迭代过程中,粒子适应度值  $f(x_i)$  决定粒子的性能<sup>[11]</sup>。

惯性权重  $\omega_i$  表示粒子惯性对速度的影响,其取值较大时全局寻优能力强,取值较小时局部寻优能力强。  $c_1 r_1 (p_{i,d}(t) - x_{i,d}(t))$  是自我认知部分,反映粒子本身记忆对寻优结果的影响,使粒子具有全局开发能力。  $c_2 r_2 (p_{g,d}(t) - x_{i,d}(t))$  是社会认知部分,体现了群体信息对粒子寻优的影响<sup>[11]</sup>。

## 3 生态金字塔粒子群优化算法

### 3.1 生态金字塔粒子群优化算法的思想

生态金字塔是由美国生态学家林德曼提出的一个营养循环模型<sup>[12]</sup>,将生态系统中各个营养级有机体的个体数量、生物量或能量,按营养级位顺序排列并绘制成图,因形似金字塔而被称为“生态金字塔”,表现出生态系统营养结构和能流过程所具有的成长性、层级性和递进性。

在生态金字塔中,每个营养级生物所获得的能量随着营养级的上升而递减,营养级数越低,该营养级所含生物个体数越多。本文受该思想启发提出生态金字塔粒子群优化算法(Ecological Pyramid Particle Swarm Optimization),该算法将生态系统金字塔中能量流的分布形式映射为在粒子群搜索空间中粒子函数适应度值的分布情况,针对寻找目标函数最优值的问题,在每次迭代的过程中依据函数适应度值,按一定比例将粒子分为不同的营养级(设生态系统金字塔营养级总数为  $N$ ,由第一营养级到最高营养级依次为  $1, 2, \dots, N$ ),又根据粒子适应度值将每个营养级分配为不同子群(营养级越低分配的子群数越多,每个子群中的粒子数相等,设每个营养级的子群个数为  $M$ ,依据适应度值分为第  $1, 2, \dots, M$  子群,每个

子群的粒子数为  $X$ )。

在迭代的过程中,每层营养级中每个子群的个体最优值由本子群决定,全局最优值由同营养级的前一子群的最优粒子决定,但该营养级中最优子群的全局最优值由上一营养级中最接近该子群信息的子群的最优粒子位置信息决定,该信息传递策略有效提高了信息流的传播效率,缩短了粒子寻优时间;除此之外,由于粒子所处营养级的等级不同,代表的信息水平也不同,营养级越高的子群,粒子的寻优能力越强,因此在迭代过程中只需考虑自身位置即可,大大简化了寻优过程。对于处于较低营养级的粒子,需借助自身速度、自我认知和社会认知这 3 部分来协调控制,随着迭代的进行,动态调整惯性权重来平衡 EP-PSO 粒子的全局搜索和局部搜索,详细内容将在后文介绍。

通过这种层级性、递进性和成长性的学习策略,有效提高了粒子的多样性,充分利用群体中所有粒子的有效信息,增强了粒子的寻优效率,加快了收敛速度。

通常生态系统不超过 5 个营养级,本次实验以 4 个营养级为例来进行,EP-PSO 算法过程示意图如图 1 所示。

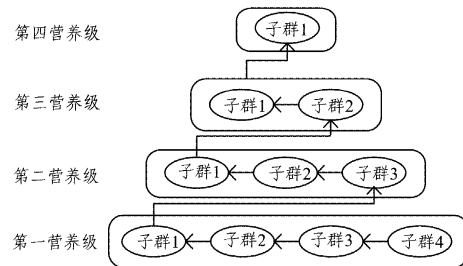


图 1 EP-PSO 算法过程示意图

### 3.2 生态金字塔粒子群优化算法原理

#### 3.2.1 生态金字塔粒子群优化算法提高收敛速度策略

从标准 PSO 模型角度来看,粒子位置  $x_i$  代表当前问题的解,优化的最终结果是使  $x_i$  无限逼近最优解  $X^*$ ,因此可以只考虑  $x_i$  的直接变化<sup>[13]</sup>。由于第四营养级和第三营养级子群中的粒子均具有较好的函数适应度值,因此为提高种群粒子的收敛速度,减小粒子速度对粒子偏离正确进化方向的影响,省去速度位移更新公式中的速度项,其优化方程简化为式(3):

$$x_{i,d}(t+1) = x_{i,d}(t) + c_1 r_1 (lpbest(t) - x_{i,d}(t)) + c_2 r_2 (xpbest(t) - x_{i,d}(t)) \quad (3)$$

其中,  $t$  为当前进化的迭代次数,  $c_1$  和  $c_2$  为两个正的加速度常数,  $r_1$  和  $r_2$  为  $[0, 1]$  之间的两个随机数,  $lpbest(t)$  为第  $i$  个粒子的个体最优值,  $xpbest(t)$  为该子群的群体最优值,两者的详细更新公式在下文说明。

#### 3.2.2 生态金字塔粒子群优化算法提高种群多样性策略

对于第二营养级和第一营养级子群中的粒子,在迭代更新的过程中主要侧重于搜索空间的探索 and 开发,有效增加种群多样性,避免粒子陷入早熟收敛。为此,惯性权重采用动态调整的策略<sup>[14]</sup>,如式(4)所示:

$$\omega_i = (\omega_{\max} - (\omega_{\max} - \omega_{\min}) * \frac{t}{T}) * rand(1) \quad (4)$$

其中,  $\omega_{\max}$  为惯性权重的最大值,  $\omega_{\min}$  为惯性权重的最小值,  $t$  为当前迭代次数,  $T$  为最大迭代次数,  $rand(1)$  为区间  $[0, 1]$  之间的随机数。随着迭代次数的变化,通过实验验证该动态调

整的惯性权重,有效平衡了算法的全局搜索和局部搜索。

取函数 Ackley 迭代过程中动态惯性权重值的变化情况进行说明,其他函数迭代过程中惯性权重值的变化趋势与之类似,惯性权重值变化曲线图如图 2 所示。

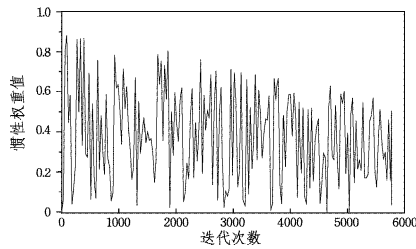


图 2 惯性权重变化曲线图

从图 2 可以看出,该算法采用的惯性权重呈整体下降的趋势,但并不表示后次迭代的惯性权重肯定比前次迭代的小,因此有效地平衡了算法的全局搜索和局部搜索,增加了算法种群的多样性。

