

具备反向学习和局部学习能力的粒子群算法

夏学文^{1),2)} 刘经南¹⁾ 高柯夫¹⁾ 李元香³⁾ 曾 辉³⁾

¹⁾(武汉大学卫星定位导航技术研究中心 武汉 430079)

²⁾(华东交通大学软件学院 南昌 330013)

³⁾(武汉大学计算机学院 武汉 430079)

摘 要 为解决粒子群优化(Particle Swarm Optimization, PSO)算法中存在的种群多样性和收敛性之间的矛盾,该文提出了一种具备反向学习和局部学习能力的粒子群优化算法(Reverse-learning and Local-learning PSO, RLPSO).该算法保留了初始种群中满足排异距离要求的多个较差粒子以及每个粒子的历史最差位置.当检测到算法陷入局部最优时,利用这些较差粒子的位置信息指导部分粒子以较快飞行速度进行反向学习,将其迅速牵引出局部最优区域.反向学习过程可改善粒子种群的多样性,保证了算法的全局探测能力;同时,利用较优粒子间的差分结果指导最优粒子进行局部学习与搜索,该过程可与粒子群的飞行过程并行执行,且局部学习的缩放因子可随进化过程动态调节.局部学习可提高算法的求解精度,保证算法的迅速收敛.实验结果表明,RLPSO 算法同其他 PSO 算法相比,在高维函数优化中具有收敛速度快、求解精度高的特点.

关键词 粒子群算法;反向学习;局部搜索;多样性保持;高维函数优化

中图法分类号 TP18 **DOI 号** 10.11897/SP.J.1016.2015.01397

Particle Swarm Optimization Algorithm with Reverse-Learning and Local-Learning Behavior

XIA Xue-Wen^{1),2)} LIU Jing-Nan¹⁾ GAO Ke-Fu¹⁾ LI Yuan-Xiang³⁾ ZENG Hui³⁾

¹⁾(GNSS Research Center of Wuhan University, Wuhan 430079)

²⁾(School of Software, East China Jiaotong University, Nanchang 330013)

³⁾(Computer School of Wuhan University, Wuhan 430079)

Abstract To resolve conflict between convergence and diversity in particle swarm optimization (PSO) algorithm, an improved PSO algorithm which called reverse-learning and local-learning PSO (RLPSO) algorithm is introduced. In RLPSO, a reverse-learning behavior is adopted by some particles and local-learning behavior is adopted by elite particles. In RLPSO, some inferior particles of initial population and each particle's historical worst position are reserved. Furthermore, the hamming distance among the inferior particles is no less than a rejection distance that predefined. While population has being trapped into a local optimum, the inferior particles and a particle's historical worst position can attract the particle to leap out of the local optimums in a high speed. This action is called reverse-learning behavior which can preservation population diversity and improve RLPSO's exploration ability. Furthermore, in each generation, the difference between the best particle and the second-best particle is adopted to guide the best one to carry out a local search process called local learning behavior by which exploitation ability of population can

收稿日期:2013-03-12;最终修改稿收到日期:2014-12-15. 本课题得到国家自然科学基金(41231174, 61165004)、江西省自然科学基金(20151BAB207022)及华东交通大学校立科研基金(14JG03)资助. 夏学文,男,1974年生,博士,副教授,主要研究方向为计算智能及其应用. E-mail: xwxia@whu.edu.cn. 刘经南,男,1943年生,教授,博士生导师,中国工程院院士,主要研究领域为全球定位系统理论、方法及数据处理. 高柯夫,男,1981年生,博士,讲师,主要研究方向为信息安全、位置服务. 李元香,男,1962年生,博士,教授,博士生导师,主要研究领域为并行计算、计算智能及其应用. 曾 辉,男,1980年生,博士研究生,主要研究方向为计算智能及其应用.

be improved. In the local learning behavior that can parallel execute with population's evolution, the local scale factor is dynamic adjusted during the evolution. The results achieved by RLPSO were compared with some modified PSO algorithm, which indicated that RLPSO has better global searching ability and higher convergence speed especially in high dimension functions.

