

# 引入多级扰动的混合型粒子群优化算法<sup>\*</sup>

徐利锋, 黄祖胜, 杨中柱, 丁维龙

(浙江工业大学 计算机科学与技术学院, 浙江 杭州 310023)

通讯作者: 丁维龙, E-mail: wlding@zjut.edu.cn



**摘要:** 为解决粒子群优化算法易陷入局部最优值的问题, 提出一种引入多级扰动的混合型粒子群优化算法. 该算法结合两种经典改进粒子群优化算法的优点, 即带惯性参数的标准粒子群优化算法和带收缩因子的粒子群优化算法, 在此基础上, 引入多级扰动机制: 在更新粒子位置时, 引入一级扰动, 使粒子对解空间的遍历能力得到加强; 若优化过程陷入“局部最优”的情况, 则引入二级扰动, 使得优化过程继续, 从而摆脱局部最优值. 使用了 6 个测试函数——Sphere 函数、Ackley 函数、Rastrigin 函数、Styblinski-Tang 函数、Duadric 函数及 Rosenbrock 函数来对所提出的混合型粒子群优化算法进行仿真运算和对比验证. 模拟运算的结果表明: 所提出的混合型粒子群优化算法在对测试函数进行仿真时, 其收敛精度和收敛速度都优于另外两种经典的改进粒子群优化算法; 另外, 在处理多峰函数时, 本算法不易被局部最优值所限制.

**关键词:** 粒子群优化算法; 混合; 多级扰动; 局部最优值; 遍历能力

**中图法分类号:** TP301

中文引用格式: 徐利锋, 黄祖胜, 杨中柱, 丁维龙. 引入多级扰动的混合型粒子群优化算法. 软件学报, 2019, 30(6): 1835-1852. <http://www.jos.org.cn/1000-9825/5600.htm>

英文引用格式: Xu LF, Huang ZS, Yang ZZ, Ding WL. Mixed particle swarm optimization algorithm with multistage disturbances. Ruan Jian Xue Bao/Journal of Software, 2019, 30(6): 1835-1852 (in Chinese). <http://www.jos.org.cn/1000-9825/5600.htm>

## Mixed Particle Swarm Optimization Algorithm with Multistage Disturbances

XU Li-Feng, HUANG Zu-Sheng, YANG Zhong-Zhu, DING Wei-Long

(School of Computer Science and Technology, Zhejiang University of Technology, Hangzhou 310023, China)

**Abstract:** To avoid the locally optimum which is frequently be the result of a calculation of particle swarm optimization (PSO) algorithm, it is proposed in this study a new mixed PSO algorithm with multistage disturbance (MPSO). MPSO combined features from two former classic improved PSO algorithms, which are standard particle swarm optimization (SPSO) and standard particle swarm optimization with a constriction factor (PSOCF). Furthermore, a strategy with multistage disturbances was also introduced into the algorithm: The first-level disturbance was used to enhance the ability of the particles to traverse the solution space when renewing the positions, while the second-level disturbance would be introduced when locally optimal solution was received to continue the optimization process. Six test functions, namely the Sphere, Ackley, Rastrigin, Styblinski-Tang, Duadric, and Rosenbrock functions, were used to simulate the optimization calculation, and the results from proposed algorithm MPSO were compared with those from SPSO and PSOCF. The results show that for the test functions, MPSO can get the optimal value much more quickly and easily than the other two algorithms, and the convergence precision of MPSO was significantly higher than the others. It can be concluded that MPSO can get over the problem of locally optimal solution when dealing with multimodal functions.

**Key words:** particle swarm optimization algorithm; mixed; multistage disturbances; locally optimum; ergodic ability

\* 基金项目: 国家自然科学基金(31301230, 61571400, 31471416); 浙江省自然科学基金(LY18C130012)

Foundation item: National Natural Science Foundation of China (31301230, 61571400, 31471416); Natural Science Foundation of Zhejiang Provinc (LY18C130012)

收稿时间: 2017-11-04; 修改时间: 2018-02-12, 2018-04-12; 采用时间: 2018-05-15

粒子群优化算法(particle swarm optimization,简称 PSO)是由 Kennedy 和 Eberhart 于 1995 年提出的一种模拟鸟群觅食特性的进化算法<sup>[1]</sup>.在算法中,粒子代表着觅食中的鸟个体.在每个粒子移动时,同时考虑整个群体的最优位置和当前粒子曾经经历过的最优位置,类似于鸟群觅食时鸟之间的信息交流.当群体中所有粒子都移动之后,记为一轮优化迭代完成,接着进行下一轮优化迭代,直到满足预设的迭代深度或者其他条件为止.整个群体就像鸟群觅食一样,总是向着最优位置移动.

PSO 算法具有易操作和收敛快等优点,但也存在着一些问题,比如因其收敛快而导致的易陷入局部最优值,以及算法在多局部峰值场景中的精确性不高等问题.研究者们一直在针对 PSO 的这些问题进行针对性的研究和改进,并提出一些解决方法,形成了若干 PSO 的改进算法.其中,Shi 和 Eberhart 于 1998 年提出的带有惯性权重的粒子群优化算法<sup>[2]</sup>,便是针对最初的 PSO 算法易陷入局部最优值的改进.Clerc 于 1999 年提出了带压缩因子的粒子群优化算法<sup>[3]</sup>,意在摆脱局部最优值的同时提高收敛速度.这是两种最经典的改进 PSO 算法,其他的改进算法一般都是在此基础之上通过与其他方法结合或是针对参数进行修改来获得新的改进算法.例如,刘丽珏等人提出了基于克隆选择的粒子群优化算法<sup>[4]</sup>,主要利用了克隆和变异操作来提高算法的收敛速度,并保持种群多样性.再如,针对多目标优化问题而提出的基于网格排序的多目标粒子群优化算法<sup>[5]</sup>、为逼近非凸或不连续的 Pareto 最优前端而提出的基于 Pareto 熵的多目标粒子群优化算法<sup>[6]</sup>、面向粒子群优化算法中学习因子自调节优化器的 PSO 算法<sup>[7]</sup>、基于粒子群优化算法的类集成测试序列确定方法<sup>[8]</sup>、加入了微生物行为机制的粒子群优化算法<sup>[9]</sup>以及微分进化算法与粒子群优化的结合<sup>[10]</sup>和其他一些针对 PSO 参数进行自适应选择的算法<sup>[11,12]</sup>等.这些改进的算法直接针对 PSO 算法本身存在的缺陷进行弥补.另外,还有一些改进算法实现了在不同领域中的适配和应用.对于算法适配,例如:对典型的 NP 难题中的 Web 服务组合优化问题,研究者提出了基于改进粒子群算法的 Web 服务组合<sup>[13]</sup>;ECT 系统中如果忽略电容传感器的敏感场(又叫软场),会使传统图像重建方法精度下降,为了清除这个瓶颈,提出了基于双粒子群协同优化的 ECT 图像重建算法<sup>[14]</sup>;为了在不修改源代码的同时能够评价优化目标的每个分支,提出了粒子群优化测试用例生成方法<sup>[15]</sup>.对于算法应用,例如跳跃-滑翔轨迹优化研究(混沌粒子群算法)<sup>[16]</sup>、电力系统环保经济负荷分配应用研究(空间粒子群优化算法)<sup>[17]</sup>以及求解服务链映射问题(离散粒子群优化算法)<sup>[18]</sup>等.

上述各种基于 PSO 的改进和适配后优化算法各有优点,但总体来说都是尝试去解决原始 PSO 算法的固有问题——易于陷入局部极值的问题.针对这种问题,本研究基于现有的两种经典改进 PSO 算法,在不过分提高算法时间复杂度的前提下,通过引入多级扰动策略来避免 PSO 算法易陷入局部最优值的问题,同时也能运用不同的扰动来处理算法探索能力差或算法提前收敛的情况.研究中,运用了 Sphere, Ackley, Rastrigin, Styblinski-Tang, Duadric 和 Rosenbrock 函数对算法的性能进行了测试,并对不同的 PSO 相关优化算法结果进行对比验证.本研究提出的改进型 PSO 算法巧妙地结合了两种经典的改进粒子群优化算法(带惯性权重和带收缩因子的粒子群优化算法)的特性,使得本算法能够将这两种优化算法的优点实现最大化的发挥.同时,针对易陷入局部最优这个问题,提出了多级扰动机制.总体来说,本算法不仅能够增加粒子对于解空间的探索能力,还能够较好地跳出局部最优值.本文首先在第 1 节介绍上述两种经典的改进粒子群优化算法.然后,在第 2 节详细讲述了提出的新的改进型 PSO 算法,即混合型粒子群优化算法的概念及流程.在第 3 节中,通过对 4 个测试函数的结合应用,对混合型粒子群优化算法的可行性及准确性进行对比和验证.最后,对本算法和相关的粒子群优化算法进行讨论,对后续研究进行展望.

## 1 传统粒子群优化算法

### 1.1 粒子群优化算法

最初提出的粒子群优化算法在模拟鸟群觅食的过程中,各个粒子的位置信息代表解空间中的一个可能的解,速度信息则代表粒子的更新(移动)动力.初始的粒子群优化算法总是能高效地进行更新,从而体现出收敛快的特点.

假设:所要优化的问题的种群大小为  $n$ ,解空间维数为  $m$ ,种群中每个粒子的位置为  $X_i(i=1,2,\dots,n)$ ,每个粒子

的适应度值为  $f(X_i)$  ( $i=1,2,\dots,n$ ),第  $k$  代中第  $i$  个粒子的位置为  $X_i^k$ ,粒子的速度为  $V_i^k$ ,第  $i$  个粒子得到的最优适应度值的位置为  $P_i^k$  ( $i=1,2,\dots,n$ ),所有粒子中得到最优适应度值的位置为  $P_g^k$ .公式(1)和公式(2)为粒子群优化算法中对于第  $k$  代粒子的速度、位置更新公式:

$$V_i^{k+1} = V_i^k + c_1 r_1 (P_i^k - X_i^k) + c_2 r_2 (P_g^k - X_i^k) \quad (1)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad (2)$$

其中,  $c_1$  和  $c_2$  为学习因子,  $r_1$  和  $r_2$  为使速度具有随机性的随机数,取值在  $[0,1]$  之间.