在第二营养级和第一营养级子群中的粒子速度迭代公式如下:

$$v_{i,d}(t+1) = \omega_i v_{i,d}(t) + c_1 r_1 (lpbest(t) - x_{i,d}(t)) + c_2 r_2 (xpbest(t) - x_{i,d}(t)) \quad (5)$$

位移迭代公式仍按式(2)进行迭代更新。

### 3.2.3 生态金字塔粒子群优化算法变异策略

在每个子群中,对经过  $i$  次迭代、函数适应度值没有明显提升的粒子的个体极值和全局极值进行变异,如式(6) — 式(9)所示。

子群中个体极值的变异:

$$lpbest(j,:) = \begin{cases} pbestc1, & fit_{pbestc1} < fit_{pbest(j,:)} \\ pbest(j,:), & fit_{pbestc1} > fit_{pbest(j,:)} \end{cases} \quad (6)$$

$$pbestc1 = rand(1) * space(j,:) + posmaskmin(j,:) \quad (7)$$

其中,  $j$  为第  $j$  个粒子,  $rand(1)$  为  $[0, 1]$  区间的随机数,  $space(j,:)$  为第  $j$  个粒子的搜索空间范围,  $posmaskmin(j,:)$  为第  $j$  个粒子的搜索空间范围的最小值。

子群中全局极值的变异:

$$xpbest(j,:) = \begin{cases} ypbest(j,:), & fit_{ypbest(j,:)} < fit_{gbest_{n,m}} \\ gbest_{n,m}, & fit_{ypbest(j,:)} > fit_{gbest_{n,m}} \end{cases} \quad (8)$$

$$ypbest(j,:) = \begin{cases} gbest_{n,m} + randn(1) * space3 * space(j,:), & rand(1) > 0.5 \\ gbest_{n,m} - randn(1) * space3 * space(j,:), & rand(1) < 0.5 \end{cases} \quad (9)$$

其中,  $j$  为第  $j$  个粒子,  $rand(1)$  为  $[0, 1]$  区间的随机数,  $randn(1)$  为根据正态分布概率生成的随机数,  $space(j,:)$  为第  $j$  个粒子的搜索空间范围,  $gbest_{n,m}$  为第  $n$  个营养级第  $m$  个子群中的全局最优值,  $space3$  为取搜索空间范围中的任意两维。该变异机制侧重强调基于更新粒子在本身对应空间维度和子群寻优范围的变异。

### 3.3 生态金字塔粒子群优化算法流程

算法 1 给出了生态金字塔粒子群优化算法主程序部分的伪代码描述。

### 算法 1 EP-PSO 算法

1. For  $i=1:me \%me$ ;最大迭代次数
2. For  $j=1:ps \%ps$ ;最大粒子个数
3. 按粒子适应度值进行升序排列,将群体分为  $N$  个营养级,每个营养级由  $M$  个子群组成;
4. If 粒子  $j$  在第一、二营养级
5. 按式(2)和式(5)进行位置和速度更新;
6. elseif 粒子  $j$  在第三、四营养级
7. 按式(3)进行位置更新;
8. If 经过  $Z$  次迭代,子群最优值变化  $\Delta gbest_{vall}(j,:) < \delta$
9. 按式(6)和式(7)对个体极值进行变异操作;
10. 按式(8)和式(9)对全局最优值进行变异操作;
11. else 返回步骤 4 或步骤 6
12. end
13. end
14. 判断是否满足终止条件;
15. end
16. end

## 4 生态金字塔粒子群优化算法性能测试

### 4.1 测试函数

选取单峰函数、多峰函数、偏移函数和旋转函数等<sup>[15]</sup>常用经典函数进行实验,以测试算法 EP-PSO 的性能。测试函数的名称、表达式及寻优范围等信息如表 1 和表 2 所列。

表 1 测试函数名称及表达式

函数名	函数表达式
Sphere	$F_1(X_i) = -\exp(-0.5 \sum_{d=1}^D X_{i,d}^2)$
Schwefel2.22	$F_2(X_i) = \sum_{d=1}^D  X_{i,d}  + \prod_{d=1}^D  X_{i,d} $
Schwefel1.2	$F_3(X_i) = \sum_{d=1}^D (\sum_{j=1}^d X_{i,j})^2$
Schwefel2.21	$F_4(X_i) = \max_{d=1,\dots,D}  X_{i,d} $
Rosenbrock	$F_5(X_i) = \sum_{d=1}^{D-1} (100(X_{i,d}^2 - X_{i,d+1})^2 + (X_{i,d} - 1)^2)$
Rastrigin	$F_6(X_i) = \sum_{d=1}^D (X_{i,d}^2 - 10 \cos(2\pi X_{i,d}) + 10)$
Non-continuous Rastrigin	$F_7(X_i) = \sum_{d=1}^D (Y_{i,d}^2 - 10 \cos(2\pi Y_{i,d}) + 10)$ where $Y_{i,d} = \begin{cases} X_{i,d}, &  X_{i,d}  < 5 \\ \text{round}(2X_{i,d})/2, &  X_{i,d}  \geq 5 \end{cases}$
Griewank	$F_8(X_i) = \sum_{d=1}^D X_{i,d}^2 / 4000 - \prod_{d=1}^D \cos(X_{i,d} / \sqrt{d}) + 1$
Ackley	$F_9(X_i) = -20 \exp(-0.2 \sqrt{\frac{\sum_{d=1}^D X_{i,d}^2}{D}}) - \exp(\frac{\sum_{d=1}^D \cos(2\pi X_{i,d})}{D} + 20 + e)$
Weierstrass	$F_{10}(X_i) = \sum_{d=1}^D [\sum_{k=0}^{k_{max}} [a^k \cos(2\pi b^k (X_{i,d} + 0.5))] - D \sum_{k=0}^{k_{max}} [a^k \cos(\pi b^k)]]$ $a=0.5, b=3, k_{max}=20$
Rotated Rastrigin	$F_{11}(X_i) = F_6(Z_i), Z_i = M \times X_i$
Rotated Non-continuous Rastrigin	$F_{12}(X_i) = F_7(Z_i), Z_i = M \times X_i$
Shifted Rastrigin	$F_{13}(Z_i) = F_6(Z_i) + f_{bias1}, Z_i = X_i - o, f_{bias1} = -330$
Shifted Non-continuous Rastrigin	$F_{14}(Z_i) = F_7(Z_i) + f_{bias2}, Z_i = X_i - o, f_{bias2} = -330$
Shifted Griewank	$F_{15}(Z_i) = F_8(Z_i) + f_{bias3}, Z_i = X_i - o, f_{bias3} = -180$

