

文章编号: 2011-0950(2012)10-1459-06

基于二阶振荡及自然选择的随机权重混合粒子群算法

白俊强, 尹戈玲, 孙智伟

(西北工业大学 航空学院, 西安 710072)

摘要: 针对粒子群算法“早熟收敛”的缺点, 提出一种混合粒子群算法. 该算法采用最大速度线性递减的方法平衡全局寻优能力与算法收敛精度的矛盾, 并用随机权重平衡算法的全局和局部搜索能力. 学习因子二阶振荡使种群在粒子数目不变的情况下维持多样性, 是提高全局搜索能力的主要方法. 自然选择原理使算法改善了因二阶振荡和随机权重的加入而造成收敛速度降低的情况. 测试实验表明, 所提出的算法能避免早熟问题, 有效地提高寻优能力.

关键词: 最大速度线性递减; 随机惯性权重; 二阶振荡; 自然选择

中图分类号: TP273

文献标志码: A

Random weighted hybrid particle swarm optimization algorithm based on second order oscillation and natural selection

BAI Jun-qiang, YIN Ge-ling, SUN Zhi-wei

(School of Aeronautics, Northwestern Polytechnical University, Xi'an 710072, China. Correspondent: BAI Jun-qiang, E-mail: junqiang@nwpu.edu.cn)

Abstract: To overcome the disadvantage of particle swarm optimization(PSO) algorithm such as easily trapping into local optimal solution, this paper proposes a kind of hybrid particle swarm optimization algorithm. The maximum speed linear degressive method can effectively compromise between global searching capability and algorithm convergence precision. The random weight can effectively balance the global and local searching ability of the algorithm. Second-order oscillative learning factor can maintain the population diversity under the condition of invariable particle number. At the same time, natural selection principle can improve the convergence speed. The function test experimental results show that the proposed algorithm can avoid premature convergence problem, and effectively improve its optimization ability.

Key words: maximum speed linear degression; random inertia weight; second-order oscillation; natural selection

1 引言

粒子群优化(PSO)算法是 Kennedy 和 Eberhart^[1-2]受鸟群群体运动的启发于 1995 年提出的一种新的群智能优化算法. 由于 PSO 算法简单、收敛速度较快、参数少、容易实现等优点, 一经提出, 便得到国内外学者的广泛关注. 目前, 人们提出了许多改进的粒子群算法, 如: 文献[3]提出了带惯性因子的粒子群优化算法; [4]提出了带压缩因子的粒子群优化算法; [5]提出了模糊自适应粒子群优化算法; [6]提出了基于模拟退火粒子群算法; [7]提出了基于混沌变异的小生境粒子群算法; [8]提出了基于遗传交叉因子的改进粒子群优化算法等. 这些算法从不同方面对 PSO 算法进行了改进, 不同程度地提高了粒子群优化算法的性能, 但这些算法本身也存在着不足.

为了克服粒子群算法易早熟收敛、进化后期局

部搜索能力差等缺点, 本文主要做了如下工作: 1) 学习因子二阶振荡, 提高群体多样性和算法的全局寻优能力; 2) 针对采用线性递减权重方法造成的早熟问题, 改用随机权重法, 以避免算法早熟, 同时增强算法的局部搜索能力; 3) 采用最大速度线性递减的方法来缩小进化后期搜索范围, 提高进化后期局部收敛能力, 增加搜索精度和速度; 4) 将自然选择与粒子群算法相结合, 提高收敛速度.

2 标准粒子群算法

在标准粒子群(SPSO)算法^[9-10]中, 首先初始化一群随机粒子(随机解), 然后让这些粒子追随当前的最优粒子在解空间中搜索, 即通过迭代找到最优解. 假设 d 维搜索空间中的第 i 个粒子的位置和速度分别为 $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,d})$ 和 $V_i = (v_{i,1}, v_{i,2}, \dots, v_{i,d})$, 在每一次迭代中, 粒子通过跟

收稿日期: 2011-06-14; 修回日期: 2011-10-27.

作者简介: 白俊强(1971—), 男, 教授, 博士生导师, 从事飞行器设计等研究; 尹戈玲(1987—), 女, 硕士生, 从事飞行器总体设计的研究.