从公式(1)和公式(2)中可以看出,粒子速度更新公式包含了 3 部分<sup>[19,20]</sup>:第 1 部分为粒子当前速度对于之前速度及方向的继承能力;第 2 部分为粒子的自我认知部分,用来继承粒子历史属性;第 3 部分为粒子的社会认知部分,表示为所有粒子之间的信息共享,合作地去探索更好的位置.但是由于此方法易陷入局部最优值,所以针对上述更新公式进行一定修改后,得到了下文所提到的两种主流的改进算法.

## 1.2 标准粒子群优化算法

标准粒子群优化算法(standard particle swarm optimization,简称 SPSO),即带有惯性权重的 PSO 算法,在继承粒子历史速度时,引入惯性权重参数.目的是在寻优问题中,能够先进行全局搜索,在缩小的搜索范围或经过一段时间的搜索之后,再针对当前局部进行更为细致的搜索来获得最优解.当惯性权重  $\omega$  取值较大时,全局搜索能力强;较小时,局部搜索能力强.所以,该算法的惯性权重通常是线性递减的<sup>[2]</sup>.

假设:所要优化问题的种群大小为  $n$ ,解空间维数为  $m$ ,种群中每个粒子的位置为  $X_i$  ( $i=1,2,\dots,n$ ),每个粒子的适应度值为  $f(X_i)$  ( $i=1,2,\dots,n$ ),第  $k$  代中第  $i$  个粒子的位置为  $X_i^k$ ,粒子的速度为  $V_i^k$ ,第  $i$  个粒子得到的最优适应度值的位置为  $P_i^k$  ( $i=1,2,\dots,n$ ),所有粒子中得到最优适应度值的位置为  $P_g^k$ .公式(3)、公式(4)为标准粒子群优化算法中对于第  $k$  代粒子的速度、位置更新公式:

$$V_i^{k+1} = \omega V_i^k + c_1 r_1 (P_i^k - X_i^k) + c_2 r_2 (P_g^k - X_i^k) \quad (3)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad (4)$$

其中,  $\omega$  为惯性权重,一般为线性递减,运算方法见公式(5).

$$\omega = \omega_s - (\omega_s - \omega_e) \times \frac{T_c}{T_m} \quad (5)$$

也有研究者使用一种基于对数递减的惯性权重运算方式<sup>[21]</sup>,见公式(6),一般取值由 0.9 递减到 0.4<sup>[22]</sup>:

$$\omega = \omega_s - (\omega_s - \omega_e) \times \log_{T_m} T_c \quad (6)$$

其中,  $\omega_s$  为起始惯性权重值,  $\omega_e$  为结束惯性权重值,  $T_c$  为当前迭代深度,  $T_m$  为最大迭代深度.

除此之外,还有许多其他有关惯性权重的修改方法.例如指数递减策略<sup>[23]</sup>、自适应惯性权重<sup>[24]</sup>等,在此不做详细介绍.

SPSO 算法的大致思路为:粒子群中每个粒子都有自己的位置及速度,每次迭代就相当于更新了群体中每个粒子的位置和速度,粒子的速度更新与粒子前一次速度及自身历史最优位置和群体历史最优位置与当前自身位置的差值有关,粒子的位置更新与速度有关.而 SPSO 算法的特点对于粒子的继承能力部分,添加了一个惯性权重,以此来改变粒子历史速度属性对于当前属性的影响能力.其主要思想为:在优化算法刚开始时,通过较大的惯性权重来使得粒子群的全局搜索能力强于局部搜索能力,而在算法不断进行迭代优化之后,通过减少惯性权重来逐步加强粒子群的局部搜索能力.

## 1.3 带收缩因子的粒子群优化算法

带收缩因子的粒子群优化算法(standard particle swarm optimization with a constriction factor,简称 PSO CF)流程与最初的 PSO 算法相似,区别在于粒子的速度更新公式不同.PSO CF 运用收缩因子来对每次更新后的速度进行压缩,从而提升收敛速度.

假设:所要优化的问题的种群大小为  $n$ ,解空间维数为  $m$ ,种群中每个粒子的位置为  $X_i$  ( $i=1,2,\dots,n$ ),每个粒子

的适应度值为  $f(X_i)(i=1,2,\dots,n)$ ,第  $k$  代中第  $i$  个粒子的位置为  $X_i^k$ ,粒子的速度为  $V_i^k$ ,第  $i$  个粒子得到的最优适应度值的位置为  $P_i^k(i=1,2,\dots,n)$ ,所有粒子中得到最优适应度值的位置为  $P_g^k$ .公式(7)、公式(8)为带收缩因子粒子群优化算法中对于第  $k$  代粒子的速度、位置更新公式:

$$V_i^{k+1} = \phi(V_i^k + c_1r_1(P_i^k - X_i^k) + c_2r_2(P_g^k - X_i^k)) \quad (7)$$

$$X_i^{k+1} = X_i^k + V_i^{k+1} \quad (8)$$

其中,  $\phi$  为收缩因子.

与 SPSO 算法不同,该算法的收缩因子不仅仅是改变了自身历史速度的影响能力,同时也改变了历史最优位置对粒子速度的影响能力.由于收缩因子是固定常量,为了能够较好地平衡全局搜索与局部搜索之间的权重,一般使用公式(9)来计算收缩因子常量<sup>[3]</sup>:

$$\phi = |2 - c + \sqrt{c^2 - 4c}| / 2, c = c_1 + c_2, c > 4 \quad (9)$$

PSOCF 算法的特点是对粒子的继承能力、自我认知能力及社会认知能力部分,添加了一个压缩因子,因此与 SPSO 算法相比,PSOCF 算法在速度更新方面会更快地趋于稳定.其主要思想为扩大了惯性权重的影响范围,因此 PSOCF 算法全局搜索能力的减弱速度以及局部搜索能力的增强速度都得到了提升,从而提升了整体算法的收敛速度.

## 2 新的带多级扰动的混合型粒子群优化算法

SPSO 和 PSOCF 这两种改进优化算法各有特点.在粒子群算法中,不同的参数影响着算法不同的方面:惯性权重影响的是算法中粒子属性的继承性;而压缩因子不仅影响继承性,更影响了整体算法的收敛性.压缩因子过大,收敛性差,且算法接近于随机搜索优化;压缩因子过小,则易提早收敛,导致结果可能为局部最优值,精度下降.从针对粒子进行分层<sup>[24]</sup>得到灵感,本研究提出的新型改进算法——带多级扰动的混合型粒子群优化算法(mixed particle swarm optimization with multistage disturbances,简称 MPSO),是将以上两种优化算法的优点进行整合,在考虑粒子能够较好地继承自身最优解和群体最优解的基础上,引入多级扰动来增加粒子位置变异的随机性以及防止粒子的早熟收敛.

粒子群优化算法具有收敛快的特点,使得算法容易陷入局部最优值而使得最终优化结果产生误差,因此在本研究中引入多级扰动来防止早熟收敛.多级扰动策略分为两级:一级扰动的主要作用是使粒子能够更充分地遍历解空间,类似于混沌扰动<sup>[25]</sup>,都是为了增加粒子的遍历能力;二级扰动则是为应对在多峰值情况下易陷入局部最优值的问题而提出的辅助策略,提高搜索结果的精度,类似于蜜蜂寻找蜂蜜.目前已寻找到暂时最佳的,但是仍需要再去其他地方探索观察是否有更为丰富的蜂蜜,那么便假设此时刻的最优位置的食物已耗尽,以此为基础再去寻找其他食物<sup>[26]</sup>.一级扰动在粒子更新位置之后引入.具体的引入概率与当前迭代深度( $T_c$ )及最大迭代深度( $T_m$ )有关,见公式(10).

$$\varepsilon = 1 - \sin\left(\frac{\pi}{2} \times \frac{T_c}{T_m}\right) \quad (10)$$

其中,  $\varepsilon$  为引入一级扰动的概率.得到的概率会随着迭代深度的增加而减少.与此方法类似的是将粒子分为两种:一种负责标准运行,另一种负责群体多样性.将两种粒子进行结合,可以趋近粒子在探索和收敛间的平衡<sup>[27]</sup>.二级扰动在一级扰动判断之后引入.是否引入二级扰动,与粒子是否早熟的判断有关:若在优化过程探索期中最优粒子连续出现 10 次相同状态(前期实验表明:触发二级扰动的相同次数分别为 10,20,30,40 时,精度都没有显著差异(见表 1),因此设置为 10 次),则判定为局部收敛(早熟),从而引入二级扰动,使粒子快速跳出局部最优值.

MPSO 算法主要是将优化过程的中前期定义为探索期,尽量保持较强的全局搜索能力;而在优化后期,为了快速收敛,尽量加强局部搜索能力.结合前述两种经典优化算法的优点:探索期中利用惯性权重及一级扰动  $A$ (后文公式(12))在优化前期较强的全局搜索能力,尽可能的去遍历解空间,并使用二级扰动来摆脱局部极值;在优化后期,则使用收缩因子及一级扰动  $B$ (后文公式(13))来增强粒子的局部搜索能力,从而完成收敛.

**Table 1** Result of four optimization algorithms for different second disturb condition of five test functions**表 1** 4 种优化算法对于不同二级扰动条件的 5 种测试函数的优化结果

测试函数	维度	二级扰动前置条件			
		10	20	30	40
Sphere	10	$10^{-33}$	$10^{-32}$	$10^{-33}$	$10^{-33}$
Ackley	10	$10^{-14}$	$10^{-14}$	$10^{-14}$	$10^{-14}$
Rastrigin	10	$10^{-15}$	$10^{-15}$	$10^{-15}$	$10^{-15}$
Duadric	10	$10^{-15}$	$10^{-15}$	$10^{-16}$	$10^{-15}$
Rosenbrock	10	$10^0$	$10^0$	$10^0$	$10^0$

基于与前述现有算法相同的假设,MP SO 算法具体优化流程如下.