表2 测试函数序号、搜索空间及目标精度等信息

序号	定义域	极值	目标精度	单峰/多峰
F1	$[-100,100]^D$	0	1.00E-06	单峰
F2	$[-10,10]^D$	0	1.00E-06	单峰
F3	$[-100,100]^D$	0	1.00E-06	单峰
F4	$[-2.048,2.048]^D$	0	1.00E-06	单峰
F5	$[-2.048,2.048]^D$	0	1.00E-02	多峰
F6	$[-5.12,5.12]^D$	0	1.00E-02	多峰
F7	$[-5.12,5.12]^D$	0	1.00E-02	多峰
F8	$[-600,600]^D$	0	1.00E-02	多峰
F9	$[-32,32]^D$	0	1.00E-02	多峰
F10	$[-0.5,0.5]^D$	0	1.00E-02	多峰
F11	$[-5.12,5.12]^D$	0	1.00E-02	多峰
F12	$[-5.12,5.12]^D$	0	1.00E-02	多峰
F13	$[-5.12,5.12]^D$	$f_{bias1}=-330$	1.00E-02	多峰
F14	$[-5.12,5.12]^D$	$f_{bias2}=-330$	1.00E-02	多峰
F15	$[-600,600]^D$	$f_{bias3}=-180$	1.00E-02	多峰

表1中, Sphere 函数为单模态二次函数, 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; Schwefel 系列函数的全局最优与最好的局部最优相距很远, 带有一定的欺骗性; Rosenbrock 函数是难极小化的非凸病态函数, 随着维数升高, 最优解分布复杂度逐渐增强, 能有效考察算法的执行效率; Griwank, Rastrigin, Weierstrass 函数是多模态函数, 在给定的定义域内有众多的局部极小值点, 难以优化, 常用来考察算法跳出局部最优解的能力, 评价算法的搜索能力和开发性能; Ackley 函数在低维数情况下局部最优点已较多, 随着维数的增高, 局部最优点的数量迅速增多, 增加了最优解的寻找难度。

### 4.2 算法性能测试及分析

为说明所提算法的有效性, 选取 15 个基准函数进行测试, 将 EP-PSO 算法与 9 种 PSO 典型改进算法 PSO-LDIW<sup>[4]</sup>, CLPSO<sup>[16]</sup>, APSO<sup>[17]</sup>, CPSO<sup>[18]</sup>, FLPSO-QIW<sup>[19]</sup>, FPSO<sup>[20]</sup>, FIPSO<sup>[21]</sup>, MP-PSO-TVAC<sup>[22]</sup>, RPPSO<sup>[23]</sup> 进行比较, 各算法的参数设置如表 3 所列。仿真中, 算法用 Matlab7.0 编程工具实现, 所有算法针对每个函数独立运算 30 次, 每个函数的测试维数为 50, 种群大小为 30, 每次算法运行最多计算 3.00E+05 次适应度函数。

在算法 EP-PSO 的实验中, 设置最高营养级级数为 4, 第一营养级至第四营养级每级所分配的子群个数依次为 4, 3, 2, 1, 每个子群中均有 3 个寻优粒子。为平衡粒子的局部探索

能力和全局开发能力, 惯性权重采用动态调整策略进行变化, 最大值设置为 0.9, 最小值为 0.4, 具体变化趋势如 3.2.2 节所述。

表3 各个算法的参数设置

算法名称	参数设置
APSO	$w: 0.9 \sim 0.4, c_1 + c_2: [3.0, 4.0], \delta = [0.05, 0.1], \sigma_{max} = 1.0, \sigma_{min} = 0.1$
CLPSO	$w: 0.9 \sim 0.4, c = 2.0, m = 7$
CPSO	$\chi = 0.729, c_1 = c_2 = 1.49445$
FLPSO-QIW	$w: 0.9 \sim 0.2, c_1: 2 \sim 1.5, c_2: 1 \sim 1.5, m = 1, P_i = [0.1, 1], K_1 = 0.1, K_2 = 0.001, \sigma_1 = 1, \sigma_2 = 0$
FPSO	$w: 0.9 \sim 0.4, \sum c_i = 4.1$
FIPSO	$\chi = 0.729, \sum c_i = 4.1$
MP-PSO-TVAC	$w: 0.9 \sim 0.4, c_1: 2.5 - 0.5, c_2: 0.5 - 2.5$
RPPSO	$w: 0.9 \sim 0.4, c_{large} = 6, c_{small} = 3$
PSO-LDIW	$w: 0.9 \sim 0.4, c_1 = c_2 = 2.0$
EPPSO	$w: 0.9 \sim 0.4, c_1 = 1.85, c_2 = 2.0$

算法性能评价指标为: 1) 优化精度, 固定函数优化迭代次数(3.00E+05)作为所有算法的终止条件, 每个测试函数与算法独立运行 30 次, 采用最佳目标函数值的平均精度和标准差作为评价准则; 2) 鲁棒性, 通过统计算法达到规定目标精度阈值的实验次数占总实验次数的比例(成功率)进行评价, 评估粒子寻优的稳定性; 3) 收敛速度, 算法在达到相同的函数迭代次数要求时所需的平均运行时间和适应度值迭代优化曲线, 衡量粒子寻优的时效性。