Keywords particle swarm optimization (PSO) algorithm; reverse-learning; local research; diversity preservation; high-dimensional function optimization

1 引 言

粒子群优化算法(Particle Swarm Optimization, PSO)是由 Kennedy 和 Eberhart^[1]提出的一种基于群体智能的随机优化方法. PSO 中每一个粒子都代表一个问题的候选解,通过粒子个体的简单行为及群体内的信息交互实现问题求解的智能性. 由于 PSO 操作简单、收敛速度快,因此在函数优化^[2]、图像处理^[3]、大地测量^[4]等众多领域都得到了广泛的应用. 同时,由于 PSO 算法存在早熟收敛、维数灾难、易于陷入局部极值等问题,因此,针对这些问题,产生了很多有效的改进方法,主要可分为以下几类:

第 1 类是通过调整 PSO 的参数来平衡算法的全局探测和局部开采能力. 如 Shi 和 Eberhart^[5]对 PSO 算法的速度项引入了惯性权重,并依据迭代进程及粒子飞行情况对惯性权重进行线性(或非线性)的动态调整,以平衡搜索的全局性和收敛速度. Trelea^[6]和 Jiang 等人^[7]先后对标准粒子群算法进行了收敛性分析,并提出了收敛性和稳定性较好的一组参数选择. 2009 年张玮等人^[8]在对标准粒子群算法位置期望及方差进行稳定性分析的基础上,研究了加速因子对位置期望及方差的影响,得出了一组较好的加速因子取值.

第 2 类是致力于设计不同类型的拓扑结构,改变粒子学习模式,从而提高种群的多样性. 如 Kennedy 等人^[9-10]提出几种基本的邻域结构,如环形结构和星形结构等,并分析了不同邻域结构对优化结果的影响. 文献[11]中,每个粒子学习的对象不是其单个邻居粒子的最优解,而是其全部邻居粒子的最优解的综合信息,在文献[12-13]中,则利用粒子间欧氏距离的大小来选择粒子的学习对象,以改善种群的多样性. 谭阳等人^[14]将 PSO 中指导粒子群进化的全局最优解扩展为由多个精英粒子组成的精英子群,精英粒子间采用了排异策略,并采用适应值竞争策略对整个粒子群进行指导,从而保证种群的多样

性. 而在 CLPSO^[15]中,每个粒子在变量的不同维上选择不同的粒子进行学习,避免所有维向同一粒子学习时产生的早熟收敛,实验也证明该算法在多峰函数上具有较好的优化效果. 这些通过对改进邻域结构来丰富粒子学习模式的方法都从不同程度上克服了标准 PSO 算法的早熟现象.

第 3 类是将 PSO 和其他优化算法(或策略)相结合,形成混合 PSO 算法. 如曾毅等人^[16]将模式搜索算法嵌入到 PSO 算法中,实现了模式搜索算法的局部搜索能力与 PSO 算法的全局寻优能力的优势互补. Angeline^[17]和 Higashi 等人^[18]分别将遗传算法中的选择操作和杂交操作引入到 PSO 算法中,有效提高了算法的收敛性. Xin 等人^[19]将 PSO 和差分进化算法(Differential Evolution, DE)相结合,通过设定一个学习统计周期对该周期内的两种算法的性能进行分析,并依据统计结果更新 PSO 和 DE 算法被选中执行的概率. 文献[20]中还提出了以 PSO 和人工蜂群算法(Artificial Bee Colony, ABC)为基础的混合算法,PSO 和 ABC 共享全局历史最优解,并分别指导 2 个子种群并行进化. 2014 年, Xia 等人^[21]在种群搜索过程中定期对每维决策变量空间进行划分,通过对历史进化信息的统计分析决定重点搜索区域,实现搜索空间的逐步收缩,进而达到提高搜索效率和求解精度的目的,此外,采用的禁忌探测技术也有效提高了算法跳出局部最优的能力.

第 4 类是采用小生境技术. 小生境是模拟生态平衡的一种仿生技术,适用于多峰函数和多目标函数的优化问题. 例如,在 PSO 算法中,通过构造小生境拓扑,将种群分成若干个子种群,动态地形成相对独立的搜索空间,实现对多个极值区域的同步搜索,从而可以避免算法在求解多峰函数优化问题时出现早熟收敛现象^[22-23]. Suganthan^[12]在 1999 年提出一种邻域可变的 PSO 算法,其基本思想是在算法开始阶段,每个个体的邻域为其自身,随着进化代数的增长其邻域范围也在不断增大直至整个种群. Parsopoulos^[22]提出一种基于“分而治之”思想的多种群 PSO 算法,