- Step 1. 初始化粒子群中所有粒子的位置和速度.
- Step 2. 计算每个粒子的适应度值.
- Step 3. 对于每个粒子,将此次优化得到的适应度值与其自身经历过的历史最优适应度值  $P_i^k$  进行比较,若优于  $P_i^k$  则更新该粒子的历史最优位置.
- Step 4. 当所有粒子都更新粒子历史最优速度  $P_i^k$  之后,将每个粒子的适应度值与整个群体的最优值  $P_g^k$  进行比较,若优于  $P_g^k$  的适应度值,则对  $P_g^k$  进行更新.
- Step 5. 根据当前迭代深度判断使用公式(3)、公式(4)还是公式(7)、公式(8)对所有粒子更新速度及位置.
- Step 6. 更新适应度值.
- Step 7. 根据公式(10)判断是否引入一级扰动:若引入扰动,则到 Step 8;若不引入,则到 Step 9.
- Step 8. 引入一级扰动后再次更新适应度值:若新适应度值优于原适应度值,则保留扰动;否则删除扰动.
- Step 9. 判断当前收敛状况是否满足提早收敛的条件:
  - 若满足,则引入二级扰动,并回到 Step 5;
  - 若不满足,则判断是否满足终止条件:若满足,则结束算法;若不满足,则回到 Step 3.

第 5 步中的速度、位置更新算法有两个:首先使用的是标准粒子群优化算法的公式(3)、公式(4),之后再使用带收缩因子的公式(7)、公式(8).之所以选择这种更新方式,是因为这两种公式各有特点.例如,SP SO 对解空间的探索好,而 PSOCF 的收敛速度快.所以定义中前期为探索期,中后期则可根据情况实现收敛.将其两者结合在一起,可使优化过程尽量彻底且全面,同时又能及时得到收敛得到优化结果.MP SO 算法流程如图 1 所示.其中的一级扰动的算法流程如图 2 所示.

探索期时间点的定义,主要由通过一系列的前期预备实验确定.要确定切换时间点,主要考虑算法的两个因素:探索范围的大小及探索最终结果的精度.由于探索范围与迭代深度正相关,因此将探索最终结果的精度作为首要评价标准,在结果精度无显著差异的情况下,将探索范围的大小纳入评判依据.

最佳切换时间的获取也是一个优化问题,因此,将算法中的切换时间点作为优化目标,通过使用本文提出的混合型粒子群优化算法来获取最优解.由于获取最佳切换时间的算法执行中还会涉及探索期切换的最佳切换时间设置,因而为简化起见,将最佳切换时间优化过程中的当次迭代深度的探索期切换时间设置为前次迭代探索到的最优解.本研究运用 Sphere 函数、Ackley 函数、Rastrigin 函数、Styblinski-Tang 函数、Duadric 函数和 Rosenbrock 函数进行仿真实验,因此也对这 6 个测试函数的优化来获取最优的探索期切换区间.

对这个子问题的优化过程的具体设置是:把切换时间点所处的算法总体迭代过程归一化为[0,1]区间(将区间归一化是因为在不同优化问题中,得到的范围是不确定的,精度也是不确定的,因此为了拓展算法的通用性,将整体区间进行归一化处理);设定优化算法中粒子个数为 10,粒子维度(探索步长)为 10;时间点探索迭代深度为 100;函数极值探索的迭代深度为 2 000;以对不同函数探索到的极值精度(或偏差度)作为适应度函数,评价比较切换时间点优劣.优化结果显示,对所有函数来说,最优值(即极值)探索到的时间点基本上都在 10/20~12/20 之间(即迭代深度 1000~1200).具体对比结果见表 2.

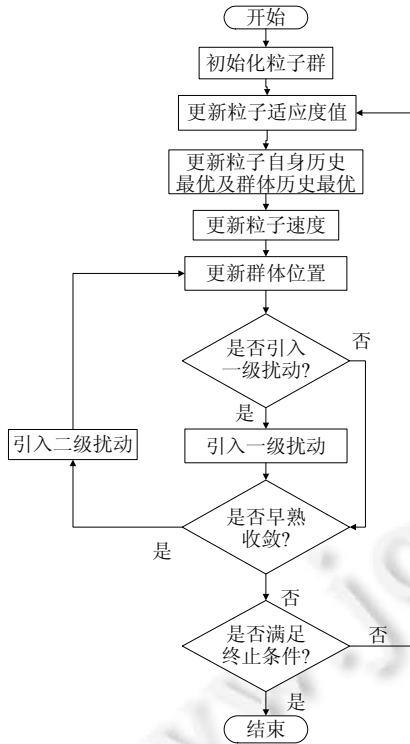


Fig.1 Flow chart of mix particle swarm optimization with perturbation

图 1 混合型粒子群优化算法整体流程图

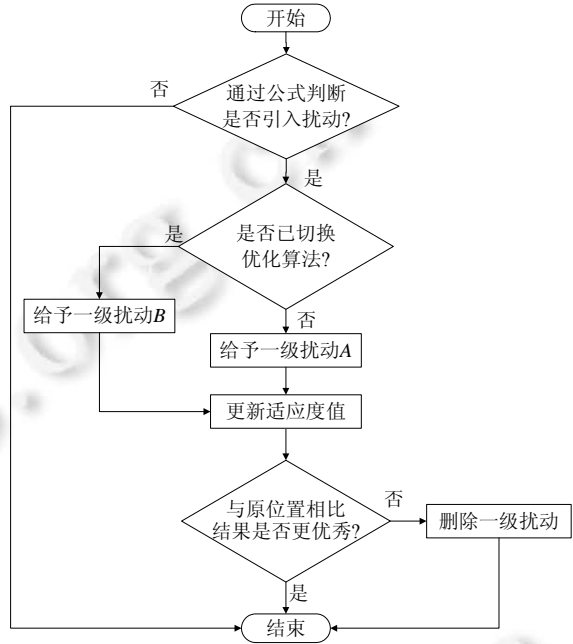


Fig.2 Flow chart of first disturbance

图 2 一级扰动流程图

Table 2 Optimization algorithm for the same dimension of the six test functions for the time switching point

表 2 优化算法对于相同维度的 6 种测试函数的时间切换点优化结果表

测试函数	优化结果精度(函数极值为 0)	优化结果偏差度(% ,函数极值不为 0)	算法切换时间点(迭代深度)
Sphere	$10^{-30}$	-	1 034
Ackley	$10^{-13}$	-	1 024
Rastrigin	$10^{-6}$	-	1 103
Styblinski-Tang	-	0.0001	1 073
Duadric	$10^{-14}$	-	1 116
Rosenbrock	$10^{-1}$	-	1 169

考虑算法存在的系统误差以及一定的随机性因素,将算法的切换时间点定义到合适的范围区间.从表 2 的结果,确定该区间范围为[1/2,3/5].

本文提出算法的迭代深度到达切换时间区间后,通过判断当前最优值第 1 次出现的迭代深度来确定本次优化探索期结束的时间点.因此,本研究提出了公式(11),使得能够方便地在归一化后的区间找到时间切换点  $T$ :

$$T = \frac{T_f}{T_m/2} \times (R_{sa} - R_{st}) + R_{sa} \tag{11}$$

其中,  $T_f$  为当前最优值第 1 次出现的迭代深度,  $R_{sa}$  为归一化整体迭代过程中切换时间区间的开始,  $R_{st}$  为切换时间区间的结束.

为了针对不同状态和特点的粒子,算法中所引入的一级扰动分为两种不同的形式,记为 A 和 B(如图 2 所示).

- 一级扰动 A 在优化前期引入,主要作用是增大粒子对本身邻域的探索能力;同时,reset 保证了粒子有突破当前局限的可能性;

- $B$  部分在算法后期引入,主要作用是对粒子位置进行小幅度的探索修正.  
第  $k$  代中第  $i$  个粒子的位置  $X_i^k$  的具体更新操作见公式(12)、公式(13).

$$A: \begin{cases} X_i^k = r_0 \times X_i^k \\ X_i^k = (r_1 + r_2) \times r_3 \times X_i^k, k < T \\ X_i^k = ga \times X_i^k \\ reset \end{cases} \quad (12)$$

$$B: \begin{cases} X_i^k = r_1 \times X_i^k \\ X_i^k = (r_1 + r_2) / 2 \times X_i^k, T \leq k < T_m \end{cases} \quad (13)$$

其中,  $r_0$  为  $[-2,2]$  之间的随机数,有关  $r_0$  的取值将在本文下文中进行验证;  $T$  为通过公式(11)获得的时间切换点;  $r_1, r_2, r_3$  为  $[0,1]$  的随机数;  $ga$  为标准高斯随机数 ( $\mu=0, \sigma=1$ , 在  $[-3,3]$  的概率为 99.8%);  $reset$  表示对当前粒子的位置在当前解空间中进行随机重置. 一级扰动  $A$  和  $B$  中是两种不同的扰动方式,在具体实施中选择其一进行,且两者选择概率是相同的. 每次引入一级扰动之后,都要检测当前引入的一级扰动是否使得适应度值更优:若更优,则保留此次一级扰动;否则,删除当次扰动(避免无用扰动对结果产生影响).

算法中对粒子陷入局部最优的判断:如果粒子连续 10 次的状态都相同,那么便定义当前粒子陷入了局部最优值. 若已陷入局部最优值,则会引入二次扰动. 二级扰动主要是将粒子的位置重置为与当前局部最优结果有关的位置(见公式(14)),并且让其脱离当前局部最优. 但脱离结果不能过于随机,因此运用了基于最优位置值附近的随机位置分布,来更新粒子位置,从而达到脱离的效果:

$$X_i^k = (X_{gb}^{k-1} + X_{gb}^{k+1}) \times r_a \quad (14)$$

其中,  $r_a$  为  $[-2,2]$  之间的随机数;  $X_{gb}^{k-1}$  和  $X_{gb}^{k+1}$  分别表示第  $k-1$  代和第  $k+1$  代当前群体最优位置值,  $k$  为开始陷入局部极值的迭代深度.