#### 1) 优化精度分析

各算法的平均精度、标准差实验统计结果如表 4 所列。

由表 4 中各算法的平均精度值可知, 在 4 个单峰函数 F1, F2, F3, F4 中, EP-PSO 算法的测试平均值都优于其他 9 种算法, 达到了极值 0。对于很多改进粒子群算法难以优化的多峰函数 F6, F7, F9, F10, F12, 算法 EP-PSO 均成功跳出局部最优, 找到最优适应度值 0。旋转函数 F11, F12, EP-PSO 的平均精度在 10 种改进算法中的 Rank 排名为 1, 偏移函数 F15 的平均精度同样达到了目标精度要求。由此可知, 提出的 EP-PSO 算法的优化精度总体要优于其他 9 种改进算法。可见, 生态金字塔系统的引入使得算法能够保持良好的种群多样性, 从而使寻优粒子不容易陷入局部最优, 得到的结果更加接近真实的最优值, 因此拥有较好的寻优精度。

表4 各算法的平均精度及标准差的对比结果

函数		APSO	CLPSO	CPSO	FLPSO-QIW	FPSO	FIPSO	MP-PSO-TVAC	RPPSO	PSO-LDIW	EPPSO
F1	$F_{mean}$	2.50E-01	3.29E-47	3.43E+03	2.90E-81	7.02E+01	2.96E-01	<b>0.00E+00</b>	1.28E-02	4.67E+03	<b>0.00E+00</b>
	$SD$	1.81E-01	1.28E-46	2.80E+03	5.97E-81	6.98E+01	8.06E-01	0.00E+00	2.98E-02	7.30E+03	0.00E+00
	$Rank$	5	3	8	2	7	6	1	4	9	1
F2	$F_{mean}$	7.51E-02	2.95E-29	1.70E+01	3.98E-57	2.83E+00	9.05E-02	7.94E-10	2.20E+01	2.90E+01	<b>0.00E+00</b>
	$SD$	3.39E-02	2.16E-29	1.37E+01	1.01E-56	1.81E+00	1.54E-01	1.59E-09	1.60E+01	1.75E+01	0.00E+00
	$Rank$	5	3	8	2	7	6	4	9	10	1
F3	$F_{mean}$	1.46E+03	5.13E+03	2.82E+04	2.63E+02	3.44E+03	8.13E+00	2.54E-02	9.12E+01	2.08E+04	<b>0.00E+00</b>
	$SD$	4.82E+02	1.00E+03	1.01E+04	8.90E+01	1.33E+03	2.47E+01	2.83E-02	4.21E+01	1.59E+04	0.00E+00
	$Rank$	6	8	10	5	7	3	2	4	9	1
F4	$F_{mean}$	1.75E+01	4.57E-01	2.82E+01	5.95E+00	5.00E+00	4.97E-01	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>
	$SD$	3.85E+00	2.10E-01	4.40E+00	2.25E+00	3.48E+00	7.87E-01	0.00E+00	0.00E+00	0.00E+00	0.00E+00
	$Rank$	7	3	8	6	5	4	1	1	1	1
F5	$F_{mean}$	4.62E+01	4.35E+01	2.33E+02	<b>4.21E+01</b>	5.68E+01	4.77E+01	4.34E+01	4.76E+01	2.10E+02	4.28E+01
	$SD$	1.53E+00	1.83E-01	1.61E+02	2.40E-01	7.08E+00	8.44E-01	5.10E-01	4.30E-01	4.34E+02	1.68E-01
	$Rank$	5	4	10	1	8	7	3	6	9	2

(续表)

函数		APSO	CLPSO	CPSO	FLPSO-QIW	FPSO	FIPSO	MPSO-TVAC	RPPSO	PSO-LDIW	EPPSO
F6	$F_{mean}$	5.81E-01	9.10E+01	2.84E+02	2.60E+00	1.85E+01	1.57E+00	3.02E-15	9.25E+00	1.15E+02	<b>0.00E+00</b>
	SD	6.29E-01	1.08E+01	4.22E+01	1.52E+00	1.02E+01	3.71E+00	6.47E-15	1.55E+01	7.78E+01	0.00E+00
	Rank	3	8	10	5	7	4	2	6	9	1
F7	$F_{mean}$	3.60E-02	8.10E+01	2.95E+02	5.58E+00	1.60E+01	5.70E-01	2.61E-15	1.25E+01	1.14E+02	<b>0.00E+00</b>
	SD	3.22E-02	9.76E+00	4.75E+01	2.36E+00	9.56E+00	8.65E-01	3.72E-15	1.94E+01	5.81E+01	0.00E+00
	Rank	3	8	11	5	7	4	2	6	9	1
F8	$F_{mean}$	1.70E-01	3.39E-11	3.78E+01	5.75E-04	1.86E+00	1.93E-01	<b>0.00E+00</b>	7.10E-03	3.92E+01	9.27E-06
	SD	8.21E-02	1.73E-10	3.78E+01	2.20E-03	9.28E-01	3.47E-01	0.00E+00	1.85E-02	7.00E+01	1.10E-02
	Rank	6	2	9	4	8	7	1	5	10	3
F9	$F_{mean}$	6.60E-02	1.15E-14	1.45E+01	3.43E-14	1.80E+00	1.70E-01	0.00E+00	7.47E-01	1.21E+01	-8.88E-16
	SD	2.57E-02	2.59E-15	1.51E+00	1.07E-14	1.10E+00	3.38E-01	0.00E+00	9.17E-01	5.99E+00	0.00E+00
	Rank	5	3	10	4	8	6	2	7	9	1
F10	$F_{mean}$	5.44E-01	<b>0.00E+00</b>	4.83E+01	1.88E-05	3.35E+00	9.80E-01	1.50E-01	4.69E-01	8.09E+00	<b>0.00E+00</b>
	SD	1.88E-01	0.00E+00	4.94E+00	8.29E-05	2.35E+00	9.54E-01	4.58E-01	1.25E+00	5.90E+00	0.00E+00
	Rank	5	1	9	2	7	6	3	4	8	1
F11	$F_{mean}$	1.83E+02	3.33E+02	3.51E+02	1.26E+02	1.80E+02	2.65E+01	7.95E+01	4.25E+01	1.70E+02	2.30E+01
	SD	5.61E+01	2.34E+01	4.07E+01	1.76E+01	5.01E+01	3.39E+01	5.80E+01	4.64E+01	7.41E+01	3.99E+00
	Rank	8	9	10	5	7	2	4	3	6	1
F12	$F_{mean}$	2.59E+02	3.21E+02	3.63E+02	1.28E+02	1.53E+02	4.15E+01	1.13E+02	8.00E+01	2.00E+02	<b>0.00E+00</b>
	SD	6.15E+01	2.52E+01	8.17E+01	2.13E+01	3.74E+01	5.13E+01	7.17E+01	5.30E+01	7.58E+01	0.00E+00
	Rank	8	9	10	4	6	2	5	3	7	1
F13	$F_{mean}$	5.91E-01	6.85E+01	5.16E+02	5.88E+00	2.08E+02	1.31E+02	2.99E-01	1.62E+02	2.93E+02	1.37E-01
	SD	7.76E-01	1.01E+01	8.59E+01	2.51E+00	4.59E+01	2.93E+01	4.64E-01	4.08E+01	5.76E+01	3.08E+01
	Rank	3	5	10	4	8	6	2	7	9	1
F14	$F_{mean}$	<b>7.20E-03</b>	6.99E+01	5.28E+02	1.20E+01	1.63E+02	1.49E+02	3.01E-01	2.09E+02	3.14E+02	2.43E+01
	SD	1.06E-02	7.32E+00	8.47E+01	3.16E+00	2.82E+01	3.94E+01	4.67E-01	5.21E+01	6.18E+01	3.70E+00
	Rank	1	5	10	3	6	7	2	8	9	4
F15	$F_{mean}$	<b>0.00E+00</b>	4.24E-09	7.69E+02	2.05E-03	1.46E+03	<b>0.00E+00</b>	<b>0.00E+00</b>	<b>0.00E+00</b>	7.21E+02	3.56E-03
	SD	0.00E+00	1.35E-08	4.09E+02	3.49E-03	4.63E+02	0.00E+00	0.00E+00	0.00E+00	4.54E+02	7.23E-03
	Rank	1	2	5	3	7	1	1	1	6	4