踪两个最优解来更新自己, 第一个就是粒子本身所找到的最优解, 即个体极值 p_{best} , 设为 $P_i = (p_{i,1}, p_{i,2}, \dots, p_{i,d})$; 另一个是整个种群目前找到的最优解, 即全局最优解 g_{best} , 记为 P_g . 在找到这两个最优值时, 粒子根据下式更新自己的速度和位置:

$$v_{i,j}(t+1) = \omega v_{i,j}(t) + c_1 r_1 [p_{i,j} - x_{i,j}(t)] + c_2 r_2 [p_{g,j} - x_{i,j}(t)], \quad (1)$$

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

其中: $x_{i,j}(t+1)$, $x_{i,j}(t)$, $v_{i,j}(t+1)$, $v_{i,j}(t)$ 分别表示第 i 个粒子在 $t+1$ 和 t 时刻的空间位移和运动速度; ω 为惯性权重因子; c_1 和 c_2 分别表示粒子个体的加速权重系数和粒子群体的加速权重系数; r_1 和 r_2 分别表示与粒子个体及群体的加速权重系数相关的随机初值, 为 0 到 1 之间均匀分布的随机数; $p_{i,j}$ 和 $p_{g,j}$ 分别表示第 i 个粒子个体在搜索过程中的最佳位置和粒子群体在搜索过程中的最佳位置. 每维中粒子的速度将限制在 $[V_{min}, V_{max}]$ 区间中, 如果 $v_{i,j}(t)$ 大于 V_{max} , 则用区间的边界值替代.

式 (1) 中的第 1 部分 $\omega v_{i,j}(t)$ 表示粒子自身的运动惯性; 第 2 部分 $c_1 r_1 (p_{i,j} - x_{i,j}(t))$ 表示粒子个体对自身搜索历史过程的认知, 引导粒子向自身经历全局最优方向运动; 第 3 部分 $c_2 r_2 (p_{g,j} - x_{i,j}(t))$ 表示粒子群体对搜索历史过程的认知, 引导粒子向全局最优方向运动.

根据上述原理得出标准粒子群优化算法的基本步骤如下:

Step 1: 随机初始化种群中各微粒的位置和速度, 粒子的种群规模设为 N ;

Step 2: 评价每个微粒的适应度, 将当前各微粒的位置和适应值存储在各微粒的 p_{best} 中, 将所有 p_{best} 中适应值最优个体的位置和适应值存储于 g_{best} 中;

Step 3: 根据式 (1) 和 (2) 更新粒子的速度和位移.

Step 4: 对每个微粒, 将其适应值与其经历过的最好位置作比较, 如果较好, 则将其作为当前的最好位置;

Step 5: 比较当前所有 p_{best} 和 g_{best} 的值, 更新 g_{best} ;

Step 6: 若满足停止条件 (通常为预设的运算精度或迭代次数), 则搜索停止, 输出结果, 否则返回 Step 3 继续搜索.

3 改进的粒子群算法

针对标准 PSO 算法的缺点, 本文提出一种基于

二阶振荡及自然选择的随机权重混合粒子群算法. 该算法改善了标准 PSO 算法早熟收敛的不足, 增强了全局搜索能力和局部搜索能力. 该算法主要有以下几方面的改进.

3.1 粒子二阶振荡

在 PSO 算法中, 速度更新公式 (如标准 PSO 算法速度更新公式 (1)) 中的第 2 部分和第 3 部分分别表示当前粒子对个体位置的认知和对群体位置的认知. 若速度变更不考虑第 2 部分 (即 $c_1 = 0$), 则粒子没有单独认知能力, 只有整体的趋向. 在微粒的相互作用下, 虽有能力到达新的搜索空间, 但对于复杂问题容易陷入局部最优点. 若速度变更不考虑第 3 部分 (即 $c_2 = 0$), 则微粒之间没有交互、没有整体信息共享, 一个规模为 m 的群体行为等价于 m 个单个个体独立运动, 因而得到最优解几率非常小. 本文中粒子速度更新采用二阶振荡^[11]的方法, 即在标准 PSO 算法的基础上, 粒子速度更新引入一个振荡环节, 这样做可进一步提高粒子群体的多样性, 从而改善算法的全局收敛性. 其速度更新方程为

