

基于非均匀变异和多阶段扰动的粒子群优化算法

赵新超¹⁾ 刘国莅²⁾ 刘虎球³⁾ 赵国帅⁴⁾

¹⁾(北京邮电大学理学院 北京 100876)

²⁾(北京大学信息科学技术学院 北京 100871)

³⁾(清华大学计算机科学与技术系 北京 100084)

⁴⁾(北京邮电大学网络技术研究院 北京 100876)

摘 要 该文提出一种基于非均匀变异和多阶段扰动的粒子群优化算法,并对算法的搜索性能进行了一般性分析.首先,在算法执行的不同阶段利用对当前最优解施加大小不同的邻域扰动操作,很好地增加了群体多样性,提高了跳出局部陷阱的概率,同时加强了对当前最优解邻域内的精细搜索;其次,在粒子群优化算法中引入非均匀变异运算,并依据非均匀变异运算规律适应性调整解向量的搜索步长.算法性能分析表明,本算法较好地兼顾了群体优化算法的多样性和精英学习强度之间的平衡问题.数值实验上,首先用12个经典测试函数,验证该文提出的几种新措施的有效性与互助性;其次,针对30维和50维的CEC2005测试函数集,所提算法NmP3PSO与经典算法wFIPS、CLPSO和OLPSO做了大量的仿真实验,结果表明该文提出的算法表现出富有竞争力的性能和稳定性.

关键词 粒子群优化;非均匀变异;多阶段扰动;群体多样性

中图法分类号 TP302 **DOI号** 10.3724/SP.J.1016.2014.02058

Particle Swarm Optimization Algorithm Based on Non-Uniform Mutation and Multiple Stages Perturbation

ZHAO Xin-Chao¹⁾ LIU Guo-Li²⁾ LIU Hu-Qiu³⁾ ZHAO Guo-Shuai⁴⁾

¹⁾(School of Science, Beijing University of Posts and Telecommunications, Beijing 100876)

²⁾(School of Electronics Engineering and Computer Science, Peking University, Beijing 100871)

³⁾(Department of Computer Science and Technology, Tsinghua University, Beijing 100084)

⁴⁾(Institute of Network Technology, Beijing University of Posts and Telecommunications, Beijing 100876)

Abstract A new Particle Swarm Optimization (PSO) algorithm is proposed based on non-uniform mutation and multiple stages perturbation. Its search mechanism is also analyzed. Firstly, multiple stages perturbation operation with different radii is executed at different stages of algorithm. It diversifies the particle population and increases the probability of escaping from local trap. It also enhances the fine search at the neighborhood of the current best solution. Secondly, non-uniform mutation operator is introduced into PSO and the proposed algorithm adaptively adjusts the step size of solution vectors with non-uniform mutation operation. The performance analysis indicates that the proposed algorithm deals well with the balance between population diversity and learning intension from elitists. Twelve classical benchmarks are firstly used to verify the validity and the cooperation of the proposed strategies. Then the comparisons with the state-of-the-art evolutionary algorithms (wFIPS, CLPSO and OLPSO) are made on the shifted and rotated benchmarks from CEC2005 with cases of 30 and 50 dimensions. Experimental results indicate the competitive performance and stability of the proposed algorithm.

Keywords particle swarm algorithm; non-uniform mutation; multiple stages perturbation; population diversity

1 引言

粒子群优化算法^[1-2] (Particle Swarm Optimization Algorithm, PSO) 是对鸟群捕食行为的模拟, 由 Eberhart 和 Kennedy 提出的一种群智能优化算法. PSO 简单易行, 没有过多的参数需要调整, 目前已广泛应用于函数优化、神经网络训练、模糊系统控制以及其他相关的工程应用领域^[3-4]. 然而, 标准粒子群 PSO 算法在进化过程中, 由于收敛速度快而导致种群多样性的快速降低, 容易导致算法出现早熟收敛等问题^[5]. 在传统 PSO 算法中, 所有粒子仅仅向自身和邻域的历史最佳位置学习, 而没有向邻域内其他个体学习 (即使这些个体很优秀), 造成启发式信息资源和计算资源的无谓浪费, 甚至由此而陷入局部最优. Mendes 等人^[6] 提出了全知粒子群算法 (Fully Informed Particle Swarm, FIPS), 在 FIPS 中, 每个粒子除了自身和邻域最佳历史位置外, 还学习邻域内其他粒子的成功经验; Liang 等人^[7] 提出了一种既可以进行 n -维空间搜索、又能在不同维上选择不同学习对象的新学习策略, 称为全面学习 PSO (Comprehensive Learning Particle Swarm Optimizer, CLPSO), CLPSO 的每个粒子都随机地向自身或其它粒子学习, 并且每一个粒子的每一维度会向不同的粒子学习, 该学习策略使得每个粒子拥有更多的学习对象, 可以在更大的潜在空间飞行, 从而有利于全局搜索; 詹志辉等人^[8] 通过对群体和粒子适应值分布的评价识别出算法当前所处的搜索状态 (Exploration, Exploitation, Convergence, Jumping out), 然后相应调整算法的关键参数和后续操作, 极大的提高了算法的性能表现; 为了改善粒子群算法的线性学习策略对历史经验利用不足的现状, 陈伟能等人^[9] 提出年龄因素和生命周期概念, 当群体领导者年龄变大时, 允许其他粒子挑战其领导权, 而领导者的领导权会随着其领导力的变化而相应调整, 以持续地维持群体的进化动力.

免疫优化算法^[10-13] 是一种借鉴生物免疫系统特性而形成的启发式搜索算法, 具有保持种群多样性、减缓算法陷入局部最优的特性. 刘丽珏和蔡自兴^[11] 利用免疫系统的克隆选择机制, 提出一种函数优化算法, 算法特点是: 在迭代过程中, 不仅抗体得到进

化, 同时建立变异向量集, 令变异向量同步进化, 协同工作, 从而达到优化的目的; 基于 Baldwin 效应, 公茂果等人^[14] 提出一种克隆选择算法, 包括 4 个运算算子: 克隆繁殖、Baldwinian 学习、超变异和克隆选择, Baldwinian 学习算子模拟免疫系统的学习机制有效的改变了搜索空间; Cai 等人^[15] 提出一种使用分布估计疫苗的 Memetic 克隆选择算法 MCSA-EDA, 为了克服传统克隆选择算法的不足, 在算法中构建了 3 个操作用来求解无约束二进制二次规划问题; 为了克服量子聚类问题陷入局部最优和处理大尺寸图像分割问题的瓶颈, Gou 等人^[16] 通过在 Multi-elitist 免疫克隆优化亲和度函数计算中嵌入 1 个潜在的进化公式和基于距离矩阵的聚类中心更新方式, 提出一种 Multi-Elitist 免疫克隆量子聚类算法 (ME-ICQC), 在该算法框架中, 精英群体是由高亲和度的个体组成, 并在进化过程中扮演支配角色.

非均匀变异演化算法^[17] 是一种能自动调整搜索步长的搜索算法, 使得在搜索过程中始终有逃出局部陷阱的可能, 可以有效克服群体算法的早熟现象, 非均匀变异运算的优良性能已经得到很好验证^[12, 17], 本文将非均匀变异运算引进粒子群算法, 并对算法的搜索性能进行一般性分析. 扰动粒子群算法^[18] 是一种基于可能性度量概念的新的粒子更新策略, 能够在保持对最优解搜索轨迹学习的同时增加一定程度的粒子群体多样性. 在粒子群算法研究中, 各种学习策略的设计与提出及其相应的研究分析^[19-21] 对促进粒子群算法的研究具有非常积极的作用.

本文第 2 节简单介绍粒子群算法; 第 3 节详细讲述本文提出的措施和算法构造与分析; 第 4 节分析验证算法新措施的有效性和互助性; 第 5 节针对 30 维 CEC2005 测试函数集, 与几种经典算法做数值实验对比与算法分析; 第 6 节是针对 50 维 CEC2005 测试函数集的数值实验对比与分析; 第 7 节给出文章的结论和进一步拟研究的问题.