注:平均精度( $F_{mean}$ )为算法在经过 3.00E+5 次迭代运算后得到的最佳目标函数值的平均收敛精度;标准差(SD)为 3.00E+5 次迭代后的适应度值的标准差;Rank 为各个算法依据平均精度的排名

2)鲁棒性分析

各算法的成功率和达到目标精度的适应度函数平均迭代次数的实验统计结果如表 5 所列。从表 5 统计的信息可知,算法 EP-PSO 在函数 F1, F2, F3, F4, F6, F7, F9, F10, F12 中

都以 100%的概率达到了函数目标精度要求,所用的平均函数计算次数也最少。对于难极小化的非凸病态函数 F5, 10 种改进算法都未能成功找到最优适应度值。EP-PSO 算法在函数 F8, F15 中的优化概率分别为 60%和 90%。

表 5 各算法的成功率和适应度函数平均迭代次数

函数		APSO	CLPSO	CPSO	FLPSO-QIW	FPSO	FIPSO	MPSO-TVAC	RPPSO	PSO-LDIW	EPPSO
F1	SR	0.00%	<b>100.00%</b>	0.00%	<b>100.00%</b>	13.33%	80.00%	<b>100.00%</b>	73.33%	63.33%	<b>100.00%</b>
	SP	Inf	1.25E+05	Inf	6.04E+05	9.68E+04	9.86E+04	4.49E+03	1.52E+04	1.63E+04	<b>3.48E+03</b>
F2	SR	0.00%	<b>100.00%</b>	0.00%	<b>100.00%</b>	3.33%	50.00%	<b>100.00%</b>	10.00%	0.00%	<b>100.00%</b>
	SP	Inf	1.41E+05	Inf	7.11E+04	3.44E+05	1.65E+05	1.02E+05	3.11E+05	Inf	<b>4.92E+03</b>
F3	SR	0.00%	0.00%	0.00%	0.00%	0.00%	70.00%	100.00%	0.00%	0.00%	<b>100.00%</b>
	SP	Inf	Inf	Inf	Inf	Inf	1.62E+05	2.66E+05	Inf	Inf	<b>3.39E+03</b>
F4	SR	0.00%	0.00%	0.00%	0.00%	0.00%	60.00%	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>	<b>100.00%</b>
	SP	Inf	Inf	Inf	Inf	Inf	1.39E+05	5.78E+03	1.17E+04	1.18E+04	<b>3.39E+03</b>
F5	SR	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%
	SP	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf	Inf
F6	SR	0.00%	0.00%	0.00%	6.67%	0.00%	40.00%	100.00%	70.00%	3.33%	<b>100.00%</b>
	SP	Inf	Inf	Inf	3.46E+06	Inf	3.56E+05	8.57E+04	4.19E+04	4.70E+05	<b>2.07E+03</b>
F7	SR	6.67%	0.00%	0.00%	0.00%	0.00%	33.33%	<b>100.00%</b>	60.00%	3.33%	<b>100.00%</b>
	SP	2.60E+06	Inf	Inf	Inf	Inf	2.66E+05	1.08E+05	2.50E+04	6.04E+05	<b>2.13E+03</b>
F8	SR	0.00%	<b>100.00%</b>	0.00%	<b>100.00%</b>	6.67%	70.00%	<b>100.00%</b>	86.67%	73.33%	60.00%
	SP	Inf	1.11E+05	Inf	5.00E+04	1.60E+05	1.05E+05	<b>4.54E+03</b>	1.34E+04	1.13E+04	1.11E+05
F9	SR	0.00%	<b>100.00%</b>	0.00%	<b>100.00%</b>	3.33%	53.33%	<b>100.00%</b>	36.67%	16.67%	<b>100.00%</b>
	SP	Inf	1.05E+05	Inf	4.79E+04	3.36E+05	1.07E+05	5.64E+03	1.25E+05	6.96E+04	<b>2.42E+03</b>
F10	SR	0.00%	<b>100.00%</b>	0.00%	<b>100.00%</b>	0.00%	16.67%	90.00%	83.33%	16.67%	<b>100.00%</b>
	SP	Inf	1.42E+05	Inf	6.67E+04	Inf	4.59E+05	1.73E+05	3.15E+04	2.11E+05	<b>3.49E+03</b>
F11	SR	0.00%	0.00%	0.00%	0.00%	0.00%	53.33%	20.00%	40.00%	6.67%	0.00%
	SP	Inf	Inf	Inf	Inf	Inf	1.07E+05	3.13E+05	3.95E+04	3.84E+05	Inf