$$v_{i,j}(t+1) = \omega v_{i,j} + c_1 r_1 [p_{i,j} - (1 + \xi_1)x_{i,j}(t) + \xi_1 x_{i,j}(t-1)] + c_2 r_2 [p_{g,j} - (1 + \xi_2)x_{i,j}(t) + \xi_2 x_{i,j}(t-1)], \quad (3)$$

$$\xi_1 \geq \frac{2\sqrt{c_1 r_1} - 1}{c_1 r_1}, \quad \xi_2 \geq \frac{2\sqrt{c_2 r_2} - 1}{c_2 r_2}, \quad (4)$$

$$\xi_1 \leq \frac{2\sqrt{c_1 r_1} - 1}{c_1 r_1}, \quad \xi_2 \leq \frac{2\sqrt{c_2 r_2} - 1}{c_2 r_2}, \quad (5)$$

其中 ξ_1, ξ_2 为随机数. 在算法前期取

$$\xi_1 \geq \frac{2\sqrt{c_1 r_1} - 1}{c_1 r_1}, \quad \xi_2 \geq \frac{2\sqrt{c_2 r_2} - 1}{c_2 r_2},$$

结合式 (3) 可推导出此时扩大粒子振荡范围. 以 2 维变量优化为例, 假设粒子 $x(t-1)$, $x(t)$, p_{best} 位置如图 1 所示, 则由式 (3) 和 (4) 可推导出粒子下一步的运动方向.

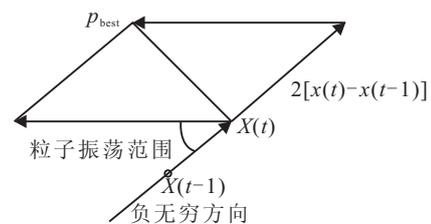


图 1 算法粒子振荡范围示意图

二阶振荡使粒子下一步运动矢量指向较优的一个角度范围之间, 提高了粒子的全局寻优能力; 粒子运动范围大, 使算法具有较强的全局搜索能力. 本文的仿真实验取前二分之一迭代步数为算法前期, 此时主要搜索全局最优值所在范围. 算法后期取

$$\xi_1 \leq \frac{2\sqrt{c_1 r_1} - 1}{c_1 r_1}, \quad \xi_2 \leq \frac{2\sqrt{c_2 r_2} - 1}{c_2 r_2},$$

通过增强局部搜索能力, 使算法能够渐近收敛. 此时

算法进入细致搜索时期, 局部搜索能力强, 适于精确寻找最优值.

由上述混合粒子群算法的速度更新公式 (3) 可以看出, 不同于标准 PSO 算法中微粒的飞行速度仅仅是微粒当前位置的函数, 改进的混合粒子群算法中, 当前粒子的个体位置认知和群体位置认知, 不仅与当前位置相关, 还与微粒位置的变化有关. 这样, 粒子可更快地向着有利的方向运动, 增强全局和局部收敛性, 并加快收敛速度.

3.2 随机权重

在粒子群算法的可调整参数中, 惯性权重 ω 是最重要的参数. 由式 (1) 可以看出, 假设速度变化只有式 (1) 中的第 1 部分 $\omega v_{i,j}(t)$ (即 $c_1 = c_2 = 0$), 则粒子将一直以当前的速度飞行, 直到到达边界. 由此可见, 较大的 ω 有利于跳出局部极小点, 提高算法的全局搜索能力; 而较小的 ω 有利于对当前的搜索区域进行精确局部搜索, 增强算法的局部搜索能力. 因此, 根据不同的权重变化公式, 可得到不同的 PSO 算法. Shi 和 Eberhart^[12] 提出了线性递减的权重, 使得算法初期有较大的 ω , 利于跳出局部极小点, 便于全局搜索; 而算法后期有较小的 ω , 利于当前区域局部搜索, 便于算法收敛. 但这种方法需要反复的实验确定最佳值, 而且如果在最初搜索阶段找不到最优值, 则随着 ω 的减小和收敛能力的增强, 易陷入局部最优. 本文将标准 PSO 算法中的惯性权重 ω 设定为服从某种随机分布的随机数, 这样可在一定程度上从两方面来克服 ω 采用线性递减所带来的不足. 首先, 如果在进化初期接近最好点, 则随机 ω 可能产生相对小的 ω 值, 加快算法的收敛速度; 如果在算法初期找不到最好点, 则 ω 的线性递减使得算法最终收敛不到此最好点, 而 ω 的随机生成可克服这种局限.