2 粒子群算法

标准粒子群优化算法是一种群智能算法, 根据对环境的适应度将群体中的个体移动到较好的区域, 从而实现优化问题的求解. 粒子群算法将每个

个体看成 n 维搜索空间中的一个没有体积的粒子, 在搜索空间中以一定的速度飞行, 每个粒子通过跟踪两个“极值”来更新自己的速度和位置: 一个极值是粒子本身找到的最优位置, 记为 $pbest$; 另一个极值是全体粒子所找到的最佳位置, 记为 $gbest$.

n 维搜索空间中, 种群由 m 个粒子组成 $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_m\}$, 其中第 i 个粒子的位置记为 $\mathbf{x}_i = \{x_{i1}, x_{i2}, \dots, x_{in}\}^T$, 飞行速度为 $\mathbf{v}_i = \{v_{i1}, v_{i2}, \dots, v_{in}\}^T$, 该粒子对应的局部最优位置为 $\mathbf{p}_i = \{p_{i1}, p_{i2}, \dots, p_{in}\}^T$, 所有粒子找到的最优位置记为 $\mathbf{p}_g = \{p_{g1}, p_{g2}, \dots, p_{gn}\}^T$, 粒子 \mathbf{x}_i 的飞行速度和位置更新公式分别为

$$\mathbf{v}_{id}^{t+1} = \omega \mathbf{v}_{id}^t + c_1 r_1 (\mathbf{p}_{id}^t - \mathbf{x}_{id}^t) + c_2 r_2 (\mathbf{p}_{gd}^t - \mathbf{x}_{id}^t) \quad (1)$$

$$\mathbf{x}_{id}^{t+1} = \mathbf{x}_{id}^t + \mathbf{v}_{id}^t \quad (2)$$

其中 $d=1, 2, \dots, n$; $i=1, 2, \dots, m$, n 为搜索空间的维数; m 为种群规模; t 为当前进化代数; c_1, c_2 是加速系数; r_1, r_2 是 $[0, 1]$ 之间的均匀分布随机数; ω 为惯性权重, 它决定了粒子先前速度对当前速度的影响程度. 为了避免粒子飞出搜索空间, 速度 v_i 通常被限制在某个飞行区域 $[-v_{\max}, v_{\max}]$, $v_{\max} = k(UB - LB)$, 通常取 $k=0.5$.

在粒子群优化算法中, 最优粒子 $gbest$ 的性能和行为对算法的最终性能有重要的影响. 扰动粒子群算法^[18]是一种在当前最优位置 $gbest$ 附近的每一个维度上做一次正态随机扰动的粒子更新策略. 若算法在搜索过程中找到的最优解是局部最优解, 则很容易陷入局部最优邻域, 导致种群多样性降低, 出现“进化停滞”现象. 通过对当前最优位置的随机扰动有可能使算法跳出当前局部陷阱, 从而在保证所有粒子精英学习机制的前提下, 又有效的保持了种群的多样性, 进而提高了搜索到全局最优解的概率.

3 混合粒子群算法 NmP3PSO

3.1 非均匀变异算子

变异是使得群体进化能够实现的关键操作, 非均匀变异运算^[17]将克隆复制产生的新粒子进行不同幅度的变异, 使粒子得以不断进化. 其工作原理如下: 假设对粒子 $\mathbf{x}_i = \{x_{i1}, x_{i2}, \dots, x_{id}, \dots, x_{in}\}^T$ 的第 d 个分量执行变异运算, 而 x_{id} 的下界和上界分别记为 LB 和 UB , 则变异后的分量:

$$x'_{id} = \begin{cases} x_{id} + \Delta(t, UB - x_{id}), & r < 0.5 \\ x_{id} - \Delta(t, x_{id} - LB), & r \geq 0.5 \end{cases} \quad (3)$$

其中: $\Delta(t, y) = y \cdot (1 - r^{(1 - \frac{t}{T})^b})$, 式(3)中 t 是循环变量, T 是最大的循环变量, r 是 $[0, 1]$ 间均匀产

生的随机数, b 是系统参数, 决定变异运算的非均匀度. 非均匀变异步长 $\Delta(t, y)$ 是一种自适应调节步长的变异算子, 程序前半段它几乎能在整个定义域中大范围搜索, 以期发现可能的潜在区域, 随着算法的进行, 搜索半径依概率减小, 到算法临近结束时仅在当前解的狭小邻域中搜索, 这样能够保证对最优解的准确定位而不会再从当前邻域中“逃逸”^[17].

在本文的数值实验中, 由于都是采用最大函数值计算次数的终止条件, 因此本文所有的数值实验对式(4)中的 t 和 T 调整为当前函数值计算次数和最大函数值计算次数.

3.2 最优粒子扰动策略及粒子更新

通常的粒子群算法群体的所有个体都向同一个全局最优粒子学习, 这样操作的优点是算法收敛速度较快, 不足之处在于学习强度过大容易导致算法群体的进化停滞. 与通常的粒子群算法不同, 基于扰动策略的粒子速度更新, 首先对全局最优粒子 $gbest$ 依据方差可调的正态随机分布进行扰动得到新的全局最优粒子 $pgbest$, 然后选定的待更新粒子向扰动后的全局最优粒子 $pgbest$ 学习, 此时的速度更新公式为

$$\mathbf{p}_{gd}^t = N(\mathbf{P}_{gd}^t, \sigma) \quad (4)$$

$$\mathbf{v}_{id}^{t+1} = \omega \mathbf{v}_{id}^t + c_1 r_1 (\mathbf{p}_{id}^t - \mathbf{x}_{id}^t) + c_2 r_2 (\mathbf{p}_{gd}^t - \mathbf{x}_{id}^t) \quad (5)$$

其中, \mathbf{p}_{gd}^t 表示第 t 代扰动后的最优粒子 $pgbest$ 的第 d 维分量, $pgbest$ 由正态分布 $N(\mathbf{p}_{gd}^t, \sigma)$ 产生, 其中 σ 表示相对于 $pgbest$ 的不确定度, σ 是对循环变量 t 的非增函数, 本文 σ 的更新方式如下

$$\sigma = \begin{cases} \sigma_1, & t < \alpha_1 T \\ \sigma_2, & \alpha_1 T < t < \alpha_2 T \\ \sigma_3, & t > \alpha_2 T \end{cases} \quad (6)$$

其中 $\sigma_1 > \sigma_2 > \sigma_3$ 表示正态扰动幅度的半径参数; α_1, α_2 是半径变化的控制参数, 且 $\alpha_1 < \alpha_2$, t 是当前函数值计算次数, T 是最大函数计算次数. 算法早期当 σ 较大时, 扰动算法群体会向全局最优解较大的邻域搜索学习, 提供简单高效的多路径并行引导搜索; 算法中后期当 σ 较小时, 扰动策略会保证算法群体在全局最优解较小的邻域进行搜索学习, 使得当前解几乎不再从较优区域跳出; 算法临近结束阶段 σ 很小, 甚至取值为零, 保证算法群体仅向全局最优解学习, 从而保证算法具有较好的收敛性.

3.3 优势选择

每一个解向量通过多阶段扰动导引及非均匀变异运算后, 保存父子两代个体, 为了保持算法进化的稳定性, 本文从父子两代的混合群体中选择适应度最好的个体作为新的当前粒子.

3.4 算法搜索的性能分析

本文在算法执行的不同阶段利用对当前全局最优解施加不同覆盖范围的邻域扰动思想,是为了保持精英学习操作存在的同时增加群体多样性,提高粒子跳出局部最优解的概率.下面就具体分析这种特性,为了分析的方便,假设对二维搜索空间的覆盖搜索范围进行分析,将式(1)和式(5)改写为向量形式(7)和式(8),其它相关的算法参数如前所述.