其核心思想是将高维的目标函数分解成多个低维函数,然后每个低维的子函数由一个子粒子群进行优化,该算法对高维问题的求解提供了一个较好的思路。

需要指出的是,很多对 PSO 算法的改进其实不仅限于其中的某一类,而更多的是不同改进方法的融合。

本文提出了一种改进的 PSO 算法:具有反向学习和局部学习能力的粒子群算法(Reverse-learning and Local-learning PSO, RLPSO)。在粒子群的每轮迭代中,RLPSO 借助最优粒子间的差分结果来指导最优粒子进行局部搜索,从而增强算法的局部开采能力,提高求解精度。当算法出现“早熟”收敛时,粒子群将分化为两个部分,一部分粒子负责协同最优粒子强化局部搜索,其他粒子则向其个体历史最差解和初始种群最差解进行反向学习,引导这些粒子逃离局部最优区域。同时,在反向学习时,放松了飞行速度限制,保证粒子能更快地飞离局部最优。反向学习行为与最优粒子的局部学习行为协同工作,有效地兼顾全局探测与局部开采能力。

2 PSO 算法

PSO 算法是从鸟群觅食模型中得到启示的一种基于群智能的随机搜索算法,已广泛应用于各类优化问题。在使用 PSO 算法时,每个粒子可视为 D 维搜索空间中的一个搜索个体,粒子的当前位置即为对应优化问题的一个候选解,粒子的飞行过程即为该个体的搜索过程。粒子的飞行速度可根据粒子历史最优位置和种群历史最优位置进行动态调整。

假设 D 维搜索空间中,有 N 个粒子组成一个群体,其中第 i ($i < N$) 个粒子在第 t 代时可用两个指标来描述:位置可表示为 $X_i^t = (x_{i1}^t, x_{i2}^t, \dots, x_{ij}^t, \dots, x_{iD}^t)$ 的 D 维向量,飞行速度可表示为 $V_i^t = (v_{i1}^t, v_{i2}^t, \dots, v_{ij}^t, \dots, v_{iD}^t)$ 的 D 维向量。若第 i 个粒子搜索至第 t 代时的个体历史最优位置为 $P_i = (p_{i1}, p_{i2}, \dots, p_{ij}, \dots, p_{iD})$,搜索至第 t 代时的整个粒子群的历史最优位置为 $P_g = (p_{g1}, p_{g2}, \dots, p_{gj}, \dots, p_{gD})$,则在第 $t+1$ 代时,第 i 个粒子的第 j 维速度和位置的迭代更新公式如下:

$$v_{ij}^{t+1} = \omega \cdot v_{ij}^t + c_1 \cdot r_1 \cdot (p_{ij}^t - x_{ij}^t) + c_2 \cdot r_2 \cdot (p_{gj}^t - x_{ij}^t) \quad (1)$$

$$x_{ij}^{t+1} = x_{ij}^t + v_{ij}^t \quad (2)$$

其中, ω 为惯性权重,衡量着前一时刻的速度对下次移动的影响, c_1 和 c_2 为学习因子, r_1 和 r_2 为 $[0, 1]$ 内的随机数。

惯性权重 ω 作为 PSO 算法的一个重要参数,对于平衡算法的收敛速度和全局搜索能力有着重要的作用。研究表明,较大的 ω 有利于全局搜索,并增加种群的多样性,而较小的 ω 则可以提高算法的局部开采能力,加快收敛速度。因此,在众多改进的 PSO 算法中,根据搜索进程而动态调节惯性权重 ω 的值已成为一个被普遍接受的观点,其中,较为常用的是线性递减权重策略,即从算法前期的较大值线性递减至一个较小值,以保证算法较好的前期全局搜索能力和后期局部搜索能力。 ω 常用的更新方式如下:

$$\omega^t = \omega_{\max} - \frac{\omega_{\max} - \omega_{\min}}{T} \cdot t \quad (3)$$

其中, t 为当前迭代次数, T 为最大迭代次数, ω_{\max} 为初始惯性权重,通常取 0.9, ω_{\min} 为粒子群最大迭代次数时的惯性权重,通常取 0.4。