ω 的计算公式如下:

$$\begin{cases} \omega = \mu + \sigma * N(0, 1), \\ \mu = \mu_{\min} + (\mu_{\max} - \mu_{\min}) * \text{rand}(0, 1). \end{cases} \quad (6)$$

其中: $N(0, 1)$ 表示标准正态分布的随机数, $\text{rand}(0, 1)$ 表示 0 到 1 之间的随机数, μ_{\max} 为随机权重平均值的最大值, μ_{\min} 为随机权重平均值的最小值, μ 为随机权重平均值, σ 为随机权重的方差.

3.3 最大速度线性递减

在粒子群算法中, 每维中的粒子速度被限制在 $[V_{\min}, V_{\max}]$ 区间. 数值实验表明, V_{\max} 对粒子群算法的收敛精度和速度都有很大的影响. 本文采用最大速度线性递减的方法, 让最大速度随着迭代步数的增加从最大值减小到最小值, 使进化前期具有较大的最大速度, 容易跳出局部最优, 增加全局搜索性, 并在进

化后期具有较小的最大速度, 缩小搜索范围, 增强进化后期局部收敛能力, 提高搜索精度和速度. 另一方面, 增加了粒子最大速度的限制, 使粒子在进化初期不易陷入边界范围, 进而陷入边界局部最优, 增强算法的全局搜索能力. 最大速度随算法迭代次数的变化公式为

$$v_{\max} = \max - (t - 1) * (\max - \min) / (M - 1). \quad (7)$$

其中: v_{\max} 表示最大速度; \max, \min 分别表示最大速度的最大值和最小值; t 表示当前迭代步数, M 表示最大迭代步数.

3.4 自然选择

改进混合粒子群算法在上述几种改进的基础上, 还加入了自然选择的思想, 最终得到基于二阶振荡及自然选择的随机权重粒子群优化算法. 其基本思想是, 在每次迭代过程中将整个粒子群按适应值排序, 用群体中最好的前 $1/N$ 粒子的速度和位置替换群体中最差的 $1/N$ 粒子的速度和位置, 同时保留原来每个个体所记忆的历史最优值, 提高粒子池中接近最优值粒子的比重, 以保证每次迭代的粒子都具有较好的性能, 同时加快收敛速度. 在本文实验中, 为了保持粒子多样性及有好的全局性, 防止粒子陷入局部最优, 取 N 为 10, 即只替换 10% 的粒子.

3.5 混合粒子群优化算法

经过上述方法将基本的粒子群优化算法进行改进, 得到基于二阶振荡及自然选择的随机权重混合粒子群算法 (RwSecSelPSO), 其流程如图 2 所示.

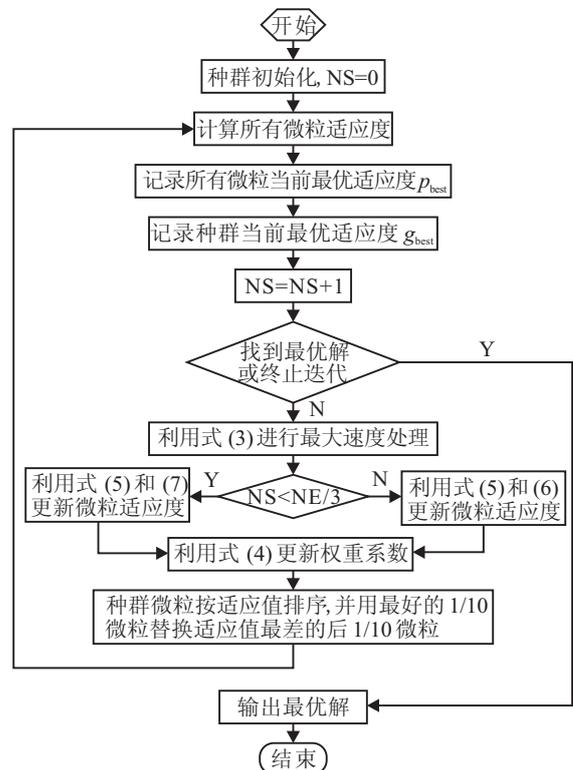


图 2 RwSecSelPSO 流程图

RwSecSelPSO 算法的具体步骤如下:

Step 1: 随机初始化种群中各微粒的速度和位置,粒子的种群规模设为 N .