引入记号: $\mathbf{P}_- \mathbf{x} = c_1 (\mathbf{p}'_i - \mathbf{x}'_i)$, $\mathbf{G}_- \mathbf{x} = c_2 (\mathbf{p}'_g - \mathbf{x}'_i)$, $\mathbf{G}'_- \mathbf{x} = c_2 (\mathbf{p}'_g - \mathbf{x}'_i)$, $\mathbf{u}_1 = \omega \mathbf{v}'_i$, $\mathbf{u}_2 = r_1 \mathbf{P}_- \mathbf{x}$, $\mathbf{u}_3 = r_2 \mathbf{G}_- \mathbf{x}$, $\mathbf{u}'_3 = r_2 \mathbf{G}'_- \mathbf{x}$, 则方程(1)改写为如下向量形式,图1给出3个向量

$$\mathbf{v}'_i{}^{+1} = \omega \mathbf{v}'_i + r_1 \mathbf{P}_- \mathbf{x} + r_2 \mathbf{G}_- \mathbf{x} = \mathbf{u}_1 + \mathbf{u}_2 + \mathbf{u}_3 \quad (7)$$

式(5)改写为向量形式

$$\mathbf{v}'_i{}^{+1} = \omega \mathbf{v}'_i + r_1 \mathbf{P}_- \mathbf{x} + r_2 \mathbf{G}'_- \mathbf{x} = \mathbf{u}_1 + \mathbf{u}_2 + \mathbf{u}'_3 \quad (8)$$

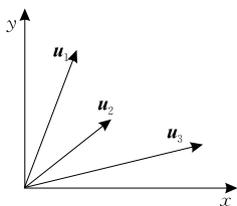


图1 二维空间中3个向量

3.4.1 粒子更新的随机参数对探测范围的影响

在向量式(7)和式(8)中,新向量都是3个向量的加权和.根据向量和的定义^[22],若参数 r_1, r_2 取常数,则速度的新向量 $\mathbf{v}'_i{}^{+1}$ 是一个确定的向量,因而更新后的新位置也是一个确定的位置向量,如图2中凸多边形右上方的端点;但由于 r_1, r_2 是0和1之间的均匀分布的随机数,因此这3个向量和所扫描的可能区域是这3个向量首尾相连构成的闭凸多边形区域,如图2中4个向量 $\mathbf{u}_1, \mathbf{u}_2, \mathbf{u}_3$ 和 \mathbf{v}'_{i+1} 所围城的凸多边形区域.

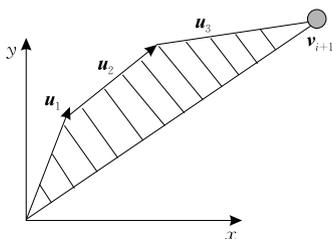


图2 二维空间中3个向量加权和覆盖的区域

图2同时也说明粒子群算法更新的优越之处,由于 r_1, r_2 的随机性,使得速度向量每次迭代都扫描了一个较大的搜索区域,从而算法的搜索引擎在一次迭代过程中可能的搜索区域也很大.同时,图2也可以解释在算法后期粒子群算法为什么容易陷入局

部最优的现象^[9],即算法后期,粒子的位置向量、它所记忆的局部最优位置和粒子群体记忆的全局最优向量很靠近,因而向量 $\mathbf{u}_2, \mathbf{u}_3$ 的模很小,甚至为零,即向最优解和历史记忆学习的强度很弱,因而速度向量覆盖的区域很小,因此新粒子几乎不可能跳出当前的局部陷阱.

进一步,也不难解释一些改进的粒子群算法的性能得到提升的原因^[18,23],即在算法后期,若适当增大差向量 \mathbf{u}_2 和 \mathbf{u}_3 ,适当的扩大粒子的搜索范围,使得陷入进化停滞的粒子依然保持跳出当前局部陷阱的能力,从而算法的有效搜索得以继续,算法就有可能找到更好的解,否则粒子的搜索引擎就是在原地踏步或陷入“Two Steps Forward, One Step Back”的窘境^[24],但在算法后期不提倡较大范围的扰动;同时,从图2中可以看出,若向量 \mathbf{u}_1 再有一定比例的伸缩,也可以达到类似的效果^[25-26],这就是速度更新中惯性系数以某种方式发生变化的原因所在.

3.4.2 加入全局最优扰动操作对探测范围的影响

加入对当前全局最优解一定范围的扰动操作,其他粒子都向扰动之后的最优粒子学习,可以有效改善算法的搜索性能,该算法的效果相当于图2中的向量 \mathbf{u}_3 不仅有一定范围的沿向量方向上的伸缩,而且有一定角度的旋转,因而算法中的其他粒子是向当前最优解所在的某个邻域进行学习,从而既保证了粒子群算法向精英个体学习的核心思想,又在一定程度上降低了向同一个精英个体过度学习的强度和速度,因而算法能够取得不错的效果^[18].

全局最优解扰动之后的向量加权和的效果如图3所示,图3是在图2的基础上,让最优解的差向量 \mathbf{u}_3 在成比例伸缩的基础上,再增加一定幅度的旋转操作,即沿着最优解可能的扰动的圆域的上下切线间做一定幅度的转动,这样扰动的好处是既保证粒子群向最优解区域学习,但减缓了学习强度,又一定程度上增加了群体多样性,因而取得的结果比较令人满意.

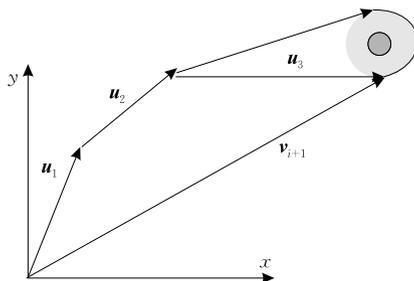


图3 二维空间中全局最优扰动后的3个向量加权和所覆盖的区域

图 3 也可以解释有些改进措施效果不理想的原因,即某些改进措施只是片面强调了群体多样性或精英学习强度的某一个方面,而忽略了两者的均衡问题。例如,如果算法后期进化停滞,单纯通过增加新个体而提高多样性的措施一般不会太理想,因为新加入的个体适应度一般较差,因此这些新个体增加的多样性在传递到精英个体和扰动邻域之前已经由于过学习操作而陷入局部陷阱,即这些新个体虽然增加了算法群体整体上的多样性,但是这种多样性对精英个体的进化基本不会发生什么影响,从而对算法性能的提高往往很有限,因此增加群体多样性的核心思想在于如何合理的增加精英解所在邻域的多样性。

3.5 算法分析与算法流程图

综合算法构造思想和以上的具体策略,现给出算法的相关分析和算法流程图。

3.5.1 算法计算复杂性分析

首先分析算法迭代一次所需要的基本操作:

计算当前最优解的扰动步长需要的基本运算是 $O(1)$; 粒子飞行速度和位置向量更新以及越界检测的复杂度是 $O(3mn)$; 计算函数适应值、更新局部和全局最优位置需要的基本运算是 $O(3m)$; 对当前最优粒子进行非均匀变异操作 $O(kn+k)$, 其中 k 是非均匀变异操作的次数; 因此算法一次迭代的时间复杂度是 $O(1) + O(3mn) + O(3m) + O(kn+k) = O(Cmn)$, 因此算法的整体复杂度是 $O(cmnT)$, 其中 m 是粒子群体规模, n 是问题的维数, T 是算法最大迭代次数, C 是常数。

3.5.2 与 CPSO 算法的对比分析

Tan 和 Xiao^[27] 提出克隆粒子群优化算法 (CPSO), CPSO 的主要思想是: 首先克隆 M 个连续进化代的最优粒子; 然后再对记忆的最优粒子施加一个高斯变异主导的变异操作(见式(9)), 有效加速了算法的收敛性; 然后所得新解和整个群体进行轮盘赌的选择操作, 选择概率主要考虑了基于函数值差异性的群体多样性, 增强了算法从当前局部陷阱逃逸的能力。

$$gbest = (1 - \mu) \otimes gbest \quad (9)$$

其中 μ 是均值为 0、方差为 1 的正态分布随机向量, $gbest$ 是记忆的最优解, \otimes 是点乘运算, 即两个相同维度的向量乘积结果是一个同样维度的向量, 结果向量的每一维度是两个相乘向量同一维度的乘积, 例如 $[2, 3] \otimes [4, 5] = [8, 15]$ 。

本文给出的算法 NmP3PSO 与 CPSO^[27] 有明

显的区别: (1) NmP3PSO 的主要操作之一是如式(4)~式(6)所示的粒子速度三阶段扰动更新策略, CPSO 没有; (2) CPSO 是连续记录 M 个进化代的最优解, 然后进行如式(9)所示的高斯正态随机变量主导的变异操作, 而 NmP3PSO 每一个进化代对当前最优解进行非均匀变异操作, 因此变异时间点和关键的变异方式都不一样; (3) CPSO 变异后的 M 个新解会与原解群体混合, 进行轮盘赌式的选择操作, CPSO 在计算混合群体中每一个解向量的浓度 (Concentration) 和选择概率时需要进行两次 $(m+M)^2$ 计算量的操作, 而 NmP3PSO 没有这些复杂运算, 其中 m 是群体规模, M 是连续记忆最优解的次数; (4) NmP3PSO 与 CPSO 在相同函数计算次数条件下, 仿真实验结果也有很大区别, 详见表 2。