前述公式中,包含两个随机数  $r_0$  和  $r_a$ . 这两个参数在整个算法中,目的是增强粒子的探索能力和跳脱出局部最优值的能力,因此其取值对优化算法的改进比较关键. 在本研究中,运用了一些前期预备实验对这两个随机数的取值范围进行探索. 实验中,首先设置了一个前提假设:这两个随机数的取值范围是基于 0 对称的;随后,选择不同的上下限范围,并对研究中涉及的测试函数(Sphere 函数、Ackley 函数、Rastrigin 函数、Styblinski-Tang 函数、Duadric 函数和 Rosenbrock 函数)进行预运算;最后,对不同取值范围下的优化结果进行评判. 表 2 中列出了不同取值组合设置下的优化结果优劣比较——选用了  $[-2,2]$  范围作为基准,将其他取值范围的优化结果与  $[-2,2]$  的优化结果作对比.

- 若优化结果比基准要好,则认为该取值范围更优;否则,记为更差;
- 在极值为 0 的测试函数优化中,若收敛精度差值不超过 1,或者在极值非 0 的测试函数仿真中收敛误差差不超过 1,则记为无差异;
- 基准本身不进行自身对比.

另外,在对不同的参数范围进行比较时,我们将最优值为 0 的函数占最终结果的比重为 0.5,最优值不为 0 的函数占最终结果的比重为 0.5. 由此,从表 3 中可以得出结论:当  $r_0, r_a$  取值  $[-2,2]$  时,对于最优值为 0 以及非 0 的情况的综合效果最好.

本研究中所引入的多级扰动机制,实质是在相对稳定的优化过程中加入随机性和变异性,并且针对于仍未跳出局部最优值的情况,提供了二级扰动,把粒子进行大幅度的移动,从而摆脱当前局部最优值的影响. 若二级扰动引入后仍陷于局部最优值,则再次引入二级扰动... 以此递进,最多可进行 64 次(前期实验表明,重复引入二级扰动的次数在 16,32,64,128 这 4 种设置下,除 Sphere 函数外,其余的函数都不能体现出显著差异. 而 Sphere 函数则是在 64 次重复引入的情况下,优化结果精度最高(见表 4),因此,这里设置为 64 次). 这是为了能在陷入无限循环之后,从其中跳脱出来的同时又能尝试引入多次扰动来摆脱局部最优.

**Table 3** Result of different  $r_0$  &  $r_a$  to optimization algorithms for different dimension of sixtest functions

**表 3** 不同  $r_0$  &  $r_a$  的 MPSO 算法对于对于不同维度的 6 种测试函数的优化结果表

$r_0$	测试函数	相比结果	$r_a$	测试函数	相比结果
[-0.5,0.5]	Sphere	+	[-0.5,0.5]	Sphere	+
	Ackley	=		Ackley	=
	Rastrigin	+		Rastrigin	+
	Styblinski-Tang	-		Styblinski-Tang	-
	Duadric	+		Duadric	+
	Rosenbrock	=		Rosenbrock	=
[-1,1]	Sphere	-	[-1,1]	Sphere	-
	Ackley	=		Ackley	=
	Rastrigin	=		Rastrigin	+
	Styblinski-Tang	-		Styblinski-Tang	-
	Duadric	=		Duadric	=
	Rosenbrock	=		Rosenbrock	=
[-2,2]	Sphere	/	[-2,2]	Sphere	/
	Ackley	/		Ackley	/
	Rastrigin	/		Rastrigin	/
	Styblinski-Tang	/		Styblinski-Tang	/
	Duadric	/		Duadric	/
	Rosenbrock	/		Rosenbrock	/
[-4,4]	Sphere	-	[-4,4]	Sphere	-
	Ackley	=		Ackley	=
	Rastrigin	+		Rastrigin	+
	Styblinski-Tang	-		Styblinski-Tang	-
	Duadric	-		Duadric	-
	Rosenbrock	=		Rosenbrock	=

注:取值更优的情况标记为“+”,更差的标记为“-”,无差异的用“=”表示,“/”表示不对比

**Table 4** Result of four optimization algorithms for different max number of second disturb of five test functions

**表 4** 4 种优化算法对于二级扰动不同最高次数的 5 种测试函数的优化结果表

测试函数	维度	优化结果精度			
		16	32	64	128
Sphere	10	$10^{-28}$	$10^{-28}$	$10^{-33}$	$10^{-27}$
Ackley	10	$10^{-14}$	$10^{-14}$	$10^{-14}$	$10^{-14}$
Rastrigin	10	$10^{-15}$	$10^{-15}$	$10^{-15}$	$10^{-15}$
Duadric	10	$10^{-14}$	$10^{-14}$	$10^{-15}$	$10^{-15}$
Rosenbrock	10	$10^0$	$10^0$	$10^0$	$10^0$

本研究使用了异步粒子群优化算法的粒子更新方式<sup>[28]</sup>.同步粒子群优化算法的粒子更新方式是在一轮粒子全部更新得到适应度值之后,再更新粒子最优及群体最优.而这种异步的形式是在搜索中,当任意一个粒子发现其计算的适应值优于共享信息中的适应值时,则立即更新共享信息,以此提高粒子的利用效率.同时,粒子群算法根据邻域内粒子数量的不同,可以分为全局和局部两个形式<sup>[11]</sup>:全局版本就是所有粒子都是其中某一颗粒子的邻域成员,局部版本就是只有一部分粒子是某一颗粒子的邻域成员.对于简单的问题,邻域内的粒子越多,效果越好;对于复杂的多模问题,邻域内的粒子越少,效果越好<sup>[29]</sup>.除此之外,还要一些其他的定义方式,如 4 类结构、金字塔结构<sup>[28]</sup>、可变多簇结构<sup>[30]</sup>等.MPSO 采用的是异步全局粒子群算法的形式.

在对指定优化问题进行优化时,粒子的运动范围往往都是限定的,因此需要对超出限定阈值的粒子进行处理,处理方法一般有吸收墙、反射墙、衰减墙和隐匿墙<sup>[31]</sup>.本研究运用了重置的方法——将超出阈值的粒子重置在阈值边界附近,此方法近似于反射墙,但是反射能力被大大削弱,可称为弱反射墙.假设当前问题限制的粒子阈值范围为 $[-\alpha, \alpha]$ ,那么结合反射墙及衰减墙,可将弱反射墙的运算定义为如公式(15)所示形式:

$$X_i^k = \begin{cases} -\alpha + \frac{T_c}{T_m \times r_4}, & X_i^k < -\alpha \\ \alpha - \frac{T_c}{T_m \times r_4}, & X_i^k > \alpha \end{cases} \quad (15)$$



其中,  $r_4$  为  $[0,1]$  之间的随机数.

算法整体的时间复杂度由原本的  $O(n)$  变为  $O(cn)$ , 其中,  $c$  为一个常数 ( $c > 1$ ), 取值与引入扰动的次数呈正相关;  $n$  与粒子维度  $d$ 、种群粒子数  $m$ 、最大迭代次数  $t$  有关.

### 3 算法仿真实验

在本研究中, 为了测试 MPSO 算法的有效性, 选择了 6 个函数——Sphere 函数、Ackley 函数、Rastrigin 函数、Styblinski-Tang 函数、Duadric 函数及 Rosenbrock 函数来进行测试. 同时, 运用现有的 3 个改进 PSO 算法——SPSO, PSO CF 和 PSO-DT<sup>[32]</sup> 进行相同条件的优化运算, 再将三者的优化过程与结果进行对比与验证. 仿真实验在 Windows 10 平台中 Eclipse 环境中进行, 硬件配置为 Inter i5-2430M, RAM 4GB.

#### 3.1 测试函数

##### 1. Sphere 函数.

函数在  $x_i=0$  时达到全局最小值 0, 为单峰值函数. 函数表达式见公式(16).

$$f_1(x) = \sum_{i=1}^n x_i^2, -100 \leq x_i \leq 100 \quad (16)$$

##### 2. Ackley 函数.

函数在  $x_i=0$  时达到全局最小值 0, 且函数在解空间中存在大量局部极小值点. 函数表达式见公式(17).

$$f_2(x) = 20 + e - 20 \times \exp\left(-0.2 \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2}\right) - \exp\left(\frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)\right), -35 \leq x_i \leq 35 \quad (17)$$

##### 3. Rastrigin 函数.

函数在  $x_i=0$  时达到全局最小值 0, 且函数在解空间中存在大量的正弦凸起的局部极小值点. 函数形式见公式(18).

$$f_3(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10), -10 \leq x_i \leq 10 \quad (18)$$

##### 4. Styblinski-Tang 函数.

函数在  $x_i=-2.903534$  时达到全局最小值  $-39.16599$  乘上  $d$  ( $d$  为维度, 表示自变量  $x$  的坐标数目), 且为多维多峰多局部最优值函数. 函数表达式见公式(19).

$$f_4(x) = \frac{1}{2} \sum_{i=1}^d (x_i^4 - 16x_i^2 + 5x_i), -5 \leq x_i \leq 5 \quad (19)$$

##### 5. Duadric 函数.

函数在  $x_i=0$  时达到全局最小值 0, 为单峰值多极值函数. 此函数与 Sphere 函数类似, 但是考虑了维度因素  $d$  (此处同样为自变量  $x$  的坐标数目), 因此更容易陷入局部极值. 函数表达式见公式(20).

$$f_5(x) = \sum_{d=1}^d \left( \sum_{i=1}^n x_i \right)^2, -100 \leq x_i \leq 100 \quad (20)$$

##### 6. Rosenbrock 函数.

函数在  $x_i=0$  时达到全局最小值 0, 为单峰值凹槽型函数. 此函数在极值附近的值存在极微小的差别:

$$f_6(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2), -5 \leq x_i \leq 5 \quad (21)$$

在标准粒子群优化算法中, 有关参数的设置通常是类似的. 一般来说, 学习因子  $c_1, c_2$  取值范围为  $[0,2]$ , 在算法中一般取值为 2.0, 而  $\omega$  则是从 0.9 递减到 0.4<sup>[21]</sup>, 见公式(22).

$$\omega = 0.9 - 0.5 \times \log_{T_m} T_c \quad (22)$$

而在带收缩因子的实验中, 发现了当  $c=c_1+c_2=4.1$ , 根据公式(7), 得到  $\varphi=0.729$  时, PSO CF 算法能够较好的平衡探索及收敛能力<sup>[3]</sup>.