Step 2: 评价每个微粒的适应度,将当前各微粒的位置和适应值存储在各微粒的 p_{best} 中,将所有 p_{best} 中适应值最优个体的位置和适应值存储于 g_{best} 中.

Step 3: 根据式 (7) 对粒子更新速度的最大值进行线性递减,以保证收敛性.

Step 4: 根据式 (2) 和 (3) 对微粒的速度和位置进行更新,算法前期取

$$\xi_1 \geq \frac{2\sqrt{c_1 r_1} - 1}{c_1 r_1}, \xi_2 \geq \frac{2\sqrt{c_2 r_2} - 1}{c_2 r_2};$$

算法后期则取

$$\xi_1 \leq \frac{2\sqrt{c_1 r_1} - 1}{c_1 r_1}, \xi_2 \leq \frac{2\sqrt{c_2 r_2} - 1}{c_2 r_2}.$$

Step 5: 根据式 (6) 更新权重系数 ω .

Step 6: 对于每个微粒,将其适应值与其经历过的最好位置作比较,如果较好,则将其作为当前的最好位置 p_{best} .

Step 7: 比较当前所有 p_{best} 和 g_{best} 的值,更新 g_{best} .

Step 8: 将整个粒子群按适应值进行排序,用群体中最好的前 $1/N$ 粒子的速度和位置替换最差的 $1/N$ 粒子的速度和位置,保持 p_{best} 和 g_{best} 不变.

Step 9: 若满足停止条件(设为迭代次数),则搜索停止,输出结果,否则返回 Step 3 继续搜索.

4 对比实验及结果分析

为了测试 RwSecSelPSO 混合粒子群算法的性能,本文分别用 5 个典型的标准函数对两种相关算法进行比较测试.使用 Matlab 2009 a (32 位) 进行算法测试.

1) 测试函数.

f_1 为 Levy No.5 函数,即

$$f(x, y) = \sum_{i=1}^5 [i \cos((i-1)x + 1)] \sum_{j=1}^5 [j \cos((j-1)y + 1)] + (x + 1.42513)^2 + (y + 0.80032)^2. \quad (8)$$

这个函数是一个著名的两维测试函数,具有 760 个局部极值点.众多的局部极值点使各种优化算法很难搜索到全局最优解.函数的全局最优解为 -176.1376 , $X = (-1.3068, -1.4248)$.

f_2 为 Rastrigin 函数,即

$$f(x) = \sum_{i=1}^n [x(i)^2 - 10 * \cos(2\pi x(i)) + 10]. \quad (9)$$

这个函数对进化算法具有很强的欺骗性,它有非常多的局部最小值点和局部最大值点,很容易使算法陷入

局部最优,而不能得到全局最优解.此函数全局最优解为 0 , 分布在 $X = (0, \dots, 0)$.

f_3 为 Shifted Rotated Rastrigin 函数^[13], 即

$$\begin{cases} f(x) = \sum_{i=1}^D (z_i^2 - 10 \cos(2\pi z_j)), \\ z = (x - o) * M. \end{cases} \quad (10)$$

这个函数相较于函数 f_2 带有旋转性,对进化算法具有更强的欺骗性.此函数全局最优解为 -330 , 其函数示意图如图 3 所示.

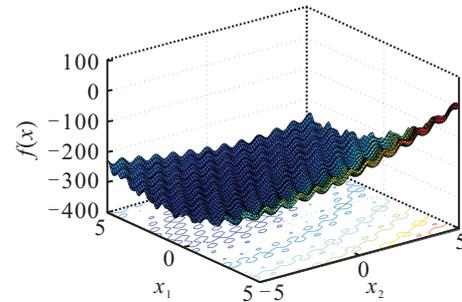


图 3 Rotated Rastrigin 函数

f_4 为 Shifted Rotated Griewank 函数^[13], 即

$$\begin{cases} f(x) = \sum_{i=1}^D \frac{z_j^2}{4000} - \prod_{i=1}^D \cos\left(\frac{\sqrt{i}}{z_i}\right) + 1, \\ z = (x - o) * M. \end{cases} \quad (11)$$

这个函数也是一个带有旋转的典型的测试函数,如图 4 所示,全局最优解为 -180 .