3.5.3 算法流程图

算法流程图见图 4。

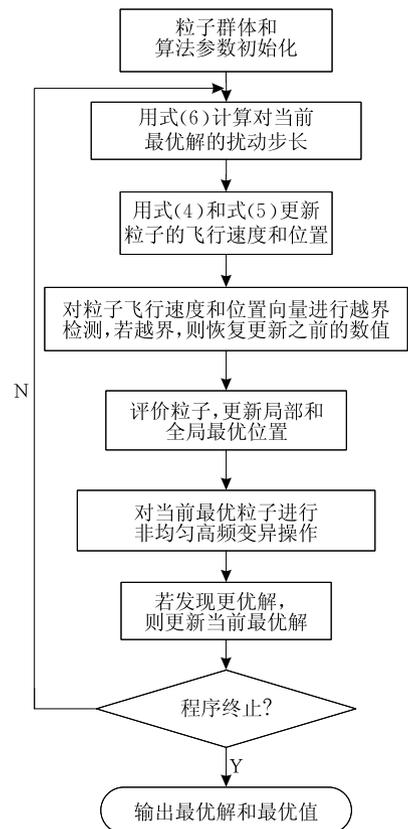


图 4 算法流程图

4 算法策略有效性和互助性分析

本节主要讨论算法策略的各自有效性和相互互助性, 算法对比实验的具体参数设置如下: 种群规模 $m=30$, 搜索空间维数 $n=30$, 惯性权值 $\omega=0.9$, 加

速常数 $c_1=c_2=0.5$, 控制参数 $\sigma_1=0.4, \sigma_2=0.001$, $\sigma_3=10^{-6}, \alpha_1=0.4, \alpha_2=0.7$. 需要说明的是, 作者对参数 σ 和 α 的数值实验表明, 算法性能对参数 σ 和 α 取值不敏感.

4.1 测试函数

为测试算法性能, 本文选用下列典型的基准测试函数^[28], 其中函数 $f_1 \sim f_7$ 为单峰函数, 函数 $f_8 \sim f_{12}$ 为多峰函数, 对每个函数独立运行 30 次, 然后对 30 次搜索到的最优结果求出统计意义上的最优值、中位值、平均值及方差, 并与基准 PSO 算法、P3PSO、NmPSO 和 CPSO^[27] 算法的运算结果进行对比分析.

4.2 算法构造策略的有效性和互助性

本文使用分阶段扰动策略和最优解的高频变异操作提升 PSO 算法的性能, 本节将在基本 PSO 算法的基础上逐步增加相应的构造措施, 采用表 1 所列的基准函数, 在相同函数值计算次数的条件下对比几种算法的计算性能. PSO 是基本 PSO 算法; P3PSO 是在 PSO 框架中融入对最优解的三阶段扰动策略引导粒子的速度更新; NmPSO 是在 PSO 框架中融入非均匀高频变异策略, 为了提高函数值计

算效用和节省计算代价, 本文只针对当前最优粒子施加非均匀变异操作; NmP3PSO 是结合 P3PSO 与 NmPSO 的综合策略算法, 即在 PSO 算法中融入最优粒子三阶段扰动引导策略和非均匀高频变异策略; CPSO^[27] 的主要策略是记忆 10 个连续进化代的最优粒子, 再施加变异操作, 加速了算法的收敛性, 然后所得新解和整个群体进行轮盘赌的选择操作, 选择概率主要考虑了基于函数值差异性的群体多样性, 增强了算法从当前局部陷阱逃逸的能力.

为了公平地比较不同的算法, 本部分实验所有算法采取相同的最大函数值计算次数: $maxFES=6$ 万, 其中算法 CPSO 的其它参数设置与文献^[27] 相同, 主要罗列如下: 群体规模 40; 惯性权值 ω 从 0.9 线性减少到 0.4, 稍有不同在于原文是随迭代次数线性减少, 本文是随函数值计算次数线性减少; 加速常数 $c_1=c_2=2$; 对飞行速度有越界检查; 为公平比较起见, 作者在 CPSO 算法中也加入粒子位置的越界检查, 即如果每一个粒子每一维度的新位置超过上界或下界, 则该维度恢复到原位置. 相关实验结果见表 2, 其中对每一个函数的最优算法的结果用黑体标示, 次最优算法的结果用斜体显示.

表 1 基准测试函数

基准函数	定义域	f_{min}
$f_1 = \sum_{i=1}^n x_i^2$	$[-100, 100]^n$	0
$f_2 = \sum_{i=1}^n x_i + \prod_{i=1}^n x_i $	$[-10, 10]^n$	0
$f_3 = \sum_{i=1}^n (\sum_{j=1}^i x_j)^2$	$[-100, 100]^n$	0
$f_4 = \max_i \{ x_i , 1 \leq i \leq n \}$	$[-100, 100]^n$	0
$f_5 = \sum_{i=1}^{n-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-30, 30]^n$	0
$f_6 = \sum_{i=1}^n [x_i + 0.5]^2$	$[-100, 100]^n$	0
$f_7 = \sum_{i=1}^n ix_i^i + \text{rand}[0, 1)$	$[-1.28, 1.28]^n$	0
$f_8 = \sum_{i=1}^n [x_i^2 - 10\cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^n$	0
$f_9 = -20\exp\left[-0.2\sqrt{\frac{1}{n}\sum_{i=1}^n x_i^2}\right] - \exp\left(\frac{1}{n}\sum_{i=1}^n \cos(2\pi x_i)\right) + 20 + e$	$[-32, 32]^n$	0
$f_{10} = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]^n$	0
$f_{11} = \frac{\pi}{n} \left\{ 10\sin^2(\pi y_1) + \sum_{i=1}^{n-1} (y_i - 1)^2 [1 + 10\sin^2(\pi y_{i+1})] + (y_n - 1)^2 \right\} + \sum_{i=1}^n u(x_i, 10, 100, 4), y_i = 1 + \frac{1}{4}(1 + x_i)$	$[-50, 50]^n$	0
$f_{12} = 0.1 \{ 10\sin^2(3\pi x_1) + \sum_{i=1}^{n-1} (x_i - 1)^2 [1 + 10\sin^2(\pi x_{i+1})] + (x_n - 1)^2 [1 + \sin^2(2\pi x_n)] \} + \sum_{i=1}^n u(x_i, 5, 100, 4)$	$[-50, 50]^n$	0

表 2 算法构造策略效用实验 (Mean±STD)

函数	PSO	P3PSO	NmPSO	NmP3PSO	CPSO
f_1	1.05e-2±2.36e-2	2.27e-11±3.34e-12	2.45e-10±2.11e-10	2.12e-11±2.60e-12	1.07e+2±6.66e+1
f_2	1.22e+1±8.31	1.13e-2±4.59e-2	3.34e-1±9.33e-1	1.35-3±1.58e-3	2.62±1.76
f_3	9.38e+2±8.25e+2	9.87e-3±1.06e-2	4.39e-3±3.16e-3	5.98e-3±7.89e-3	1.98e+4±6.50e+3
f_4	1.59±4.83e-1	3.64e-1±2.98e-1	4.78e-1±3.93e-1	2.21e-1±2.37e-1	2.86±5.48e-1
f_5	1.59e+2±1.99e+2	3.99e+1±6.32e+1	9.08e+1±8.64e+1	2.96e+1±2.84e+1	4.84±6±5.39e+6
f_6	1.70±1.24	0.50±0.68	0±0	0±0	0±0
f_7	2.53±2.95	1.62e-1±6.06e-2	2.51e-1±1.53e-1	7.75e-2±3.24e-2	5.05e+8±5.39e7
f_8	1.03e+2±2.74e+1	9.86e+1±2.14e+1	9.08e+1±1.74e+1	8.34e+1±2.15e+1	2.08e+2±2.67e+1
f_9	3.04±6.71e-1	2.96±7.08e-1	3.09±1.13	2.52±8.41e-1	4.24e-1±2.53e-1
f_{10}	1.97e-2±1.79e-2	1.57e-2±1.82e-2	1.21e-2±9.98e-2	7.87e-3±9.52e-3	6.51e-1±1.67e-1
f_{11}	1.10e+1±3.36	5.59±2.42	8.52±4.52	4.75±2.58	8.50e+9±1.17e+10
f_{12}	3.98e+1±1.26e+1	3.32e+1±1.08e+1	3.46e+1±9.11	2.85e+1±1.02e+1	3.42e+4±8.19e+4

表 2 的实验结果表明:(1)在相同函数值计算次数的条件下,本文所提算法 NmP3PSO 在 12 个测试函数中取得 10 个最优计算结果和 2 个次优结果, NmPSO 和 CPSO 各取得一个最优计算结果;(2)融合 3 阶段扰动策略的算法 P3PSO 在大部分测试函数上都明显优于 PSO 算法,在函数 f_8, f_9 上表现相当,取得 7 个测试函数计算的次优结果;(3)融合非均匀高频变异策略的算法 NmPSO 在大多数测试函数上都比 PSO 具有优势,只有在 f_9 上稍差于 PSO;(4)与前 3 个算法比较对象对比,融合两种策略的算法 NmP3PSO 在 10 个测试函数上都取得最优的结果,只有在函数 f_3 上稍差于算法 NmPSO,在函数 f_9 上差于 CPSO. 综上所述,在相同计算代价的条件下,本文提出的两种策略对算法性能的提升都能起到很积极的作用;其次两者的融合能够更进一步的大幅促进算法性能的提升,从而导致本文提出的算法能够取得更满意的结果.