(续表)

函数		APSO	CLPSO	CPSO	FLPSO-QIW	FPSO	FIPSO	MPSO-TVAC	RPPSO	PSO-LDIW	EPPSO
F12	SR	0.00%	0.00%	0.00%	0.00%	0.00%	43.33%	16.67%	20.00%	0.00%	100.00%
	SP	Inf	Inf	Inf	Inf	Inf	2.84E+05	1.83E+05	9.26E+04	Inf	2.05E+03
F13	SR	0.00%	0.00%	0.00%	0.00%	0.00%	0.00%	70.00%	0.00%	0.00%	0.00%
	SP	Inf	Inf	Inf	Inf	Inf	Inf	2.65E+05	Inf	Inf	Inf
F14	SR	86.67%	0.00%	0.00%	0.00%	0.00%	0.00%	70.00%	0.00%	0.00%	0.00%
	SP	2.51E+05	Inf	Inf	Inf	Inf	Inf	2.74E+05	Inf	Inf	Inf
F15	SR	100.00%	100.00%	6.67%	100.00%	0.00%	100.00%	100.00%	100.00%	13.33%	90.00%
	SP	1.20E+04	7.44E+05	9.05E+05	4.88E+04	Inf	2.47E+03	4.42E+03	9.09E+02	4.54E+03	1.13E+05

注:SR为成功率,表示算法达到规定阈值的实验次数占总实验次数的比例;SP为收敛速度,表示算法在固定函数迭代次数内满足规定阈值所需的平均函数迭代次数

由此可知,生态金字塔系统中第三、四营养级有效地提高了种群粒子的收敛速度和多数函数的寻优效率,降低了寻优时间,但对于解决复杂高维函数(如F8和F15),EP-PSO算法的鲁棒性略低于CLPSO等部分改进算法,不能以100%的概率达到函数目标的精度要求。

3) T-检验分析

本次对比实验产生了大量实验数据,为分析对比算法性能带来了许多困难,对此采用双侧T检验方法对数据进行统计分析,客观评价EP-PSO算法与对比算法之间的性能差异

程度,差异显著性检验结果如表6所列。设显著性水平 $\alpha=0.05$ ,每个测试重复执行30次,自由度 $n=29$ ,查表得 $t_{0.05}(29)=2.0452$ 。当 $|t|>t_{0.05}(29)$ 时,认为差异显著,记为 $h=“+”$ ;当 $|t|<t_{0.05}(29)$ 时,认为EP-PSO算法与相应改进PSO算法无显著差异,记为 $h=“-”$ 或 $h=“=”$ 。最后一行数值“w/l/t”统计15个标准测试函数中优化结果有显著差异的函数个数(w)、无显著差异的个数(l)和效果相差不大的函数个数(t)。由表6可见,与9种改进算法相比,EP-PSO算法对大多数测试函数的优化性能差异显著。

表6 EP-PSO算法与改进PSO算法的差异显著性检验

函数		APSO	CLPSO	CPSO	FLPSO-QIW	FPSO	FIPSO	MPSO-TVAC	RPPSO	PSO-LDIW
F1	T-检验	7.57E+00	2.41E+00	6.71E+00	2.66E+00	5.51E+00	2.01E+00	NaN	2.35E+00	3.50E+00
	h	+	+	+	+	+	=	=	+	+
F2	T-检验	1.21E+01	7.48E+00	6.80E+00	2.16E+00	8.56E+00	3.22E+00	2.74E+00	7.53E+00	9.08E+00
	h	+	+	+	+	+	+	+	+	+
F3	T-检验	1.66E+01	2.81E+01	1.53E+01	1.62E+01	1.42E+01	2.80E+00	4.92E+00	1.19E+01	7.17E+00
	h	+	+	+	+	+	+	+	+	+
F4	T-检验	2.49E+01	1.19E+01	3.51E+01	1.45E+01	7.87E+00	3.46E+00	-Inf	-Inf	-Inf
	h	+	+	+	+	+	+	+	+	+
F5	T-检验	5.01E+00	-3.89E+01	6.40E+00	-6.16E+01	9.28E+00	1.88E+01	-1.50E+01	3.57E+01	2.08E+00
	h	+	+	+	+	+	+	+	+	+
F6	T-检验	5.06E+00	4.62E+01	3.69E+01	9.37E+00	9.93E+00	2.32E+00	2.56E+00	3.27E+00	8.10E+00
	h	+	+	+	+	+	+	+	+	+
F7	T-检验	6.12E+00	4.55E+01	3.40E+01	1.30E+01	9.17E+00	3.61E+00	3.84E+00	3.53E+00	1.07E+01
	h	+	+	+	+	+	+	+	+	+
F8	T-检验	1.07E+01	-2.93E+08	5.48E+00	-2.16E+01	1.09E+01	2.90E+00	-Inf	-6.42E-01	3.07E+00
	h	+	+	+	+	+	+	+	=	+
F9	T-检验	1.41E+01	2.60E+01	5.26E+01	1.80E+01	8.96E+00	2.75E+00	Inf	4.46E+00	1.11E+01
	h	+	+	+	+	+	+	+	+	+
F10	T-检验	1.58E+01	NaN	5.36E+01	2.24E+00	7.81E+00	5.63E+00	1.79E+00	2.06E+00	7.51E+00
	h	+	=	+	+	+	+	+	+	+
F11	T-检验	1.56E+01	7.26E+01	4.41E+01	3.21E+01	1.72E+01	5.65E-01	5.34E+00	2.30E+00	1.09E+01
	h	+	+	+	+	+	+	+	+	+
F12	T-检验	2.31E+01	6.98E+01	2.43E+01	3.29E+01	2.24E+01	4.43E+00	8.63E+00	8.27E+00	1.45E+01
	h	+	+	+	+	+	+	+	+	+
F13	T-检验	-9.63E+02	-3.71E+01	2.42E+01	-2.86E+02	8.47E+00	-1.12E+00	-1.61E+03	3.36E+00	1.48E+01
	h	+	+	+	+	+	+	+	+	+
F14	T-检验	-1.26E+04	3.41E+01	3.26E+01	-2.13E+01	2.69E+01	1.73E+01	-2.81E+02	1.94E+01	2.57E+01
	h	+	+	+	+	+	+	+	+	+
F15	T-检验	-Inf	-1.44E+06	1.03E+01	-2.37E+00	1.73E+01	-Inf	-Inf	-Inf	8.70E+00
	h	+	+	+	+	+	+	+	+	+
w/l/t		15/0/0	14/0/1	15/0/0	15/0/0	15/0/0	15/0/0	14/0/1	14/0/1	15/0/0