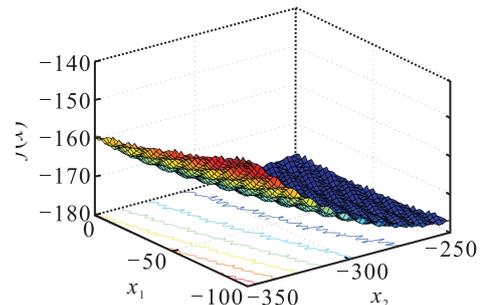


图 4 Rotated Griewank 函数

f_5 为 Shifted Rotated Expanded Scaffer's F6 函数^[13], 即

$$\begin{cases} F(x, y) = 0.5 + \frac{(1 + 0.001(x^2 + y^2))^2}{(\sin^2(\sqrt{x^2 + y^2}) - 0.5)}, \\ f(x) = EF(z_1, z_2, \dots, z_D) = \\ F(z_1, z_2) + F(z_2, z_3) + \dots + \\ F(z_{D-1}, z_D) + F(z_D, z_1), \\ z = (x - o) * M, \\ x = [x_1, x_2, \dots, x_D]. \end{cases} \quad (12)$$

这个函数是一个复杂的复合函数,而且带有旋转性,为选取的 5 个测试函数中最复杂的一个.此函数对进化算法具有非常强的欺骗性,极易使算法陷入局

部最优, 而不能得到全局最优解. 此函数全局最优解为 -300 , 函数示意图如图 5 所示.

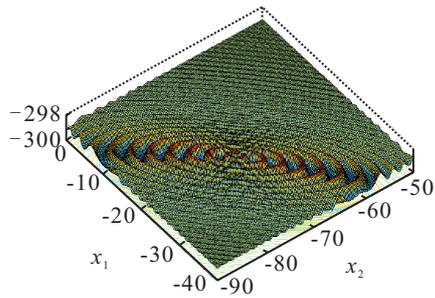


图 5 Rotated Expanded Scaffer's F6 函数

2) 算法种类.

为了比较的需要, 本文同时对混合粒子群算法 RwSecSelPSO, 标准粒子群算法在上述 5 个测试函数中进行优化测试, 并将结果进行对比.

3) 参数设置.

RwSecSelPSO 算法的参数设置如下: 粒子数为 20, 自身学习因子为 2, 全局学习因子为 2, 权重平均最大值为 0.9, 权重平均最小值为 0.2, 权重方差为 0.2, 淘汰概率为 0.1, 迭代代数 为 500, 测试维数为 2; 测试函数 f_1 的最大速度 v_{max} 的最大值 max 为 10, 最小值 min 为 0.000 1, 上下边界为 ± 10 ; 测试函数 f_2 的最大速度 v_{max} 的最大值 max 为 3, 最小值 min 为 0.000 1, 上下边界为 ± 3 ; 测试函数 f_3 的最大速度 v_{max} 的最大值 max 为 10, 最小值 min 为 0.000 1, 上下边界为 ± 5 ; 测试函数为 f_4 的最大速度 v_{max} 的最大值 max 为 10, 最小值 min 为 0.000 1, 上下边界为 ± 1000 ; 测试函数 f_5 的最大速度 v_{max} 的最大值 max 为 10, 最小值 min 为 0.000 1, 上下边界为 ± 100 .

PSO 算法的参数设置如下: 粒子数为 20, 自身学习因子为 2, 全局学习因子为 2, 权重系数为 0.5, 迭代代数为 500, 问题维数为 2; 测试函数 f_1 的上下边界为 ± 10 ; 测试函数 f_2 的上下边界为 ± 3 ; 测试函数 f_3 的上下边界 ± 5 ; 测试函数 f_4 的上下边界为 ± 1000 ; 测试函数 f_5 的上下边界为 ± 100 .

4) 测试结果.