5 使用 CEC2005 测试函数集与知名算法的对比实验与分析:30 维

为了分析验证算法的性能,本文引进 3 种知名算法 wFIPS^[6]、CLPSO^[7]、OLPSO^[29] 作为比较对象,对比分析算法 NmP3PSO 的性能表现,其中 wFIPS 是 FIPS(Fully Informed Particle Swarm)算法的改进版本,其中每一个邻居的贡献通过它自己的先前最优解来衡量^[6].

5.1 测试函数与参数设置

本组实验采取 CEC2005^[30] 提供的实参数优化函数,包括最优解偏移的(Shifted)、旋转的(Rotated)单峰和多峰函数,所选择的 3 组测试函数简单信息如下,详细信息请参看文献^[30],由于函数的搜索区域问题,函数 F7 需要做特殊的处理,因此本部分实验没有选择函数 F7.

(1) Unimodal Functions (5);

F1: Shifted Sphere Function;

F2: Shifted Schwefel's Problem 1.2;

F3: Shifted Rotated High Conditioned Elliptic Function;

F4: Shifted Schwefel's Problem 1.2 with Noise in Fitness;

F5: Schwefel's Problem 2.6 with Global Optimum on Bounds;

(2) Multimodal Functions (6);

F6: Shifted Rosenbrock's Function;

F8: Shifted Rotated Ackley's Function with Global Optimum on Bounds;

F9: Shifted Rastrigin's Function;

F10: Shifted Rotated Rastrigin's Function;

F11: Shifted Rotated Weierstrass Function;

F12: Schwefel's Problem 2.13;

(3) Expanded Multimodal Functions (2);

F13: Expanded Extended Griewank's plus Rosenbrock's Function (F8F2);

F14: Shifted Rotated Expanded Scaffer's F6.

本节数值实验所有算法的终止条件采取相同的最大函数值计算次数: $maxFES = 10$ 万, 算法 wFIPS^[6]、CLPSO^[7]、OLPSO^[29] 的所有其他参数和操作都与原文献相同, NmP3PSO 的所有参数与第 4 节相同.

5.2 4 种算法数值实验对比

算法 wFIPS、CLPSO、OLPSO 和 NmP3PSO 在 25 次独立运行的数值结果统计见表 3, 该统计结果包括 25 次独立运行最终结果的平均最优值 Mean 和标准差 STD, 每一个函数获得的 4 组结果中, 最优的计算结果用黑体标示. 表 3 最后一行“w/l”表示“win”或“lose”, 用来衡量算法 NmP3PSO 相比于相应算法“赢”或“输”的函数个数.

表 3 4 种算法数值实验 25 次独立计算结果统计 (Mean±STD)

函数	wFIPS	CLPSO	OLPSO	NmP3PSO
F_1	$8.23e-7 \pm 4.98e-7$	$2.06e-7 \pm 9.10e-8$	$4.11e-17 \pm 3.67e-17$	$1.64e-11 \pm 2.81e-12$
F_2	$1.47e+3 \pm 3.65e+2$	$8.40e+3 \pm 1.28e+3$	$2.35e+5 \pm 2.33e+5$	$2.46e-7 \pm 5.16e-7$
F_3	$3.65e+7 \pm 1.03e+7$	$2.76e+7 \pm 6.75e+6$	$9.02e+6 \pm 4.31e+6$	$6.41e+5 \pm 2.90e+5$
F_4	$5.38e+3 \pm 2.97e+3$	$1.69e+4 \pm 3.32e+3$	$2.69e+4 \pm 1.13e+4$	$1.69e+4 \pm 2.07e+3$
F_5	$3.10e+3 \pm 4.08e+2$	$5.78e+3 \pm 4.22e+2$	$3.61e+3 \pm 6.34e+2$	$6.79e+3 \pm 2.37e+3$
F_6	$4.48e+1 \pm 3.28e+1$	$1.21e+2 \pm 4.17e+1$	$1.79e+2 \pm 2.76e+2$	$2.06e+1 \pm 1.88e+1$
F_8	$2.09e+1 \pm 2.68e-2$	$2.09e+1 \pm 4.11e-2$	$2.06e+1 \pm 1.94e-1$	$2.04e+1 \pm 1.18e-1$
F_9	$9.04e+1 \pm 1.25e+1$	$1.91e-2 \pm 1.72e-2$	5.07 ± 2.59	$9.50e+1 \pm 2.45e+1$
F_{10}	$1.91e+2 \pm 1.04e+1$	$1.62e+2 \pm 2.77e+1$	$5.78e+1 \pm 1.68e+1$	$1.37e+2 \pm 3.06e+1$
F_{11}	$3.99e+1 \pm 1.39$	$3.01e+1 \pm 1.21$	$1.89e+1 \pm 5.32$	$3.02e+1 \pm 1.19$
F_{12}	$1.14e+5 \pm 2.93e+4$	$5.00e+4 \pm 1.26e+4$	$1.27e+4 \pm 1.51e+4$	$1.09e+5 \pm 1.24e+4$
F_{13}	$1.36e+1 \pm 1.24$	$4.08 \pm 4.74e-1$	$2.24 \pm 5.86e-1$	8.44 ± 3.32
F_{14}	$1.34e+1 \pm 1.63e-1$	$1.31e+1 \pm 2.07e-1$	$1.31e+1 \pm 3.14e-1$	$1.26e+1 \pm 4.74e-1$
w/l	11/2	9/4	6/7	

从表 3 数值结果看出:(1)在 13 个测试函数中, NmP3PSO、OLPSO、CLPSO 和 wFIPS 分别取得 6 个、5 个、1 个和 1 个最优结果;(2)算法 NmP3PSO 的数值实验结果在 11 个函数上超过 wFIPS,在 2 个函数上输给 wFIPS;算法 NmP3PSO 的数值实验结果在 9 个函数上超过 CLPSO,在 4 个函数上输给 CLPSO,因此在 CEC2005 这些复杂测试函数上, NmP3PSO 比 wFIPS 和 CLPSO 有更好的性能;(3)算法 NmP3PSO 的数值实验结果在 6 个函数上优于 OLPSO,在 7 个函数上输给 OLPSO,因此可以看出在 CEC2005 这些复杂测试函数上 NmP3PSO 的表现与 OLPSO 基本相当.总之,在相同的最大函数值计算次数条件下,在 CEC2005 这些带 Shifted 和 Rotated 属性的复杂测试函数上,本文提出的改进粒子群算法框架 NmP3PSO 与基于正交学习的算法 OLPSO 基本相当,但比带权重的全信息粒子群算法 wFIPS 和全学习机制的粒子群算法 CLPSO 取得更好的性能.

5.3 4 种算法在线性能的进化趋势对比

为考查算法的平均进化趋势和综合在线性能,图 5 给出 4 个算法在 25 次独立运行中在线性能对比分析,其中 x 轴是函数值计算次数, y 轴是 4 个算法在 25 次独立运行中每一个进化代最优适应值的平均值.从图 5(a)、图 5(b)和图 5(c)这 3 类测试函数中各选 2 个,即函数 $F_2, F_3; F_6, F_{10}$ 和 F_{13}, F_{14} 为代表.

从图 5 看出,与数值实验结果类似,几个算法的进化趋势也是互有优劣,但总体而言,OLPSO 与 NmP3PSO 进化趋势表现最好,其次是算法 CLPSO,算法 wFIPS 对测试函数的进化趋势表现相对差些.