注:T检验为EP-PSO算法与其他9种算法进行比较后的T检验值,以置信度95%为界,所有算法针对每个函数独立运算n次(n=30),目的是验证性能的提升是否具有统计学意义;h为根据T检验值判断EP-PSO算法的性能是否得到了提升,若显著提升则表示为“+”,具有相同的统计学意义则为“=”,否则为“-”;w/l/t为EP-PSO算法有w个函数寻优结果优于对比算法,有l个函数的结果差于对比算法,有t个函数与对比算法结果一致

4)算法复杂度分析及各算法平均计算时间的比较

算法复杂度是评估算法性能的一个重要指标,决定了算法的执行效率,并且在很大程度上影响了计算机的求解能力。由 3.3 节的算法流程可知,生态金字塔粒子群算法主要在算法的迭代进化过程中花费计算时间。设求解问题的维数为  $D$ ,种群规模为  $N$ ,最大迭代次数为  $T$ 。在每一次的进化迭代过程中,算法主要的计算量包含粒子个体最优位置更新与全局最优位置更新两部分。在迭代更新过程中,最多对种群中的每个粒子进行 2 次操作,最复杂情况下的最大计算时间复杂度为  $2T \times O(N \times D)$ ,保留最优个体的存储空间复杂度为  $O(N \times D)$ ,因此该改进算法的最大复杂度为:

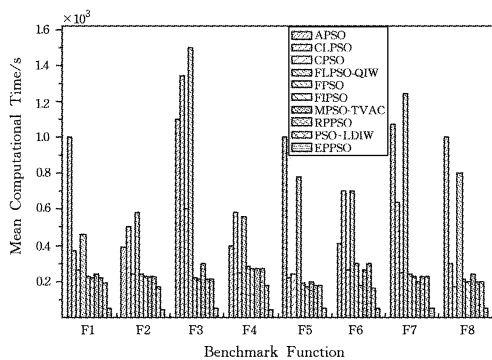
$$O(T, N, D) = 2T \times O(N \times D) + O(N \times D) \quad (10)$$

基本粒子群算法在迭代过程中同样需要更新各粒子的速度和位置,并计算适应度值及保留最优个体,其总体算法复杂度为:

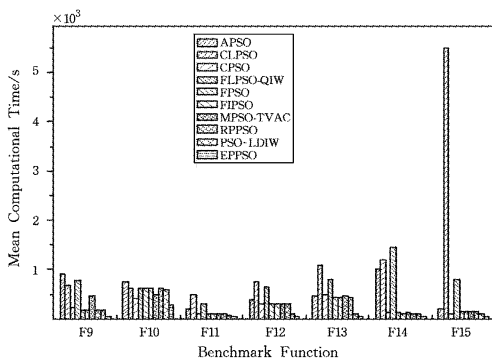
$$O(T, N, D) = T \times O(N \times D) + O(N \times D) \quad (11)$$

由式(10)和式(11)可知,相对于基本粒子群算法,该生态金字塔粒子群算法并没有明显地增加算法的复杂度。

图 3 给出了各改进算法在 15 个测试函数经过 30 次实验迭代优化后所用的平均计算时间,因函数特性不同,各算法在每个函数迭代的过程中所用的时间不尽相同。



(a) F1-F8



(b) F9-F15

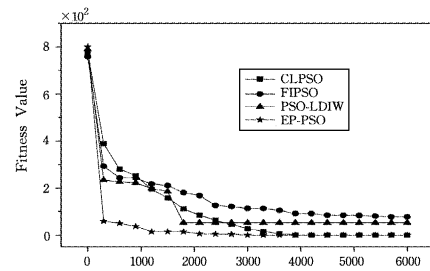
图 3 各算法的平均计算时间

总体来说,改进算法 APSO, CLPSO 和 FLPSO-QIW 的时间复杂度最大, CPSO, FPSO, FIPSO, MPSO-TVAC, RPPSO 和 PSO-LDIW 次之,本文所提出的 EP-PSO 改进算法在函数

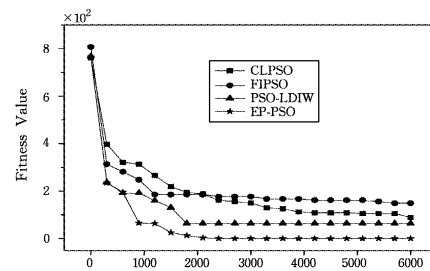
F1-F15 的迭代过程中达到相同的函数迭代次数要求时所用时间最少。

5)适应度值迭代优化曲线分析

为更直观地反映算法的寻优效果,将 EP-PSO 与改进算法 CLPSO, FIPSO, PSO-LDIW 进行比较分析。以多峰函数 F6 和旋转函数 F12 为例,可以看出 EP-PSO 算法由于采用生态金字塔寻优策略,拥有较快的收敛速度和较高的收敛精度,提高了传统粒子群优化算法搜索全局最优解的能力。



(a) F6



(b) F12

图 4 部分算法的收敛曲线分析

**结束语** 针对粒子群优化算法在解决复杂高维函数时容易陷入局部最优和早熟收敛的问题,受生态金字塔系统启发,提出生态金字塔粒子群优化(EP-PSO)算法。与 APSO, CLPSO, CPSO, FLPSO-QIW 等 9 种改进算法相比,引入生态金字塔系统的生态金字塔粒子群优化(EP-PSO)算法有效提高了寻优粒子多样性,该算法拥有更好的寻优能力,更容易避免陷入早熟收敛。EP-PSO 算法得到的函数适应度值精度高,寻优时效好,具有更好的优越性。但在解决函数 F8, F15 时, EP-PSO 算法仍然有一些不足之处,达到目标精度的成功率略低于改进算法 CLPSO, FLPSO-QIW, MPSO-TVAC 和 FIPSO,在以后的研究中,我们将针对这些问题对 EP-PSO 算法进行进一步改进,提升该算法的综合性能,增强其普适性。