两种算法在 5 个测试函数的测试次数均为 50 次, 具体测试结果如表 1 所示. 其中: 全局最优值次数是指成功找到全局最优值的次数; 寻优度是指成功搜索的比率, 为成功找到全局最优值的次数与总测试次数的比值; 最优值是指所有搜索结果中的最优值; 平均全局最优值是指成功搜索时最优值的平均值; 全局最优值标准差是指成功搜索时最优值与函数最优解的标准方差.

表 1 测试结果对比

函数	参数	SPSO	RwSecSelPSO
f_1	全局最优值次数	37	50
	寻优度/%	74	100
	最优值	-176.137 6	-176.137 6
	平均全局最优值	-176.137 6	-176.137 45
	全局最优值标准差	0	0.000 258 851
f_2	全局最小值次数	32	50
	寻优度/%	64	100
	最优值	0	0
	平均全局最小值	0	0
f_3	全局最优值标准差	0	0
	全局最优值次数	27	47
	寻优度/%	54	94
	最优值	-330	-330
f_4	平均全局最优值	-330	-329.999 667
	全局最优值标准差	0	0.000 571 268
	全局最优值次数	15	39
	寻优度/%	30	78
f_5	最优值	-180	-180
	平均全局最优值	-180	-179.999 998 2
	全局最优值标准差	0	3.05179e-06
	全局最优值次数	4	30
	寻优度/%	8	60
f_5	最优值	-300	-300
	平均全局最优值	-300	-299.999 883
	全局最优值标准差	0	0.000 202 675

图 6 为两种算法对 5 个测试函数的收敛曲线图. 图 6 中: PSO1, PSO2, ..., PSO5 为标准 PSO 算法经过测试分别找到全局最优和陷入局部最优的收敛曲线; RwSecSelPSO, RwSecSelPSO1, RwSecSelPSO2 为改进的混合粒子群算法分别找到全局最优和陷入局部最优的收敛曲线.

5) 结果分析.

由表 1 可以看出: 标准 PSO 算法的搜索比较精细, 由式 (1) 可推出标准 PSO 算法的粒子运动范围, 其速度变化范围小, 且随着进化代数的增加速度逐渐减小, 虽然这种速度进化方式使进化速度很快, 且在进化后期的搜索很精细, 但此种进化方式极易陷入局部极值, 全局收敛性并不好; 而改进后的混合粒子群算法 RwSecSelPSO 在保证局部收敛精度的同时, 提高了全局收敛性, 不易陷入局部最优值, 寻优度显著优于标准粒子群算法, 有效地改进了标准 PSO 算法“早熟收敛”的不足. 由图 6 可以看出: 虽然标准 PSO 算法收敛到最小值的进化步数少于 RwSecSelPSO 算法, 但却极易陷入局部最优值; 而改进的 RwSecSelPSO 算法虽然需要更多步进化才能收敛到最优值, 但其在收敛过程中粒子一直保持振荡性, 有好的全局收敛性, 可保证其能最终收敛到全局最优值. 实验证明, 改进算法具有更好的全局收敛性的同时仍然具有较好的局部收敛精度, 对于相对复杂一些的问题有更强的实用性和有效性.