6 使用 CEC2005 测试函数集与知名算法的对比实验与分析:50 维

为了进一步分析验证算法的性能,针对 50 维情形的 CEC2005^[30] 实参数优化函数,本文所提算法 NmP3PSO 将与 3 种知名 PSO 算法:wFIPS^[6]、CLPSO^[7]、OLPSO^[29] 做进一步的仿真对比分析.除了搜索空间维数 $n=50$ 外,本组实验中算法的参数设置与前面两组实验的参数相同,包括最大函数值计算次数 $maxFES=10$ 万.

6.1 4 种算法数值实验对比

算法 wFIPS、CLPSO、OLPSO 和 NmP3PSO 在 25 次独立运行中的数值结果统计见表 4,该统计结果包括 25 次独立运行最终结果的平均最优值、Mean 和标准差 STD,每一个函数获得的 4 组结果中,最优的计算结果用黑体标示.表 4 最后一行“ w/l ”表示“win”或“lose”,用来衡量算法 NmP3PSO 相比于相应算法“赢”或“输”的函数的个数.

从表 4 仿真结果看:(1)所有测试函数的 Mean 和 STD 数据项相较于 30 维测试函数都有一定程度的下降;(2)在 13 个测试函数中,NmP3PSO、OLPSO、CLPSO 分别取得 7 个、4 个和 2 个测试函数上的最优结果;(3)算法 NmP3PSO 的数值实验结果在 11 个函数上超过 wFIPS,在 9 个测试函数上超过 CLPSO,在 8 个函数上超过 OLPSO,由此可以看出,针对 50 维情形的 CEC2005 测试函数集,NmP3PSO 的性能要优于 wFIPS 和 CLPSO;(4)NmP3PSO 的表现要稍优于 OLPSO,NmP3PSO 在单峰函数和部分多峰函数上优于 OLPSO,但 OLPSO 在部分多峰函数上要优于 NmP3PSO,因此 OLPSO 和 NmP3PSO

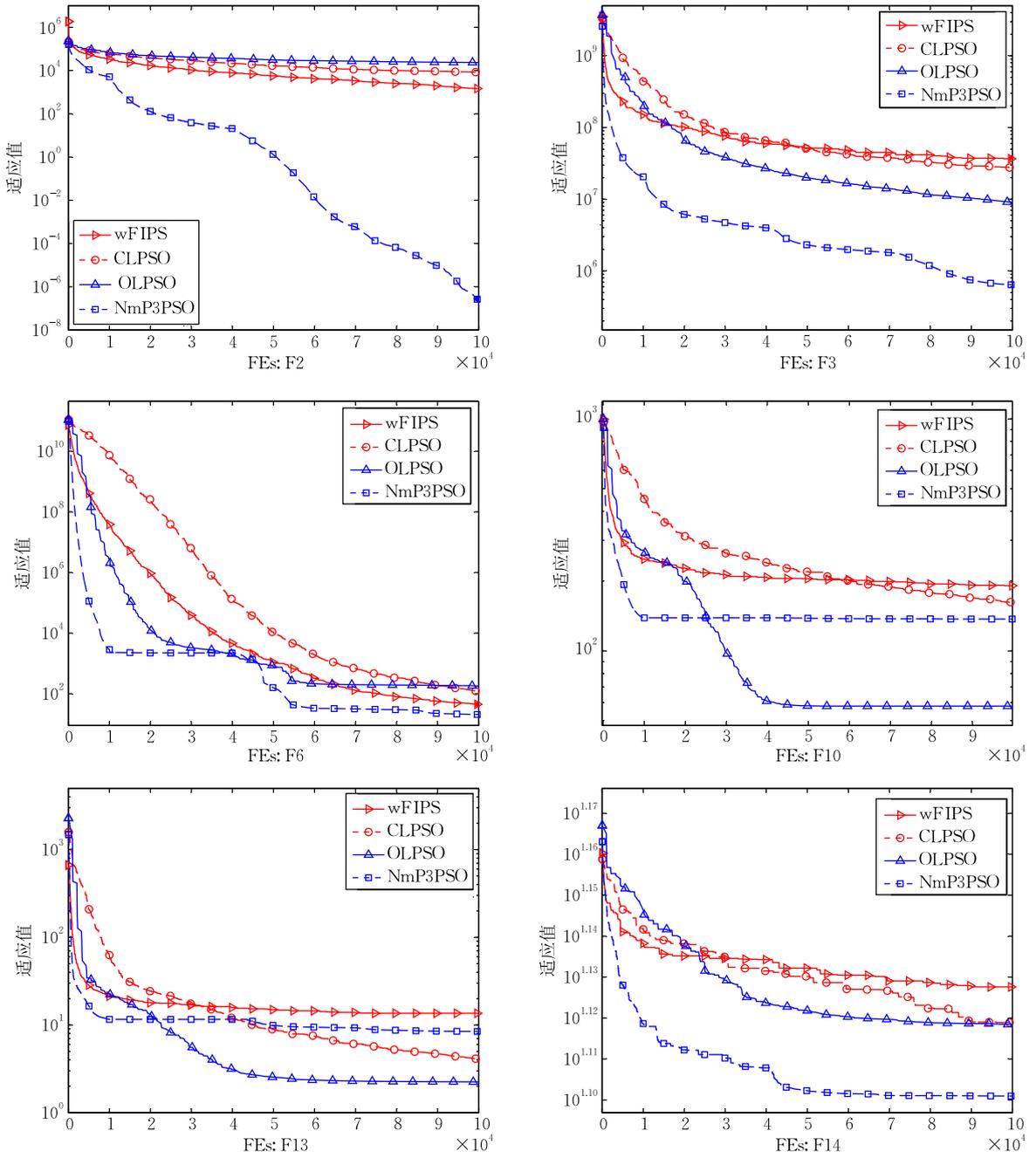


图 5 wFIPS, CLPSO, OLPSO 与 NmP3PSO 进化趋势(30 维函数)

表 4 4 种算法数值实验 25 次独立计算结果统计 (Mean±STD)

函数	wFIPS	CLPSO	OLPSO	NmP3PSO
F_1	$2.17e-1 \pm 5.97e-2$	$1.31e-2 \pm 3.87e-3$	$2.51e-2 \pm 5.29e-3$	$5.24e-11 \pm 6.83e-12$
F_2	$3.86e+4 \pm 8.68e+3$	$4.26e+4 \pm 4.59e+4$	$1.18e+5 \pm 1.42e+4$	1.47 ± 1.49
F_3	$1.98e+8 \pm 3.12e+7$	$1.59e+7 \pm 3.39e+7$	$1.33e+8 \pm 5.42e+7$	$3.69e+6 \pm 2.69e+6$
F_4	$6.36e+5 \pm 1.13e+4$	$6.98e+4 \pm 8.13e+3$	$1.24e+5 \pm 1.51e+4$	$6.48e+4 \pm 8.11e+4$
F_5	$9.25e+3 \pm 1.01e+3$	$1.59e+4 \pm 1.36e+3$	$9.17e+3 \pm 1.53e+3$	$1.63e+4 \pm 1.73e+3$
F_6	$4.26e+3 \pm 1.91e+3$	$1.94e+3 \pm 3.49e+2$	$3.22e+3 \pm 5.32e+3$	$8.48e+1 \pm 9.60e+1$
F_8	$2.12e+1 \pm 2.95e-2$	$2.12e+1 \pm 3.24e-2$	$2.13e+1 \pm 4.09e-2$	$2.07e+1 \pm 2.23e-1$
F_9	$2.50e+2 \pm 2.32e+1$	9.24 ± 3.18	$1.95e+1 \pm 6.84$	$1.90e+2 \pm 6.24$
F_{10}	$3.94e+2 \pm 2.20e+1$	$4.32e+2 \pm 1.68e+1$	$2.02e+2 \pm 6.57e+1$	$3.06e+2 \pm 6.27e+1$
F_{11}	$7.40e+1 \pm 1.39$	$5.85e+1 \pm 2.36$	$5.22e+1 \pm 1.04e+1$	$5.51e+1 \pm 4.75$
F_{12}	$7.25e+5 \pm 1.53e+5$	$2.98e+5 \pm 5.83e+4$	$1.33e+5 \pm 4.36e+4$	$9.05e+5 \pm 4.93e+5$
F_{13}	$3.28e+1 \pm 1.54$	9.94 ± 1.27	$1.55e+1 \pm 2.24$	$2.90e+1 \pm 9.84$
F_{14}	$2.33e+1 \pm 1.53e-1$	$2.31e+1 \pm 1.74e-1$	$2.34e+1 \pm 3.01e-1$	$2.19e+1 \pm 4.51e-1$
w/l	11/2	9/4	8/5	

在不同的测试函数上互有优劣. 总之, 在相同的函数值计算次数条件下, 在 50 维 CEC2005 这些带 Shifted 和 Rotated 属性的复杂测试函数集上, 本文提出的改进粒子群算法 NmP3PSO 稍优于基于正交学习的算法 OLPSO, 比带权重的全信息粒子群算法 wFIPS 和全学习机制的粒子群算法 CLPSO 取得更好的性能表现.