参考文献

[1] KENNEDY J. Particle Swarm Optimization[C]//Proc of IEEE International Conference on Neural Networks, 1995:1942-1948.  
[2] LIU J, MA D, ZHANG W, et al. Dynamic Multi-Species Particle Swarm Optimization Based on Food Chain Model[J]. Journal of Computer Application, 2016, 36(5):1341-1346. (in Chinese)

- 刘角, 马迪, 张玮, 等. 基于食物链机制的动态多物种粒子群算法[J]. 计算机应用, 2016, 36(5): 1341-1346.
- [3] ZHANG W. Research on Particle Swarm Optimization Algorithm and its Application in Array Antenna [D]. Taiyuan: Taiyuan University of Technology, 2010. (in Chinese)  
张玮. 粒子群优化算法研究及在阵列天线中的应用[D]. 太原: 太原理工大学, 2010.
- [4] SHI Y, EBERHART R. A modified particle swarm optimizer [C]// Proceedings of the 1998 IEEE World Congress on Computational Intelligence. Piscataway, NJ: IEEE, 1998: 69-73.
- [5] ZHANG W, LIU J, FAN L B, et al. Control Strategy PSO[J]. Applied Soft Computing, 2016, 38(C): 75-86.
- [6] LIANG J J, QIN A K, SUGANTHAN P N, et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions [J]. IEEE Transactions on Evolutionary Computation, 2006, 10(3): 281-295.
- [7] LIM W H, MATISA N A. Teaching and peer-learning particle swarm optimization[J]. Applied Soft Computing, 2014, 18(18): 39-58.
- [8] ZHANG Y J, LI L, ZHANG Y H, et al. An Improved Particle Swarm Optimization Algorithm Based on Two subpopulation [J]. Journal of Hunan University (Natural Sciences), 2011, 38(1): 84-88. (in Chinese)  
张英杰, 李亮, 张英豪, 等. 一种基于双子群的改进粒子群优化算法[J]. 湖南大学学报(自然科学版), 2011, 38(1): 84-88.
- [9] WANG W B, FENG Q Y. Chaotic particle swarm optimization algorithm based on hierarchical multi-subpopulation[J]. Control and Decision, 2010, 25(11): 1663-1668. (in Chinese)  
王维博, 冯全源. 基于分层多子群的混沌粒子群优化算法[J]. 控制与决策, 2010, 25(11): 1663-1668.
- [10] ZHAO X C, LIU G L, LIU H Q, et al. Particle swarm optimization based on non-uniform mutation and multiple stages perturbation[J]. Chinese Journal of Computers, 2014, 37(9): 2058-2070. (in Chinese)  
赵新超, 刘国莅, 刘虎球, 等. 基于非均匀变异和多阶段扰动的粒子群优化算法[J]. 计算机学报, 2014, 37(9): 2058-2070.
- [11] ZHAO Y D, FANG Z H. Pariticle swarm optimization algorithm with weight function's learning factor[J]. Journal of Computer Application, 2013, 33(8): 2265-2268. (in Chinese)  
赵远东, 方正华. 带有惯性权重函数学习因子的粒子群算法[J]. 计算机应用, 2013, 33(8): 2265-2268.
- [12] LINDEMAN R L. The trophic-dynamic aspect of ecology[J]. Ecology, 1942, 23(10): 399-418.
- [13] HU W, LI Z S. A simpler and More Effective Particle Swarm Optimization[J]. Journal of Software, 2007, 18(4): 861-868. (in Chinese)  
胡旺, 李志蜀. 一种更简化而高效的粒子群优化算法[J]. 软件学报, 2007, 18(4): 861-868.
- [14] ZOU D X, WANG G N, GAO L Q, et al. Application of Improved PSO Algorithm to Reliability Problems[J]. Journal of Northeastern University (Natural Science), 2010, 31(9): 1234-1237. (in Chinese)  
邹德璇, 王高楠, 高立群, 等. 改进的粒子群优化算法在可靠性问题中的应用[J]. 东北大学学报(自然科学版), 2010, 31(9): 1234-1237.
- [15] SUGANTHAN P N, HANSEN N, LIANG J J, et al. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization [R]. Singapore, Nanyang Technological University, 2005.
- [16] LIANG J J, QIN A K, SUGANTHAN P N, et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions[J]. IEEE Transactions on Evolutionary Computation, 2006, 10(3): 281-295.
- [17] ZHAN Z H, ZHANG J, LI Y, et al. Adaptive particle swarm optimization[J]. IEEE Transactions on Systems, Man, and Cybernetics, 2009, 39(6): 1362-1381.
- [18] CLERC M, KENNEDY J. The particle swarm explosion, stability, and convergence in a multidimensional complex space[J]. IEEE Transactions on Evolutionary Computation, 2002, 6(1): 58-73.
- [19] TANG Y, WANG Z, FANG J A. Feedback learning particle swarm optimization[J]. Applied Soft Computing, 2011, 11(8): 4713-4725.
- [20] DE OCA M A M, STUTZLE T, BIRATTARI M, et al. Frankenstein's PSO: a composite particle swarm optimization algorithm[J]. IEEE Transactions on Evolutionary Computation, 2009, 13(5): 1120-1132.
- [21] MeENDES R, KENNEDY J, NEVES J. The fully informed particle swarm: simpler, maybe better[J]. IEEE Transactions on Evolutionary Computation, 2004, 8(3): 204-210.
- [22] RATNAWEERA A, HALGAMUGE S K, WATSON H C. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients[J]. IEEE Transactions on Evolutionary Computation, 2004, 8(3): 240-255.
- [23] ZHOU D, GAO X, LIU G, et al. Randomization in particle swarm optimization for global search ability[J]. Expert Systems with Applications, 2013, 38(12): 15356-15364.