因此,在本研究的仿真实验中,设置各个算法的粒子规模  $n$  为 5,10,15 和 20;标准粒子群算法中, $c_1=c_2=2$ ;惯性权重  $\omega$  采用对数递减,从 0.9 对数递减到 0.4;带收缩因子的粒子群算法中,收缩因子  $\varphi=0.729$ ,学习因子  $c_1=c_2=2.05$ .

3.2 仿真实验结果与分析

通过仿真实验,对上述 4 个函数进行优化测试:定义函数的全局极值为最优值;运用 3 种基于粒子群的优化算法(混合粒子群优化算法、标准粒子群优化算法以及带收缩因子的粒子群优化算法)进行优化运算;算法中,设置 4 种步长维度(5,10,15 和 20,表示优化算法中粒子个数)分别进行运算;单次优化的终止条件为迭代次数达到 2 000 次;实验重复次数为 10 次,取平均值对不同设置和场景(函数)分别进行比较.前 3 个函数——Sphere 函数、Ackley 函数和 Rastrigin 函数因其全局最小值都为零,因此优化算法的优化效果以其最终优化结果值的精度表示,具体对比结果见表 5.而 Styblinski-Tang 的全局最小值不为 0,因此优化结果以算法优化值与函数极值之间的偏差来表示.偏差  $\phi$  的计算方法由公式(23)计算而来:

$$\phi = \left| \frac{V_t - op}{V_t} \right| \tag{23}$$

其中, $V_t$  为 Styblinski-Tang 函数的结果真值, $op$  为优化算法得到的结果值.具体的结果对比见表 6.

Table 5 Result of four optimization algorithms for different dimension of five test functions

表 5 4 种优化算法对于不同维度类型的 5 种测试函数的优化结果表

测试函数	维度	优化结果精度				运行时间(ms)			
		MPSO	SPSO	PSODT	PSOCF	MPSO	SPSO	PSODT	PSOCF
Sphere	5	$10^{-59*}$	$10^0$	$10^{-3*}$	$10^{-40*}$	243	150	105	102
	10	$10^{-33*}$	$10^1$	$10^{-2*}$	$10^{-4*}$	254	72	66	66
	15	$10^{-18*}$	$10^1$	$10^{-1}$	$10^0$	349	122	95	95
	20	$10^{-14*}$	$10^2$	$10^1$	$10^1$	432	128	91	107
Ackley	5	$10^{-15*}$	$10^0$	$10^{-1}$	$10^{-1}$	230	177	109	136
	10	$10^{-14*}$	$10^0$	$10^{-1}$	$10^{-1}$	399	249	262	264
	15	$10^{-10*}$	$10^0$	$10^0$	$10^0$	399	197	138	253
	20	$10^{-7*}$	$10^0$	$10^0$	$10^0$	485	215	215	211
Rastrigin	5	$10^{-15*}$	$10^1$	$10^0$	$10^0$	362	142	79	167
	10	$10^{-15*}$	$10^2$	$10^1$	$10^1$	374	138	107	130
	15	$10^{-13*}$	$10^2$	$10^1$	$10^1$	469	204	188	206
	20	$10^{-7*}$	$10^2$	$10^1$	$10^1$	565	186	174	178
Duadric	5	$10^{-34*}$	$10^1$	$10^{-26*}$	$10^{-3*}$	254	95	112	87
	10	$10^{-15*}$	$10^1$	$10^{-12*}$	$10^{-1}$	397	108	97	99
	15	$10^{-8*}$	$10^1$	$10^{-5*}$	$10^{-1}$	400	82	123	144
	20	$10^{-4*}$	$10^2$	$10^{-2*}$	$10^0$	711	220	141	182
Rosenbrock	5	$10^{-1}$	$10^4$	$10^4$	$10^4$	198	148	103	85
	10	$10^0$	$10^4$	$10^4$	$10^4$	240	101	90	122
	15	$10^1$	$10^4$	$10^4$	$10^4$	161	103	80	152
	20	$10^1$	$10^4$	$10^4$	$10^4$	304	110	105	179

注:\*表示优化结果与函数极值的真值无显著差异

Table 6 Result of three optimization algorithms for different dimension of Styblinski-Tang test functions

表 6 3 种优化算法对于不同维度的 Styblinski-Tang 测试函数的优化结果表

测试函数	维度	$\phi$ (%)				运行时间(ms)			
		MPSO	SPSO	PSODT	PSOCF	MPSO	SPSO	PSODT	PSOCF
Styblinski-Tang	5	0.000 4	3.2	0.07	10.1	145	47	70	53
	10	0.36	3.2	6.9	14.2	162	64	63	60
	15	2.2	9.1	7.7	17.2	320	103	136	86
	20	2.7	16.0	7.9	21.3	472	131	154	112

从表 5 可以看出,在优化结果精度方面,在所有的 4 种不同维度设置下,本研究提出的 MPSO 算法对 3 种最优值为 0 的测试函数的优化结果都显著优于另外两种改进算法的结果;并且在其他算法无法定位到最优值的

情况下,MPSO 算法依然能够找到最优值,且具有较高的精度.具体来说,在 Sphere 和 Ackley 函数上,PSOCF 算法也能找到正确的最优值,但是在最后结果的精度上与 MPSO 算法仍有较大差距;对 Rastrigin 函数的优化结果来说,PSOCF 不能找到正确的最优值,而 MPSO 的最终结果精度较高;而 SPSO 算法对这 3 个测试函数的优化结果的精度都处于最低的位置,几乎都未能定位到最优值.

从维度上来说,随着测试函数维度的提升,所有算法的优化结果精度都相应降低.从表 5 中可以看出,在实验中设置的维度范围内,本研究提出的 MPSO 都能准确的找到最优值并保证一定的精度;SPSO 算法在此范围内基本上都没有能够找到最优值;PSOCF 算法则是在对 Sphere 函数和 Ackley 函数的优化中,以 5 和 10 为维度时能够找到最优值并进行一定的精度探索;但在 15 和 20 维度,则并不能够找到最优值.

整体优化迭代过程如图 3~图 8 所示.结合图 3~图 6 可以发现,在优化前期(探索期),MPSO 算法一直在探索,直到后半期转入低几率弱一级扰动  $B$  后开始逐渐收敛,而在切换更新公式后进行了快速收敛;而 SPSO 算法则较为稳定,一直在探索,最终并没有很好地收敛在最优值上;PSOCF 算法则是在一开始就在最优值附近开始快速收敛;PSODT 与 PSOCF 算法类似,但是最终优化结果的精度在 5 维度、10 维度时没有 PSOCF 算法精确.同时能够发现,其他 3 种算法在 5 维度、10 维度的 Sphere 函数优化时效果较好;在 5 维度、10 维度的 Ackley 函数优化时效果一般,但还是找到了最优值,但在一定深度之后则趋于稳定,不再继续探索;而在其他情况下则并没有找到最优值,而是在局部最优值附近进行快速收敛.从图中我们可以清楚地发现 MPSO 算法的特点:在探索期中,以探索为主;在后半优化过程中,以快速收敛为主.这样就能较好地在平衡探索与收敛基础之上,优化到最优值.而从图 7 能够发现:无论哪种优化算法在维度超过 5 时,得到的优化结果都不够精确;但对同样的优化问题而言,本研究提出的混合型粒子群优化算法总是能得到精度更高的优化结果.

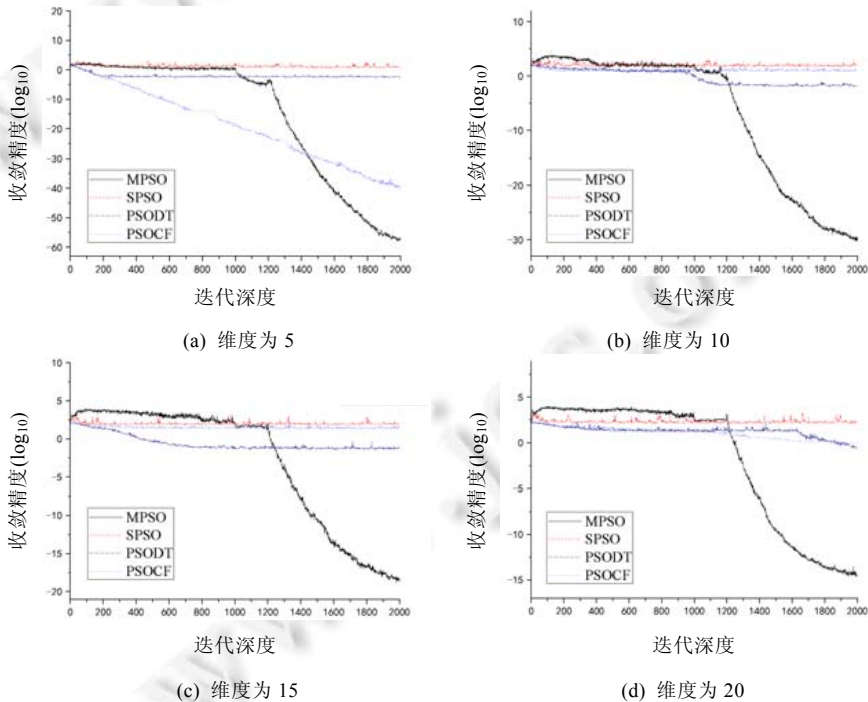


Fig.3 Charts of the optimization dynamics and results of the function Sphere  
图 3 4 种优化算法对 4 种维度下的 Sphere 测试函数优化过程数据曲线