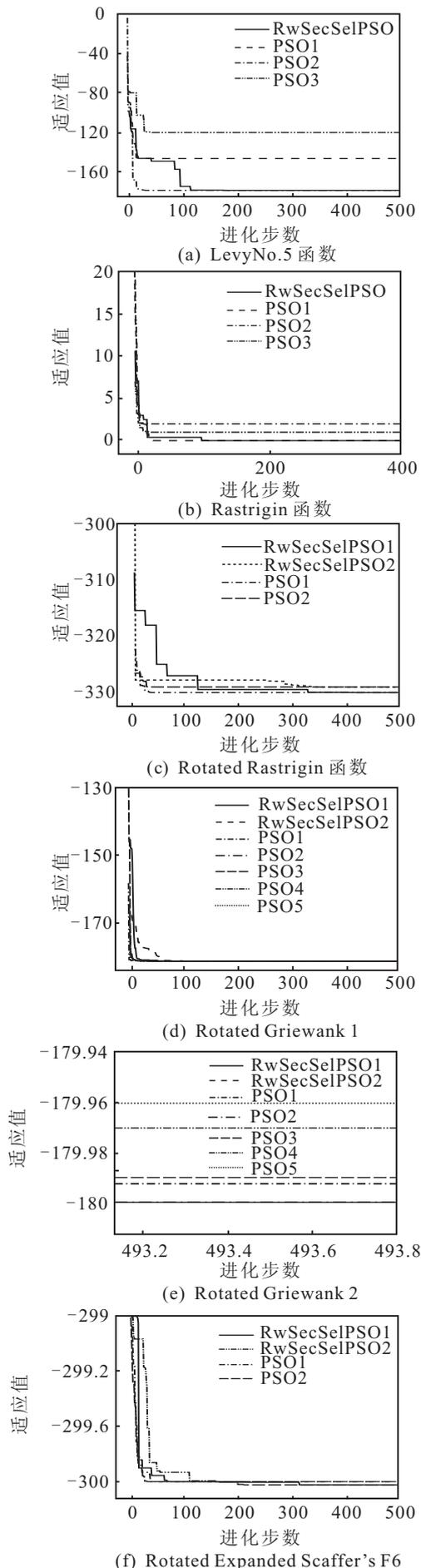


图 6 函数测试收敛曲线

5 结 论

本文针对粒子群算法易陷入局部最优以及由于种群多样性减少、搜索精度不高的特点,提出了基于二阶振荡和自然选择的随机权重混合粒子群优化算法.通过实验可以看出,改进的混合粒子群算法较标准粒子群算法而言提高了算法的鲁棒性,具有更好的全局寻优能力,适用于解决复杂函数的优化问题.

参考文献(References)

- [1] Kennedy J, Eberhart R C, Shi Y. Swarm intelligence[M]. San Francisco: Morgan Kaufmann Publishers, 2001: 227-229.
- [2] Kennedy J, Eberhart R C. Particle swarm optimization[C]. Proc of IEEE Int Conf on Neural Networks. Washington, 1995: 1942-1948.
- [3] Shi Y, Eberhart R. A modified particle swarm optimizer[C]. IEEE World Congress on Computational Intelligence. Indianapolis: Indiana University, 1998: 69-73.
- [4] Clerc M. The swarm and the queen towards a deterministic and adaptive particle swarm optimization[C]. Proc of the Congress of Evolutionary Computation. Annecy: France Telecom, 1999: 1951-1957.
- [5] Shi Y, Eberhart R C. Fuzzy adaptive particle swarm optimization[C]. Proc of the Congress on Evolutionary Computation. Seoul Korea, 2001: 101-106.
- [6] 高鹰, 谢胜利. 基于模拟退火的粒子群优化算法[J]. 计算机工程与应用, 2004, 40(1): 47-49.
(Gao Y, Xie S L. Particle swarm optimization algorithms based on simulated annealing[J]. Computer Engineering and Applications, 2004, 40(1): 47-49.)
- [7] 贾东立, 张家树. 基于混沌变异的小生境粒子群算法[J]. 控制与决策, 2007, 22(1): 117-120.
(Jia D L, Zhang J S. Niche particle swarm optimization combined with chaotic mutation[J]. Control and Decision, 2007, 22(1): 117-120.)
- [8] 李季, 孙秀霞, 李士波, 等. 基于遗传交叉因子的改进粒子群优化算法[J]. 计算机工程, 2008, 34(2): 181-183.
(Li J, Sun X X, Li S B. Improved particle swarm optimization based on genetic hybrid genes[J]. Computer Engineering, 2008, 34(2): 181-183.)
- [9] Riccardo Poli, James Kennedy, Tim Blackwell. Particle swarm optimization[J]. Swarm Intell, 2007, 1(1): 33-57.
- [10] Bratton D, Kennedy J. Defining a standard for particle swarm optimization[C]. Proc of IEEE Swarm Intelligence Symposium. Hawaii, 2007: 120-127.
- [11] 龚纯, 王正林. 精通 Matlab 最优化计算[M]. 北京: 电子工业出版社, 2009: 296-299.
(Gong C, Wang Z L. To be proficient in matlab optimization calculation[M]. Beijing: Publishing House of Electronics Industry, 2009: 296-299.)

(下转第1470页)