粒子的飞行速度 v 表征着粒子遍历解空间的能力,它直接影响着粒子位置更新的步长。最大飞行速度 v_{\max} 决定着当前位置与最好位置之间区域的分辨率,如果速度太大,尽管收敛速度很快,但可能会飞过最优解;反之,若速度太小,尽管收敛的分辨率很高,但又不易跳出局部极值,不利于搜索到全局最优解。因此, v_{\max} 一般取变量可行域的 10%~20%。迭代过程中,粒子的飞行速度将被约束在设定的范围内,即如果 $v_{ij}^{t+1} > v_{\max}$ 时, $v_{ij}^{t+1} = v_{\max}$; 如果 $v_{ij}^{t+1} < -|v_{\max}|$ 时, $v_{ij}^{t+1} = -|v_{\max}|$ 。

c_1 和 c_2 表征着粒子个体向个体历史最优解和种群历史最优解的学习能力大小。当 $c_2 = 0, c_1 \neq 0$ 时,粒子的学习过程类似于在其个体历史最优解附近的一种变异操作,整个粒子群表现出较好的分布多样性,但群体信息交互能力太弱,因此全局收敛的速度很慢;当 $c_1 = 0, c_2 \neq 0$ 时,粒子仅向群体历史最优解学习,粒子群全局收敛的能力较强,但超级个体的巨大吸引力使得算法很容易陷入局部最优。以往的研究表明, c_1, c_2 的取值一般在 1.5~2.0 之间时,算法效果较好^[8]。

3 具备反向-局部学习的 RLPSO 算法

和其他群智能算法一样,PSO 算法在优化过程中,种群的多样性和算法的收敛速度之间始终存在着矛盾。对标准 PSO 算法的改进,无论是参数的选取、小生境技术的采用或是其他技术与 PSO 的融合,其目的都是希望在加强算法局部搜索能力的同时,保持种群的多样性,防止算法在快速收敛的同时出现早熟收敛。RLPSO 算法也是基于这一思想的,

即在加强种群历史最优个体的局部开采能力的同时,依靠其他多个粒子的反向学习能力,保持种群的多样性.具体采用的改进策略如下:

(1) 种群历史最优位置邻域的局部搜索. 粒子群在每代飞行更新位置后,更新种群历史最优位置 P_{g1} 和历史次优位置 P_{g2} , 并利用 P_{g1} 与 P_{g2} 之间的差分结果指导 P_{g1} 进行局部搜索, 增强 P_{g1} 邻域内的搜索能力, 具体如下:

$$P'_{g1} = P_{g1} + r \cdot d_t \cdot (P_{g1} - P_{g2}) \quad (4)$$

式中, r 为 $[-1, +1]$ 之间均匀分布的随机数, 用以控制局部搜索的方向, d_t 为第 t 代时的局域缩放因子. 一般而言, 在进化初期, P_{g1} 距离函数最优解一般较远, 这时较大的 d_t 能使其以较大的半径进行局部搜索, 加快收敛速度; 而到进化末期, P_{g1} 距离函数最优解一般较近, 这时则希望通过较小的 d_t 使其开采更高精度的解. 因此, 这里采用了和惯性权重 ω 类似的线性递减策略, 即 d_t 随着种群迭代次数 t 的增加而线性递减, 具体为

$$d_{t+1} = d_t \cdot (1 - t/T) \quad (5)$$

对局部搜索的结果 P'_{g1} 采用贪心保留策略, 即

$$P_{g1} = \begin{cases} P'_{g1}, & \text{fit}(P'_{g1}) > \text{fit}(P_{g1}) \\ P_{g1}, & \text{其他} \end{cases} \quad (6)$$

式中, $\text{fit}(x)$ 为 x 的适应值.

(2) 粒子的反向学习. 当检测到种群历史最优位置在一定迭代步数内未更新时, 则认为算法已陷入停滞, 并启动当前种群中的 n 个粒子进行 s 代的反向学习过程, 其他的 $N - n$ 个粒子的学习方式不变. 第 i 个粒子反向学习的对象为该粒子的历史最差位置 $W_i = (\omega_{i1}, \omega_{i2}, \dots, \omega_{ij}, \dots, \omega_{iD})$, 以及初始种群时选择的规模为 m 的初始最差粒子个体的位置集合 $W = \{W_1^0, W_2^0, \dots, W_m^0\}$ 中的任一个体 $W_k^0 = (\omega_{k1}^0, \omega_{k2}^0, \dots, \omega_{kj}^0, \dots, \omega_{kD}^0) (1 \leq k \leq m)$. 第 i 个粒子反向学习过程时的速度更新公式为

$$v_{ij}^{t+1} = \omega \cdot v_{ij}^t + c_3 \cdot r_3 \cdot (x_{ij}^t - w_{ij}^t) + c_4 \cdot r_3 \cdot (x_{ij}^t - w_{kj}^0) \quad (7)$$

式中, $w_{ij}^t (1 \leq i \leq n)$ 为第 i 个粒子进化到第 t 代时, 其历史最差位置 W_i^t 的第 j 维的值, $w_{kj}^0 (1 \leq k \leq m)$ 为随机选择的初始最差粒子个体位置 W_k^0 的第 j 维的值.