6.2 4 种算法在线性能的进化趋势对比

为进一步考查算法的平均进化趋势和综合在线性能, 与上一组实验选取同样的 3 类测试函数的代表, 即函数 $F_2, F_3; F_6, F_{10}$ 和 F_{13}, F_{14} , 图 6 给出 4 个算法在 25 次独立运行中在线性能对比分析, 其中 x 轴是函数值计算次数, y 轴是 4 个算法在 25 次独立运行中每一个进化代最优适应值的平均值.

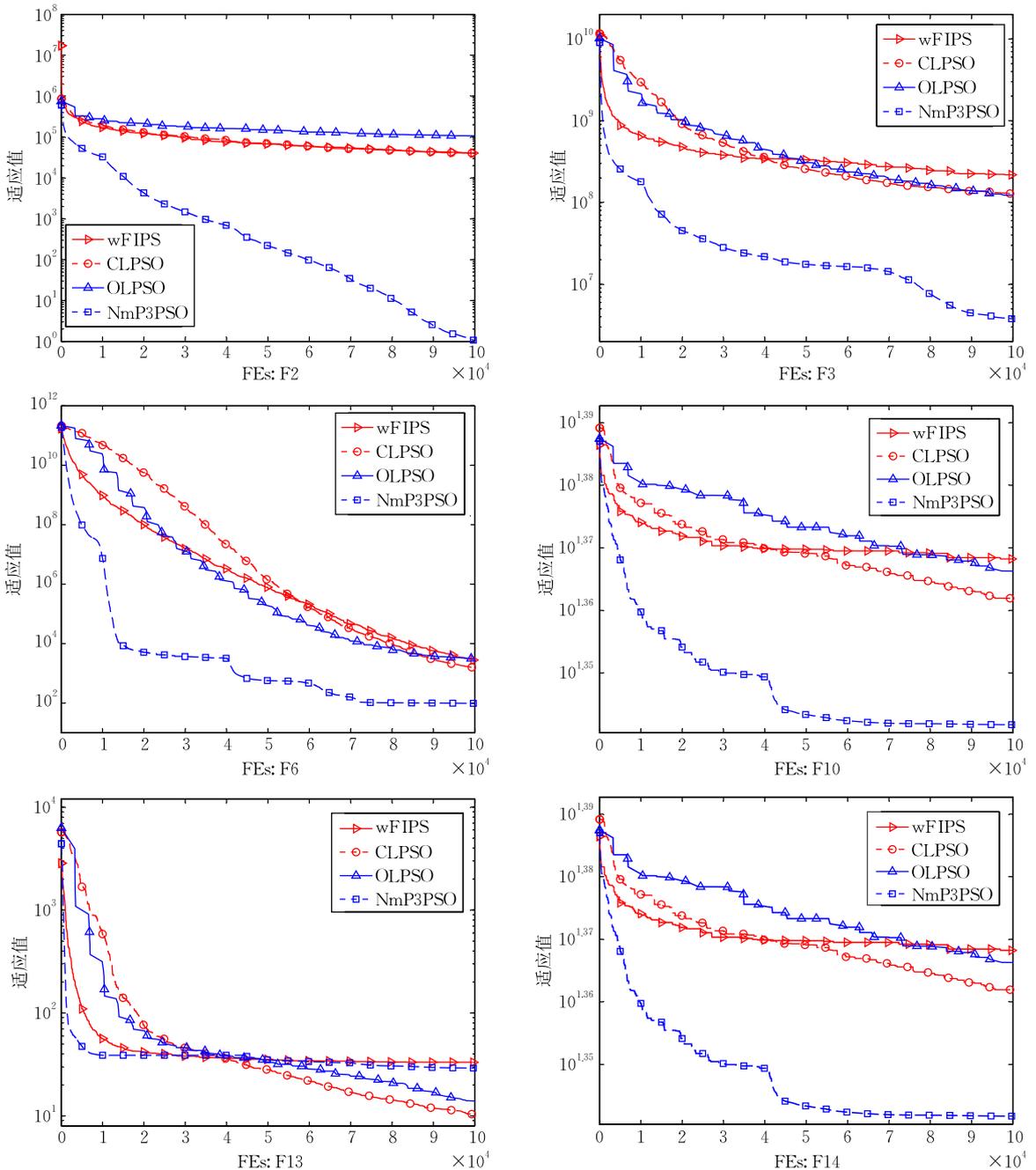


图 6 wFIPS, CLPSO, OLPSO 与 NmP3PSO 进化趋势(50 维函数)

从图 6 看出, 从测试函数的算法平均最优函数值的进化曲线来看, 本文提出的多阶段扰动策略在算法进化过程中有较为明显的表现, 尤其是对函数

F_2, F_3, F_6, F_{14} , 在扰动半径变小的函数值计算次数附近表现出明显的加速收敛属性; 几个经典 PSO 算法的进化趋势互有优劣, NmP3PSO 算法针对 50 维

带扰动和偏移的测试函数表现的进化趋势相对最好,其次是带正交学习策略的算法 OLPSO,再次是带全学习策略的 CLPSO,带权重全信息的算法 wFIPS 的进化趋势表现相对差些。

7 结 论

本文在融合非均匀变异和优秀粒子的多阶段扰动策略的基础上,提出一种混合粒子群优化算法,并对粒子群算法的搜索探测进行了一般性分析。非均匀变异算子适应性的调整搜索步长,同时考虑了算法群体的多样性和收敛性;算法群体的所有粒子向扰动之后的全局最优解学习,使得算法在保持粒子群精英学习机制的基础上,扩大算法群体搜索的有效范围,该思想旨在向最优解学习,但放缓了学习的强度和速度,从而较好保持全局探测和局部勘探之间的平衡;针对向扰动后的全局最优解学习策略的分析表明,本文所提算法是通过扩大全局最优解搜索邻域的思路来提高群体多样性的,只是适度减缓了精英学习的强度,从而避免过早陷入局部陷阱的问题。

第 1 组仿真实验表明,本文提出的几种措施是有效的,相互的组合是协作互补的;第 2 组是基于 30 维情形的 CEC2005 实参数优化的复杂函数集,与知名算法 wFIPS、CLPSO 和 OLPSO 的仿真对比表明,本文所提算法 NmP3PSO 的性能与新近的经典算法 OLPSO 基本相当,比经典粒子群优化算法 CLPSO 和 wFIPS 具有更优越的表现;第 3 组是基于 50 维情形的 CEC2005 实参数优化的复杂函数集,稍优于 OLPSO,比算法 CLPSO 和 wFIPS 具有较为明显的优势。

进一步的研究会探测大幅变异运算与该算法的融合和相互促进,比如 Cauchy 或 Levy 变异操作;高维优化或多目标优化问题的求解算法也是进一步有意义的研究问题。

致 谢 新加坡南洋理工大学 Suganthan 教授提供 CLPSO、wFIPS 算法 Matlab 源程序,中南大学王勇博士提供了 OLPSO 算法 Matlab 源程序,研究生林文桥(现在北京邮电大学信息光子学与光通信国家重点实验室攻读博士)帮助组合和撰写部分程序,审稿专家提出很多建设性的审稿意见,作者在此对以上帮助一并表示感谢。本文 2013 年 11 月份的重大修改是第一作者受国家留学基金委资助以访问学者

身份在英国埃塞克斯大学访问 ZHANG Qing-Fu 教授期间完成,2014 年 4 月份的修改是第一作者在英国伯明翰大学访问 YAO Xin 教授期间完成!