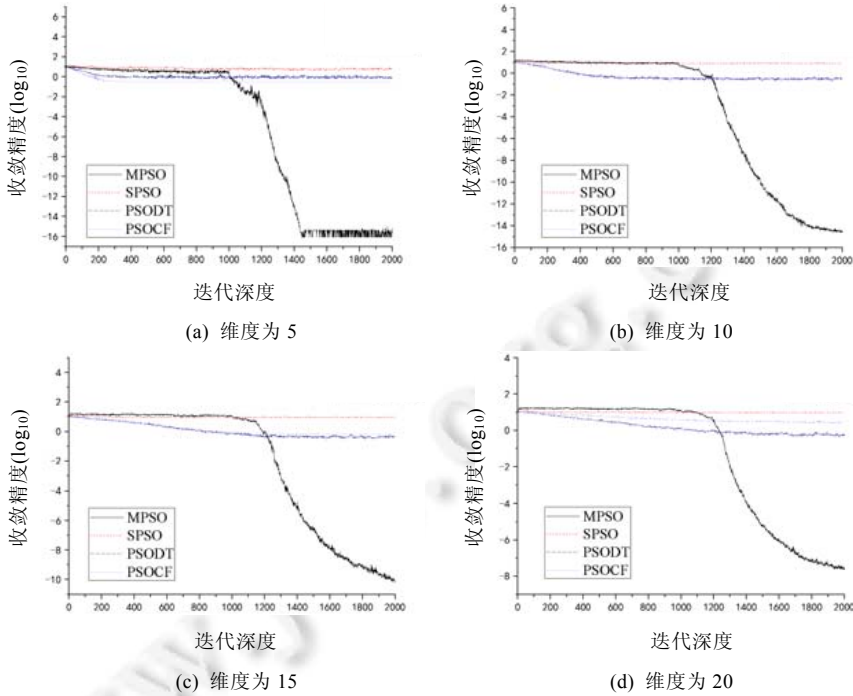


Fig.4 Charts of the optimization dynamics and results of the function Ackley  
图 4 4 种优化算法对 4 种维度下的 Ackley 测试函数优化过程数据曲线

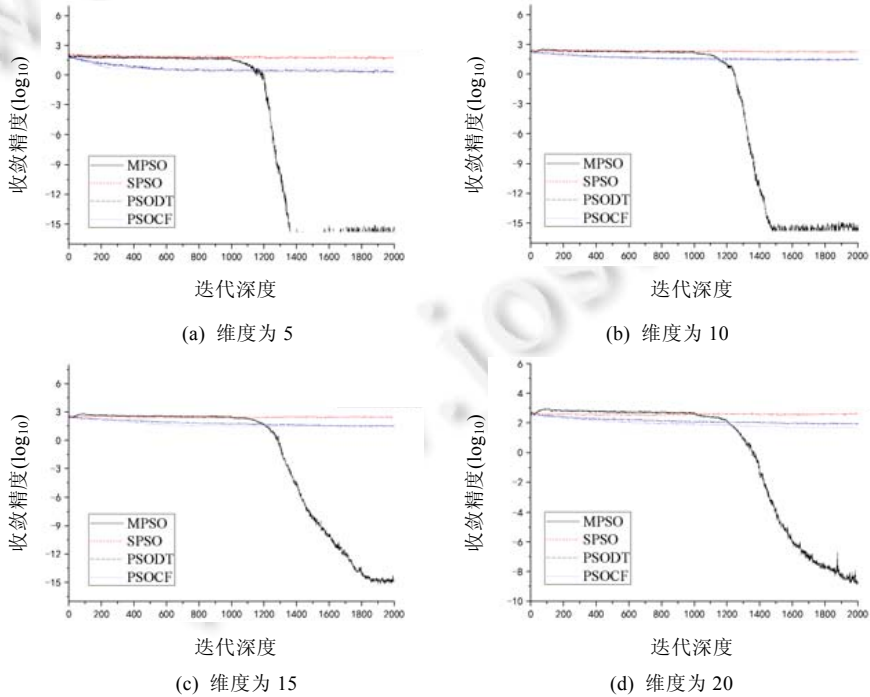


Fig.5 Charts of the optimization dynamics and results of the function Rastrigin  
图 5 4 种优化算法对 4 种维度下的 Rastrigin 测试函数优化过程数据曲线

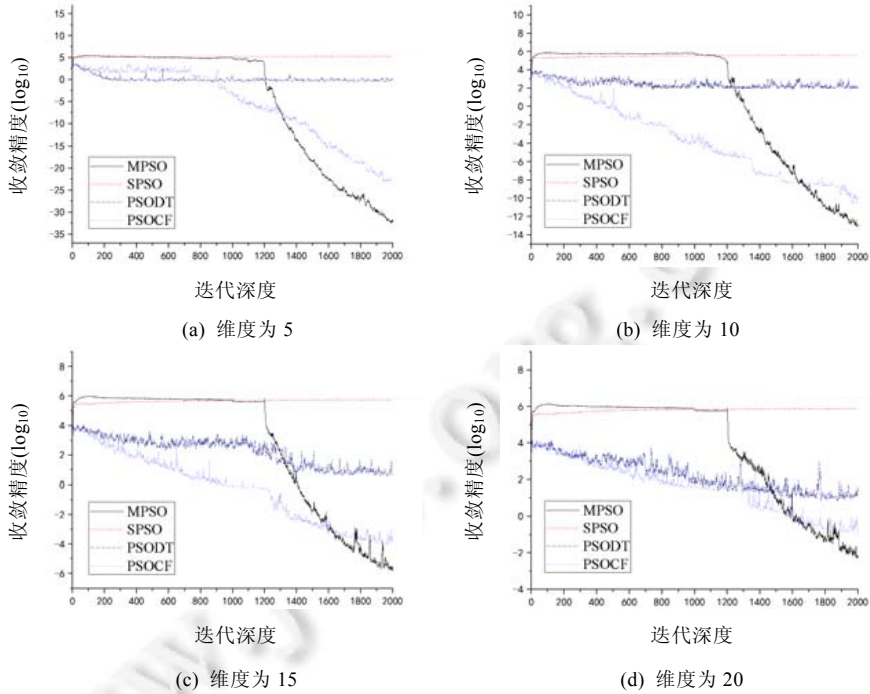


Fig.6 Charts of the optimization dynamics and results of the function Duadic  
图 6 4 种优化算法对 4 种维度下的 Duadic 测试函数优化过程数据曲线

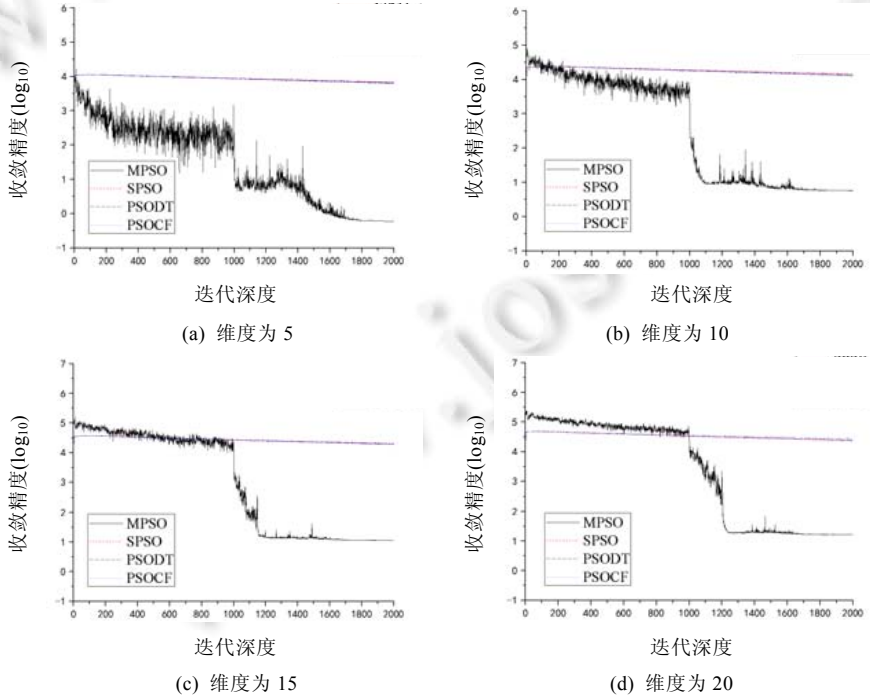


Fig.7 Charts of the optimization dynamics and results of the function Rosenbrock  
图 7 4 种优化算法对 4 种维度下的 Rosenbrock 测试函数优化过程数据曲线

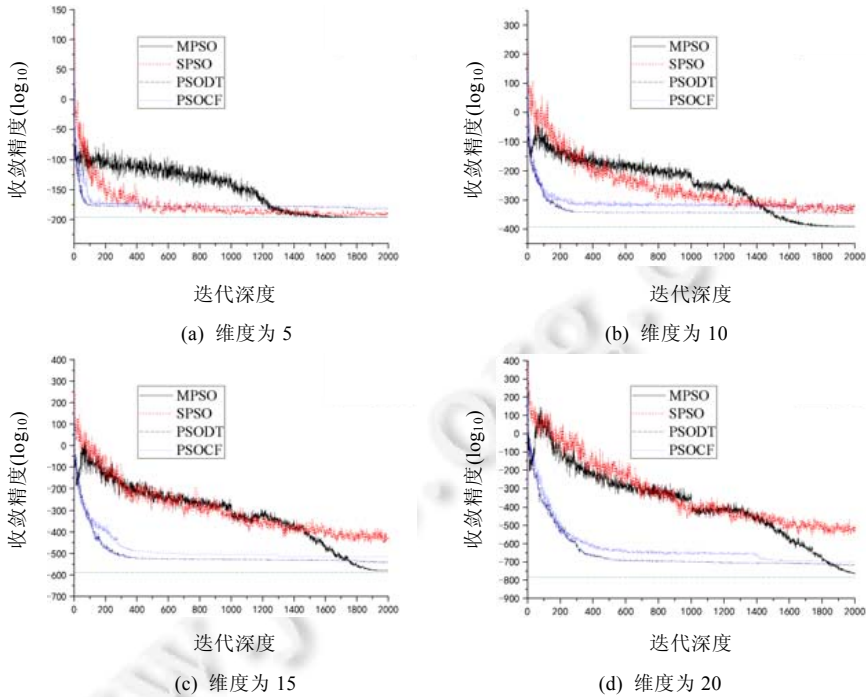


Fig.8 Charts of the optimization dynamics and results of the function Styblinski-Tang

图8 4种优化算法对4种维度下的 Styblinski-Tang 测试函数优化过程数据曲线

从时间上来说,本研究提出的 MPSO 算法中,耗时的主要是引入的多级扰动,因为每次引入一级扰动之后都要重新计算一次适应度值,并因为早熟而引入二级扰动后,重新去引入一级扰动,所以整体时间为原本的 2 倍~4 倍.多出的时间主要为计算适应度值情况下,对原本算法的整体算法流程上所耗费的时间并没有增加,只是增加了一些引入扰动后重新计算适应度值的时间.