为了保证 m 个初始最差粒子能将反向学习的粒子牵引出当前的局部最优区域, 并较为广泛地分布到搜索区域中, 这 m 个初始最差粒子间应该具有较大的欧式距离, 所以在选择它们时需要保证其两

两间的距离大于预设的排异半径 R . 最差粒子构成子种群的生成算法见算法 1.

算法 1. 初始最差粒子子种群的生成算法.

输入: 初始种群 $X^0 = \{X_1^0, X_2^0, \dots, X_N^0\}$ 、排异半径 R 、规模

为 m 的初始最差粒子群体 W , 初始时为空

输出: 初始最差粒子群体 $W = \{W_1^0, W_2^0, \dots, W_m^0\}$

Begin

1. 依照粒子个体 $X_1^0 \sim X_N^0$ 的适应值 fit 升序排列, 不失一般性, 仍记为

$\{X_1^0, X_2^0, \dots, X_N^0\}, \text{fit}_{X_i^0} \leq \text{fit}_{X_j^0}, i, j \in [1, N]$ 且 $i < j$;

$W_1^0 = X_1^0, \text{count} = 1, W = W \cup \{W_{\text{count}}^0\}, i = 1$;

2. FOR $i = 2$ TO N

3. IF $\text{count} \geq m$ break;

4. ELSEIF $\forall j (\|W_j^0 - X_i^0\| > R), 1 \leq j \leq \text{count}$

5. $\text{count}++$; $W_{\text{count}}^0 = X_i^0$; $W = W \cup \{W_{\text{count}}^0\}$;

6. END IF

7. ENDFOR

8. WHILE $\text{count} < m$ // 初始最差粒子群体 W 未满

9. 随机生成新的粒子个体 ind ;

10. IF $\forall j (\|\text{ind} - W_j^0\| > R), 1 \leq j \leq \text{count}$

11. $\text{count}++$; $W_{\text{count}}^0 = \text{ind}$; $W = W \cup \{W_{\text{count}}^0\}$;

12. END IF

13. END WHILE

End

(3) 反向学习期间, 粒子的最大飞行速度 v_{\max} 进行动态调整. 粒子进行反向学习的目的是利用种群初始最差位置和个体历史最差位置的牵引作用, 迅速逃离局部最优, 因此较大的 v_{\max} 值可以提高其逃逸速度. 此外, 由于粒子群在开始反向学习之前, 算法已陷入停滞, 即种群历史最优解 P_{g1} 在其邻域内无法找到更优解, 因此也需要增大其 v_{\max} 值, 使其能在更大活动范围内进行搜索, 提高求解的成功率. 当反向学习结束后, v_{\max} 将再次设置为正常学习时的取值.

RLPSO 算法的伪代码见算法 2.

算法 2. RLPSO 算法.

输入: 算法搜索空间维数 D , 种群规模 N , 规模为 m 的

初始最差粒子群体 W (初始时为空), 学习因子

c_1, c_2 , 最大惯性权重 ω_{\max} 和最小惯性权重 ω_{\min} ,

最大飞行速度 v_{\max} , 反向学习因子 c_3, c_4 , 反向学习

代数 L_{times} , 反向学习子种群规模 n , 扰动系数 d_0

输出: 种群历史最优解 P_g

Begin

1. 根据 D 和 N , 随机初始化粒子种群 $X^0 = \{X_1^0, X_2^0, \dots, X_N^0\}$ 及其飞行速度 $V^0 = (V_1^0, V_2^0, \dots, V_N^0)$, 并令 $P_i = B_i = X_i, t = 0$;

2. 计算 X^t 中粒子的适应值, 令 P_{g1} = 最优粒子个体, P_{g2} = 次优粒子个体;

3. 根据算法 1 构造最差粒子子种群 W ;