参 考 文 献

- [1] Kennedy J, Eberhart R C. Particle swarm optimization// Proceedings of the 1995 IEEE International Conference on Neural Networks. Perth, Australia, 1995: 1942-1948
- [2] Shi Y H, Eberhart R C. A modified particle swarm optimizer //Proceedings of the 1998 IEEE International Conference on Evolutionary Computation. Piscataway, USA, 1998: 67-73
- [3] Cui Guang-Zhao, Li Xiao-Guang, Zhang Xun-Cai, et al. The optimization of DNA encodings based on modified PSO/GA algorithm. Chinese Journal of Computers, 2010, 33(2): 311-316(in Chinese)
(崔光照, 李小广, 张勋才等. 基于改进的粒子群遗传算法的 DNA 编码序列优化. 计算机学报, 2010, 33(2): 311-316)
- [4] Zhang Chang-Sheng, Sun Ji-Gui, Ouyang Dan-Tong, Zhang Yong-Gang. A self-adaptive hybrid particle swarm optimization algorithm for flow shop scheduling problem. Chinese Journal of Computers, 2009, 32(11): 2137-2146 (in Chinese)
(张长胜, 孙吉贵, 欧阳丹彤, 张永刚. 求解车间调度问题的自适应混合粒子群算法. 计算机学报, 2009, 32(11): 2137-2146)
- [5] Ji Zhen, Liao Hui-Lian, Wu Qing-Hua. Particle Swarm Optimization and Its Applications. Beijing: Science Press, 2009(in Chinese)
(纪震, 廖惠连, 吴青华. 粒子群算法及应用. 北京: 科学出版社, 2009)
- [6] Mendes R, Kennedy J, Neves J. The Fully informed particle swarm: Simpler, maybe better. IEEE Transactions on Evolutionary Computation, 2004, 8(3): 204-210
- [7] Liang J J, Qin A K, Suganthan P N, Baskar S. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Transactions on Evolutionary Computation, 2006, 10(3): 281-295
- [8] Zhan Z H, Zhang J, Li Y, Chung H S H. Adaptive particle swarm optimization. IEEE Transactions on Systems, Man, Cybernetics; B, Cybernetics, 2009, 39(6): 1362-1381
- [9] Chen W N, Zhang J, Lin Y, et al. Particle swarm optimization with an aging leader and challengers. IEEE Transactions on Evolutionary Computation, 2013, 17(2): 241-258
- [10] Li Li, Li Hong-Qi, Xie Shao-Long, Li Xiong-Yan. Immune particle swarm optimization algorithms based on clone selection. Computer Science, 2008, 35(10): 253-255(in Chinese)
(李莉, 李洪奇, 谢邵龙, 李雄炎. 基于克隆选择的免疫粒子群优化算法. 计算机科学, 2008, 35(10): 253-255)

- [11] Liu Li-Jue, Cai Zi-Xing. Immunity clone algorithm with mutation co-evolution. *Control and Decision*, 2008, 23(6): 613-618(in Chinese)
(刘丽珏, 蔡自兴. 变异协同进化的免疫克隆算法. *控制与决策*, 2008, 23(6): 613-618)
- [12] Zhong Yan-Fei, Zhang Liang-Pei, Li Ping-Xiang. A clonal selection algorithm based on non-uniform adaptive mutation. *Geomatics and Information Science of Wuhan University*, 2009, 34(3): 308-312(in Chinese)
(钟燕飞, 张良培, 李平湘. 基于非一致性自适应变异的克隆选择算法. *武汉大学学报(信息科学版)*, 2009, 34(3): 308-312)
- [13] Jiao Li-Cheng, Du Hai-Feng, Liu Fang, Gong Mao-Guo. *Immune Optimization—Computing, Learning and Recognition*. Beijing: Science Press, 2006(in Chinese)
(焦李成, 杜海峰, 刘芳, 公茂果. 免疫优化——计算、学习与识别. 北京: 科学出版社, 2006)
- [14] Gong M G, Jiao L C, Zhang L N. Baldwinian learning in clonal selection algorithm for optimization. *Information Sciences*, 2010, 180(8): 1218-1236
- [15] Cai Y Q, Wang J H, Yin J, Zhou Y L. Memetic clonal selection algorithm with EDA vaccination for unconstrained binary quadratic programming problems. *Expert Systems with Applications*, 2011, 38(6): 7817-7827
- [16] Gou S P, Zhuang X, Li Y Y, et al. Multi-elitist immune clonal quantum clustering algorithm. *Neurocomputing*, 2013, 101: 275-289
- [17] Zhao X C, Gao X S, Hu Z C. Evolutionary programming based on non-uniform mutation. *Applied Mathematics and Computation*, 2007, 192(1): 1-11
- [18] Zhao X C. A perturbed particle swarm algorithm for numerical optimization. *Applied Soft Computing*, 2010, 10(1): 119-124
- [19] Huang H, Qin H, Hao Z F, Lim A. Example-based learning particle swarm optimization for continuous optimization. *Information Sciences*, 2012, 182(1): 125-138
- [20] Sabat S L, Ali L, Udgata S K. Integrated learning particle swarm optimizer for global optimization. *Applied Soft Computing*, 2011, 11(1): 574-584
- [21] Wang Y, Li B, Weise T, et al. Self-adaptive learning based particle swarm optimization. *Information Sciences*, 2011, 181(20): 4515-4538
- [22] Department of Mathematics of Tongji University. *Advanced Mathematics (II, 6th Edition)*. Beijing: Higher Education Press, 2008(in Chinese)
(同济大学数学系. 高等数学(下)(第6版). 北京: 高等教育出版社, 2008)
- [23] Van den Bergh F, Engelbrecht A P. A new locally convergent particle swarm optimizer//*Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics*. Yasmine Hammamet, Tunisia, 2002: 96-101
- [24] van den Bergh F, Engelbrecht A P. A cooperative approach to particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 2004, 8(3): 225-239
- [25] Eberhart R C, Shi Y H. Comparing inertia weights and constriction factors in particle swarm optimization//*Proceedings of IEEE Congress on Evolutionary Computation*. La Jolla, USA, 2000: 84-88
- [26] Eberhart R C, Shi Y H. Tracking and optimizing dynamic systems with particle swarms//*Proceedings of the IEEE Congress on Evolutionary Computation*. Seoul, Korea, 2001: 94-100
- [27] Tan Y, Xiao Z M. Clonal particle swarm optimization and its applications//*Proceedings of the 2007 IEEE International Congress on Evolutionary Computation*. Singapore, 2007: 2303-2309
- [28] Yao X, Liu Y, Lin G M. Evolutionary programming made faster. *IEEE Transactions on Evolutionary Computation*, 1999, 3(2): 82-102
- [29] Zhan Z H, Zhang J, Li Y, Shi Y H. Orthogonal learning particle swarm optimization. *IEEE Transactions on Evolutionary Computation*, 2011, 15(6): 832-847
- [30] 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. Nanyang Technological University, Singapore, 2005



ZHAO Xin-Chao, born in 1976, Ph. D., professor. His research interests include swarm intelligence, evolutionary computation and engineering optimization.

LIU Guo-Li, born in 1989, M.S. candidate. His research interests include low-level system software and full system simulation.

LIU Hu-Qiu, born in 1989, Ph.D. candidate. His research interests include operation systems and software.

ZHAO Guo-Shuai, born in 1990, M. S. candidate. His research interests include computer network, evolutionary computing and data mining.

Background

The preliminary results of this paper are completed as one undergraduate innovation project of Beijing University of Posts and Telecommunications, the research team is composed by Guo-Li LIU, Hu-Qiu LIU and Guo-Shuai ZHAO.

During we are doing researches, we find that particle swarm optimization algorithm (PSO) is good at the global search, however, easily trapped by local optima. How to maintain its advantages and reduce its disadvantages as much as possible becomes our research focus. At the same time, non-uniform mutation operation has its inherent excellent properties as we analyzed in our previous research works. Multiple stages perturbation operation on the superior solutions simultaneously considers the global exploration and the fine local search around the promising neighborhoods. So it is the initial motivation to combine them together and next to

check their cooperative benefits and the final effects.

As we know, how to balance the global search and local search between their pros and cons is pivotal to the performance of population-based algorithms. According to our previous research experience, non-uniform mutation and perturbed updating strategy are consistent to the inherent requirement of search algorithms. Under these guidelines, the first author let three students do some preliminary simulation works on how to hybridize them cooperatively. Of course, most of the experimental comparisons are replaced by more compelling experiments before the initial submission. The algorithmic analysis and simulation experiments are hugely improved and enhanced based on the constructive and critical comments from the domain experts, especially in two most recent revisions.