从表 6 可以看出,对最优值不为 0 的 Styblinski-Tang 函数的优化测试中,当维度分别为 5 和 10 时,MPSO 算法的精确度较好;但当维度为 15 和 20 时,偏差接近 3%.在考虑到真值较大的情况下,这种偏差率可以接受;SPSO 算法在维度为 5 和 10 时,偏差度达到了 3%左右.虽然偏差度有些大,但还在可接受范围;但是在维度为 15 时,偏差超过了 9%;而在维度为 20 时,偏差更是达到了 16%,这表示了偏差过大,优化结果不够准确;而 PSOCF 算法在测试的维度为 5,10,15 中的偏差都超过了 10%,在维度为 20 时更是超过了 20%,说明 PSOCF 算法对于 Styblinski-Tang 函数优化能力很差;PSODT 算法相对于 SPSO 和 PSOCF 算法来说要更为优秀,但其优化精度仍然不如 MPSO 算法高.结合图 8 同样也发现,MPSO 在前半优化过程中以探索为主,在后半优化过程中以快速收敛为主.这样就能较好地在平衡探索与收敛基础之上,优化到最优值.这也体现了 MPSO 算法在上述情况下的优化结果都优于其他两种算法,同时,耗费时间大概为其他两种算法的 2 倍~4 倍.

## 4 结论

针对经典粒子群优化算法易陷入局部最优及收敛速度较慢等问题,本研究在分析经典的两种粒子群优化算法 SPSO 和 PSOCF 的基础上,提出了带多级扰动的混合型粒子群优化算法.原始的粒子群优化算法在多局部极值情况下容易陷入局部最优值.SPSO 和 PSOCF 算法正式针对这一特点而进行了改进.本研究中提出的 MPSO 算法结合了这两种经典改进优化算法的优点,引入多级扰动的策略,利用多级扰动干扰粒子探索进程,使粒子在本身速度和位置更新之外,获得了一种“震动”,不但增长了算法对局部极值附近的探索能力,也扩展了算法在陷入局部最优值时的跳脱能力,从而在保证算法及时收敛的情况下,使粒子群优化算法的探索能力和最优

值的精度上都得到极大的提升。

通常认为,在 PSO 相关的算法中最重要的参数是惯性权重和学习因子<sup>[33]</sup>。如何选择、调整这些参数,会直接影响粒子群优化算法的收敛性和结果的有效性。在本文中,粒子更新公式中的惯性权重使用的是对数递减形式。另外,还有其他方法对惯性权重进行递减用以针对不同的优化问题,而如何结合具体优化问题设置不同的惯性权重并加以应用、如何对其他参数进行适应性设置以及如何根据迭代情况自适应的去寻找最优参数,这些都没能在本研究中进行实验和探讨。

本研究中对整个优化周期进行了简单的定义——分为探索期和收敛期。在探索期内,粒子保持跳动能力,对整个解空间进行搜索;而过了探索期之后,则允许其进行收敛。这样的分割对学习因子和惯性权重的应用起到了至关重要的作用。因此,研究中对两个时期的切换时间点做了细致的前期实验,而在后续的仿真实验中,发现前期实验所得的最佳切换时间点所处区间确实很好得平衡了搜索能力和收敛能力。另外,算法中运用的两个随机参数  $r_0$  和  $r_a$  取值对整体算法的优化能力有比较关键的影响。本研究中,在基于经验取值的基础上,对着两个参数的取值进行了不同设置下的对比和前期验证。对于优化算法中的参数取值,总是在不同的应用场景下往往有不同的表现。例如,本文提出的 MPSO 算法在对间作中的不同作物的间作模式有较好的优化和预测作用,但其与对不同环境下的水稻田间种植模式优化过程的参数设置和适应度函数选取都有不同的策略。因此,很难说在普遍的场景下有一个通用的取值。在这种情况下,运用其他较为成熟和稳健算法先期对参数进行优化是一个比较好的选择。另外,如果能开发出对不同场景自适应调整的智能策略,将能够比较有力地推动优化问题解决及优化算法方面的研究。

## References:

- [1] Kennedy J, Eberhart RC. Particle swarm optimization. In: Proc. of the IEEE Int'l Conf. on Neural Networks. Piscataway: IEEE Service Center, 1995. 1942–1948.
- [2] Shi Y, Eberhart RC. A modified particle swarm optimizer. In: Proc. of the IEEE Int'l Conf. on Evolutionary Computation. Piscataway: IEEE Press, 1998. 69–73.
- [3] Clerc M, Kennedy J. The particle swarm: Explosion, stability, and convergence in a multidimensional complex space. *IEEE Trans. on Evolutionary Computation*, 2002,6:58–73.
- [4] Liu LJ, Cai ZX. Particle swarm optimization algorithms based on CLONAL. *Mini-micro Systems*, 2006,27(9):1708–1710 (in Chinese with English abstract). [doi: 10.3969/j.issn.1000-1220.2006.09.025]
- [5] Li L, Wang WL, Xu XL, Li WK. Multi-objective particle swarm optimization based on grid ranking. *Journal of Computer Research and Development*, 2017,54(5):1012–1023 (in Chinese with English abstract). [doi: 10.7544/j.issn1000-1239.2017.20160074]
- [6] Hu W, Yen GG, Zhang X. Multi-objective particle swarm optimization based on Pareto entropy. *Ruan Jian Xue Bao/Journal of Software*, 2014,25(5): 1025–1050 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/4496.htm> [doi: 10.13328/j.cnki.jos.004496]
- [7] Ratnaweera A, Halgamuge SK, Watson HC. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients. *IEEE Trans. on Evolutionary Computation*, 2004,8(3):240–255. [doi: 10.1109/TEVC.2004.826071]
- [8] Zhang YM, Jiang SJ, Chen RY, Wang XY, Zhang M. Class integration testing order determination method based on particle swarm optimization algorithm. *Chinese Journal of Computers*, 2016, 41(4):931–945 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2018.00931]
- [9] Cheng J. Modification and application of particle swarm optimization algorithm based on biological behavior mechanism [Ph.D. Thesis]. Guangzhou: South China University of Technology, 2014 (in Chinese with English abstract).
- [10] Li BY, Xiao YS, Wu QD. Hybrid algorithm based on particle swarm optimization for solving constrained optimization problems. *Control and Decision*, 2004,19(7):804–807 (in Chinese with English abstract). [doi: 10.3321/j.issn:1001-0920.2004.07.019]
- [11] Tao XM, Liu FR, Liu Y, Tong ZJ. Multi-scale cooperative mutation particle swarm optimization algorithm. *Ruan Jian Xue Bao/Journal of Software*, 2012,23(7):1805–1815 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/4128.htm> [doi: 10.3724/SP.J.1001.2012.04128]