4. WHILE (不满足停机条件)

5. 按照式(1)、(2)更新 V' 和 X' ;
 6. 计算 X' 中粒子的适应值, 并更新 P_i 、 B_i 、 P_{g1} 和 P_{g2} .
 7. 根据式(4)和(6)对 P_{g1} 进行局部搜索; 根据式(5)对 d_i 进行更新;
 8. IF 满足反向学习条件
 9. 临时调整 v_{\max} ;
 10. 根据式(7)、(2)分别更新 $V'_1 \sim V'_n$ 和 $X'_1 \sim X'_n$;
 11. 按照式(1)、(2)分别更新 $V'_{n+1} \sim V'_N$ 和 $X'_{n+1} \sim X'_N$;
 12. END IF
 13. $t=t+1$;
 14. END WHILE
- End

4 实验与分析

4.1 算法及参数设置

本文选取 7 个改进的 PSO 算法与本文提出的 RLPSO 算法进行比较. 这些算法涵盖了前文介绍的多种改进策略, 如 SPSO2007、FDR-PSO^[13] 对参数进行了优化, wFIPS^[11] 对邻域的拓扑结构进行了优化, RCP SO^[14]、CLPSO^[15] 和 CPSO^[23] 对粒子个体的学习模式进行了优化, DEPSO^[19] 则属于 DE 算法和 PSO 算法的混合. 需要指出的是, 上述改进的 PSO 算法其实都涉及到多种改进策略, 具体内容可参见相应的文献. 在本文的实验中, 这些算法的参数设置同原文献相同, 具体如表 1. RLPSO 算法的参数设置为: 初始最差粒子群体规模为 $M=N/4$, 排异半

径 $R = \sum_{i=1}^D (x_{i \max} - x_{i \min}) / \sqrt{D}$, $x_{i \max}$ 和 $x_{i \min}$ 分别为函数变量第 i 维的取值上限与下限, 学习因子 $c_1 = 1.85$ 、 $c_2 = 2.0$, 最大惯性权重为 ω_{\max} 和最小惯性权重为 ω_{\min} , 最大飞行速度 v_{\max} 初始为变量可行域的 $1/8$, 当进入反向学习阶段时调整为变量可行域的 $1/4$, 反向学习因子 $c_3 = 0.7$ 、 $c_4 = 0.3$, 反向学习代数 $L_{\text{times}} = 20$, 反向学习子种群规模 $n = 0.8 \times N$, 扰动系数初始值 $d_0 = 40$, 反向学习条件为种群最优解连续 20 代未发生变化.

表 1 对比算法的具体参数设置

算法	参数设置
FDR 2003 年	$\omega: [0.4, 0.9]$, $c_1 = c_2 = 1$, $c_3 = 2$
CPSO 2004 年	$\omega: [0.4, 0.9]$, $c_1 = c_2 = 1.4945$
wFIPS 2004 年	$\chi = 0.729$, $\sum c_i = 4.1$
CLPSO 2006 年	$\omega: [0.4, 0.9]$, $c_1 = c_2 = 1.4945$
SPSO 2007 年	$\omega = 0.721$, $c_1 = c_2 = 1.193$, $K = 3$
DEPSO 2010 年	$\chi = 0.729$, $\sum c_i = 4.1$, DE/rand/1/bin, LP=10
RCP SO 2011 年	$\omega = 0.5$, $c_1 = 1.44$, $c_2 = 2$, $M = 10$

4.2 测试函数

为了评价算法的收敛速度和全局搜索能力, 本文对 10 个常用的测试函数进行了对比实验. 其中, $F_1 \sim F_6$ 为多峰函数, 主要用于检验算法的全局搜索能力; $F_7 \sim F_{10}$ 为单峰函数, 主要用于检验算法的收敛速度和求解精度. 函数详细信息见表 2.

表 2 本文选用的 10 个测试函数

	测试函数	搜索空间	最优值 f_{\min}	最优解 x^*
Aceley	$F_1(X) = 20 + e - 20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right)$	$[-32, 32]^D$	0	0.0^D
Alpine	$F_2(X) = \sum_{i=1}^D x_i \sin(x_i) + 0.1x_i $	$[-10, 10]^D$	0	0.0^D
Rastrigin	$F_3(X) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10)$	$[-5.12, 5.12]^D$	0	0.0^D
Schwefel P1.2	$F_4(X) = 418.98288727216249D - \sum_{i=1}^D (x_i \sin(\sqrt{ x_i }))$	$[-500, 500]^D$	0	420.9687^D
Girewank	$F_5(X) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]^D$	0	0.0^D
Salomon	$F_6(X) = -\cos\left(2\pi \sqrt{\sum_{i=1}^D x_i^2}\right) + 0.1 \times \sqrt{\sum_{i=1}^D x_i^2} + 1$	$[-100, 100]^D$	0	0.0^D
Sphere	$F_7(X) = \sum_{i=1}^D x_i^2$	$[-100, 100]^D$	0	0.0^D
Rosenbrock	$F_8(X) = \sum_{i=1}^{D-1} (100(x_{i+1} - x_i^2)^2 + (1 - x_i)^2)$	$[-30, 30]^D$	0	0.0^D
Schwefel P2.22	$F_9(X) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	$[-10, 10]^D$	0	0.0^D
Sum of different power	$F_{10}(X) = \sum_{i=1}^D x_i ^{(i+1)}$	$[-1, 1]^D$	0	0.0^D

本文利用 8 种算法对 $F_1 \sim F_{10}$ 进行了 2 组对比实验,第 1 组实验中,函数变量维数 $D=30$,种群规模 $N=20$,最大评价次数 $maxFEs=1e5$;第 2 组中,函数变量维数 $D=100$,种群规模 $N=30$,最大评价次数 $maxFEs=2e5$. 每组实验重复执行 30 次. 实验

环境为: Intel i3 CPU 2.93 GHz. RAM 4.00 GB, Windows 7 操作系统, MATLAB 2009a. 实验统计结果包括优化结果均值(Mean)和标准差(Std. Dev),具体结果见表 3 和表 4,其中,加粗数值表示对比算法在相应函数上得到的最优结果.

表 3 对比算法对测试函数的优化结果($D=30, N=20$)

测试函数		FDR	CPSO	wFIPS	CLPSO	SPSO	DEPSO	RCPSO	RLPSO
F_1	Mean	7.18e+00	9.99e-05	5.51e-07	9.70e-01	1.65e+01	7.47e+00	6.30e+00	4.99e-08
	Std. Dev	6.14e-01	7.84e-05	2.13e-07	2.89e-01	3.94e-01	3.12e+00	5.83e-01	2.66e-08
F_2	Mean	7.92e+00	1.44e-05	6.64e-03	4.47e+00	2.68e+01	2.47e-01	9.13e+00	3.50e-08
	Std. Dev	2.05e+00	1.19e-05	1.58e-03	1.19e+00	1.66e+00	3.11e-01	2.01e+00	7.66e-08
F_3	Mean	1.38e+02	1.55e-07	7.89e+01	1.05e+02	2.35e+02	7.77e+01	1.51e+02	1.56e+00
	Std. Dev	1.96e+01	3.11e-07	1.73e+01	8.98e+00	1.03e+01	7.50e+01	2.12e+01	1.37e+00
F_4	Mean	4.47e+03	4.34e+02	1.09e+02	9.55e+01	7.00e+03	3.36e+03	6.32e+03	4.91e+03
	Std. Dev	6.75e+02	2.25e+02	1.17e+02	9.80e+01	4.47e+02	9.27e+02	7.77e+02	7.08e+02
F_5	Mean	7.25e+00	5.25e-02	6.98e-04	8.49e-01	1.25e+02	7.04e+00	6.25e+00	1.63e-02
	Std. Dev	1.45e+00	4.48e-02	2.29e-03	1.17e-01	1.23e+01	1.00e+01	1.15e+00	1.98e-02
F_6	Mean	3.54e+00	1.45e+00	2.06e-01	1.27e+00	1.29e+01	1.99e+00	3.07e+00	1.56e+00
	Std. Dev	3.97e-01	3.34e-01	2.30e-02	1.76e-01	7.63e-01	1.17e+00	2.91e-01	3.29e-01
F_7	Mean	7.80e+02	2.44e-07	5.78e-12	1.13e+00	1.43e+04	4.74e+02	5.41e+02	1.23e-13
	Std. Dev	1.84e+02	5.05e-07	3.85e-12	3.42e-01	1.31e+03	6.97e+02	1.27e+02	2.82e-13
F_8	Mean	5.94e+04	4.41e+01	3.22e+01	5.33e+02	1.23e+07	4.66e+05	3.21e+04	2.03e+02
	Std. Dev	2.78e+04	3.66e+01	2.03e+01	1.58e+02	2.66e+06	9.84e+05	9.84e+05	5.86e+02
F_9	Mean	1.09e+01	6.92e-05	1.06e-07	1.71e-01	4.78e+01	1.65e+00	1.11e+01	9.02e-08
	Std. Dev	1.45e+00	3.11e-05	1.47e-07	3.56e-02	3.04e+00	1.94e+00	3.18e+00	2.92e-08
F_{10}	Mean	1.98e-13	1.98e-13	6.62e-45	1.28e-16	2.33e-03	1.89e-04	1.56e-07	1.97e-26
	Std. Dev	9.58e-13	9.58e-13	1.75e-44	1.45e-16	9.28e-04	7.66e-04	1.56e-07	9.34e-26