- [12] Xia XW, Liu JN, Gao KF, Li YX, Zeng H. Particle swarm optimization algorithm with reverse-learning and local-learning behavior. *Chinese Journal of Computers*, 2015,38(7):1397–1407 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2015.01397]
- [13] Wen T, Sheng GJ, Guo Q, Li YQ. Web service composition based on modified particle swarm optimization. *Chinese Journal of Computers*, 2013,36(5):1032–1046 (in Chinese with English abstract). [doi: 10.3724/SP.J.1016.2013.01031]
- [14] Zhao YL, Guo BL, Wu XX, Wang P. Image reconstruction algorithm for ECT based on dual particle swarm collaborative optimization. *Journal of Computer Research and Development*, 2014,51(9):2094–2100 (in Chinese with English abstract). [doi: 10.7544/j.issn1000-1239.2014.20131006]
- [15] Jiang SJ, Wang LS, Xue M, Zhang YM, Yu Q, Yao HR. Test case generation based on combination of schema using particle swarm optimization. *Ruan Jian Xue Bao/ Journal of Software*, 2016,27(4):785–801 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/4966.htm> [doi: 10.13328/j.cnki.jos.004966]
- [16] Shi JF, Li L, Lv JQ. Skip-glide trajectory optimization based on chaotic particle swarm optimization algorithm. *Aerospace Control*, 2017,35(4):20–25 (in Chinese with English abstract). [doi: 10.16804/j.cnki.issn1006-3242.2017.04.004]
- [17] Zhou RJ, Li SJ, Li HY, Kang XW, Liu LP, Zhou SY. Space particle swarm optimization algorithm and its application in environmental & economic load distribution of power system. *Electric Power Automation Equipment*, 2014,34(9):7–12 (in Chinese with English abstract). [doi: 10.3969/j.issn.1006-6047.2014.09.002]
- [18] Ma D, Zhuang L, Lan JL. Discrete particle swarm optimization algorithm for service chain mapping problem. *Journal of Chinese Computer System*, 2017,38(8):1811–1817 (in Chinese with English abstract). [doi: 10.3969/j.issn.1000-1220.2017.08.028]
- [19] Peng Y, Peng XY, Liu ZQ. Statistic analysis on parameter efficiency of particle swarm optimization. *Acta Electronica Sinica*, 2004,32(2):209–213 (in Chinese with English abstract). [doi: 10.3321/j.issn:0372-2112.2004.02.008]
- [20] Wang L, Liu B. *Particle Swarm Optimization and Scheduling Algorithms*. Beijing: Tsinghua University Press, 2008 (in Chinese).
- [21] Dai WZ, Yang XL. Particle swarm optimization algorithm based on inertia weight logarithmic decreasing. *Computer Engineering and Applications*, 2015,51(17):14–19 (in Chinese with English abstract). [doi: 10.3778/j.issn.1002-8331.1412-0259]
- [22] Lin W, Liu PX. Parameter estimation for time-varying system based on combinatorial PSO. In: Sben WM, ed. *Proc. of the Information Technology for Balanced Manufacturing Systems*. Springer-Verlag, 2006. 357–368.
- [23] Chauhan P, Deep K, Pant M. Novel inertia weight strategies for particle swarm optimization. *Memetic Computing*, 2013,5(3): 229–251. [doi: 10.1007/s12293-013-0111-9]
- [24] Li XJ, Xu J, Zhu EZ, Zhang YW. A novel computation method for adaptive inertia weight of task scheduling algorithm. *Journal of Computer Research and Development*, 2016,53(9):1990–1999 (in Chinese with English abstract).
- [25] Lu ZY. Some improvements and applications of particle swarm optimization algorithm [MS. Thesis]. Xuzhou: China University of Mining and Technology, 2014 (in Chinese with English abstract).
- [26] Zhou XY, Wu ZJ, Wang MW. Artificial bee colony algorithm based on orthogonal experimental design. *Ruan Jian Xue Bao/ Journal of Software*, 2015,26(9): 2167–2190 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/4800.htm> [doi: 10.13328/j.cnki.jos.004800]
- [27] Du WB, Ying W, Yan G, Zhu YB, Cao XB. Heterogeneous strategy particle swarm optimization. *IEEE Trans. on Circuits & Systems II Express Briefs*, 2017,64(4):467–471. [doi: 10.1109/TCSII.2016.2595597]
- [28] Zhang LP. The theorem and practice upon the particle swarm optimization algorithm [Ph.D. Thesis]. Hangzhou: Institute of Material Science and Chemical Engineering Zhejiang University, 2005 (in Chinese with English abstract).
- [29] Kennedy J. Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance. In: *Proc. of the IEEE Congress on Evolutionary Computation (CEC'99)*, Vol.3. New York: IEEE Press, 1999. 1931–1938. [doi: 10.1109/cec.1999.785509]
- [30] Ni QJ, Zhang ZZ, Wang QQ, Xing CH. Dynamic probabilistic particle swarm optimization based on varying multi-cluster structure. *Ruan Jian Xue Bao/ Journal of Software*, 2009,20(2):339–349 (in Chinese with English abstract). <http://www.jos.org.cn/1000-9825/3237.htm> [doi: 10.3724/SP.J.1001.2009.03237]
- [31] Wang DF, Meng L, Zhao WJ. Improved bare bones particle swarm optimization with adaptive search center. *Chinese Journal of Computers*, 2016,39(12):2652–2667 (in Chinese with English abstract). [doi: 10.11897/SP.J.1016.2016.02652]



- [32] He QY, Han CJ. Improved particle swarm optimization algorithm with disturbance term. *Computer Engineering and Applications*, 2007,43(7):84–86 (in Chinese with English abstract). [doi: 10.3321/j.issn:1002-8331.2007.07.026]
- [33] Liu Y. Improvements and applications of particle swarm optimization algorithm [Ph.D. Thesis]. Xi'an: Xidian University, 2013 (in Chinese with English abstract).

#### 附中文参考文献:

- [4] 刘丽珏,蔡自兴.基于克隆选择的粒子群优化算法. *小型微型计算机系统*,2006,27(9):1708–1710. [doi: 10.3969/j.issn.1000-1220.2006.09.025]
- [5] 李笠,王万良,徐新黎,李伟琨.基于网格排序的多目标粒子群优化算法. *计算机研究与发展*,2017,54(5):1012–1023. [doi: 10.7544/issn1000-1239.2017.20160074]
- [6] 胡旺,Yen GG,张鑫.基于 Pareto 熵的多目标粒子群优化算法. *软件学报*,2014,25(5):1025–1050. <http://www.jos.org.cn/1000-9825/4496.htm> [doi: 10.13328/j.cnki.jos.004496]
- [8] 张艳梅,姜淑娟,陈若玉,王兴亚,张妙.基于粒子群优化算法的类集成测试序列确定方法. *计算机学报*,2016,41(4):931–945. [doi: 10.11897/SP.J.1016.2018.00931]
- [9] 程军.基于生物行为机制的粒子群优化算法改进及应用[博士学位论文].广州:华南理工大学,2014.
- [10] 李炳宇,萧蕴诗,吴启迪.一种基于粒子群算法求解约束优化问题的混合算法. *控制与决策*,2004,19(7):804–807. [doi: 10.3321/j.issn:1001-0920.2004.07.019]
- [11] 陶新民,刘福荣,刘玉,童智靖.一种多尺度协同变异的粒子群优化算法. *软件学报*,2012,23(7):1805–1815. <http://www.jos.org.cn/1000-9825/4128.htm> [doi: 10.3724/SP.J.1001.2012.04128]
- [12] 夏学文,刘经南,高柯夫,李元香,曾辉.具备反向学习和局部学习能力的粒子群算法. *计算机学报*,2015,38(7):1397–1407. [doi: 10.11897/SP.J.1016.2015.01397]
- [13] 温涛,盛国军,郭权,李迎秋.基于改进粒子群算法的 Web 服务组合. *计算机学报*,2013,36(5):1032–1046. [doi: 10.3724/SP.J.1016.2013.01031]
- [14] 赵玉磊,郭宝龙,吴宪祥,王湃.基于双粒子群协同优化的 ECT 图像重建算法. *计算机研究与发展*,2014,51(9):2094–2100. [doi: 10.7544/issn1000-1239.2014.20131006]
- [15] 姜淑娟,王令赛,薛猛,张艳梅,于巧,姚慧冉.基于模式组合的粒子群优化测试用例生成方法. *软件学报*,2016,27(4):785–801. <http://www.jos.org.cn/1000-9825/4966.htm> [doi: 10.13328/j.cnki.jos.004966]
- [16] 施健峰,李伶,吕建强.基于混沌粒子群算法的跳跃-滑翔轨迹优化. *航天控制*,2017,35(4):20–25. [doi: 10.16804/j.cnki.issn1006-3242.2017.04.004]
- [17] 周任军,李绍金,李红英,康信文,刘乐平,周胜瑜.空间粒子群优化算法及其在电力系统环保经济负荷分配中的应用. *电力自动化设备*,2014,34(9):7–12. [doi: 10.3969/j.issn.1006-6047.2014.09.002]
- [18] 马丁,庄雷,兰巨龙.一种求解服务链映射问题的离散粒子群优化算法. *小型微型计算机系统*,2017,38(8):1811–1817. [doi: 10.3969/j.issn.1000-1220.2017.08.028]
- [19] 彭宇,彭喜元,刘兆庆.微粒群算法参数效能的统计分析. *电子学报*,2004,32(2):209–213. [doi: 10.3321/j.issn:0372-2112.2004.02.008]
- [20] 王凌,刘波.微粒群优化与调度算法.北京:清华大学出版社,2008.
- [21] 戴文智,杨新乐.基于惯性权重对数递减的粒子群优化算法. *计算机工程与应用*,2015,51(17):14–19. [doi: 10.3778/j.issn.1002-8331.1412-0259]
- [24] 李学俊,徐佳,朱二周,张以文.任务调度算法中新的自适应惯性权重计算方法. *计算机研究与发展*,2016,53(9):1990–1999. [doi: 10.7544/issn1000-1239.2016.20151175]
- [25] 鲁姝颖.粒子群优化算法的几种改进算法及应用[硕士学位论文].徐州:中国矿业大学,2014.
- [26] 周新宇,吴志健,王明文.基于正交实验设计的人工蜂群算法. *软件学报*,2015,26(9):2167–2190. <http://www.jos.org.cn/1000-9825/4800.htm> [doi: 10.13328/j.cnki.jos.004800]
- [28] 张丽平.粒子群优化算法的理论与实践[博士学位论文].杭州:浙江大学材料与化学工程学院,2005.
- [30] 倪剑剑,张志政,王蓁蓁,邢汉承.一种基于可变多簇结构的动态概率粒子群优化算法. *软件学报*,2009,20(2):339–349. <http://www.jos.org.cn/1000-9825/3237.htm> [doi: 10.3724/SP.J.1001.2009.03237]

- [31] 王东风,孟丽,赵文杰.基于自适应搜索中心的骨干粒子群算法.计算机学报,2016,39(12):2652-2667. [doi: 10.11897/SP.J.1016.2016.02652]
- [32] 何庆元,韩传久.带有扰动项的改进粒子群算法.计算机工程与应用,2007,43(7):84-86. [doi: 10.3321/j.issn:1002-8331.2007.07.026]
- [33] 刘逸.粒子群优化算法的改进及应用研究[博士学位论文].西安:西安电子科技大学,2013.



徐利锋(1983—),男,浙江临安人,博士,讲师,CCF 专业会员,主要研究领域为虚拟仿真,优化算法,生物数据分析.



黄祖胜(1994—),男,学士,主要研究领域为优化算法,植物虚拟模型.



杨中柱(1992—),男,学士,主要研究领域为虚拟可视化建模,优化算法.



丁维龙(1973—),男,博士,教授,博士生导师,CCF 专业会员,主要研究领域为虚拟仿真,优化算法与智能系统.

www.jos.org.cn

www.jos.org.cn