表 4 对比算法对测试函数的优化结果($D=100, N=30$)

测试函数		FDR	CPSO	wFIPS	CLPSO	SPSO	DEPSO	RCPSO	RLPSO
F_1	Mean	1.17e+01	4.76e-03	2.03e+00	6.05e+00	1.93e+01	8.66e+00	1.08e+01	4.89e-04
	Std. Dev	4.44e-01	1.25e-03	2.79e-01	3.03e-01	1.60e-01	2.29e+00	4.83e-01	3.75e-04
F_2	Mean	6.61e+01	1.91e-03	1.15e+01	3.14e+01	1.49e+02	1.43e+00	7.94e+01	3.75e+00
	Std. Dev	5.24e+00	6.30e-04	3.82e+00	3.18e+00	5.08e+00	7.08e+00	8.57e+00	4.41e+00
F_3	Mean	7.55e+02	5.06e-04	7.24e+02	4.69e+02	1.09e+03	2.13e+02	8.54e+02	1.89e+01
	Std. Dev	5.65e+01	3.08e-04	3.48e+01	1.95e+01	2.44e+01	4.43e+01	4.35e+01	2.09e+01
F_4	Mean	2.12e+04	8.37e+02	1.89e+04	3.37e+03	2.40e+04	1.49e+04	2.85e+04	1.70e+04
	Std. Dev	3.50e+03	3.14e+02	1.17e+03	5.76e+02	7.24e+02	4.89e+03	1.31e+03	2.13e+03
F_5	Mean	1.03e+02	1.34e-02	1.31e+00	1.06e+01	1.05e+03	6.26e+00	8.06e+01	8.71e-03
	Std. Dev	1.36e+01	1.59e-02	5.37e-02	1.11e+00	3.55e+01	5.59e+00	1.12e+01	1.02e-02
F_6	Mean	1.24e+01	4.72e+00	2.94e+00	7.64e+00	1.94e+01	4.42e+00	1.07e+01	3.43e+00
	Std. Dev	5.56e-01	7.73e-01	2.05e-01	4.88e-01	1.04e+00	1.87e+00	9.50e-01	7.75e-01
F_7	Mean	1.11e+04	1.31e-03	3.15e+01	1.05e+03	1.14e+05	5.08e+02	8.69e+03	8.12e-05
	Std. Dev	1.88e+03	8.66e-04	7.30e+00	1.47e+02	5.15e+03	5.32e+02	1.25e+03	7.01e-05
F_8	Mean	3.71e+06	3.13e+02	1.01e+04	1.46e+05	3.16e+08	1.07e+06	2.22e+06	1.52e+02
	Std. Dev	8.00e+05	5.65e+02	2.98e+03	3.74e+04	2.79e+07	1.21e+06	4.44e+05	6.33e+01
F_9	Mean	8.37e+01	1.51e-02	1.38e+00	1.63e+01	2.52e+02	8.59e-01	1.21e+02	2.06e-06
	Std. Dev	6.73e+00	2.21e-03	1.38e-01	1.20e+00	8.89e+00	2.06e+00	7.08e+01	9.97e-06
F_{10}	Mean	2.82e-07	2.42e-09	1.20e-18	6.61e-20	4.19e-02	8.75e-06	4.60e-07	3.36e-19
	Std. Dev	2.87e-07	8.14e-09	2.14e-18	9.24e-20	1.59e-02	7.35e-06	5.29e-07	5.89e-19

4.3 实验结果分析

从表 3 和表 4 可以看出,在低维($D=30$)时, RLPSO 在 F_1, F_2, F_7 和 F_9 上取得了最优的表现,而

CLPSO、wFIPS 和 CPSO 在其他函数上则分别获得了更好的优化效果. 其中, wFIPS 在 F_{10} 以及 CPSO 在 F_3 上的优势较明显. 当函数维数增至 $D=100$ 时,

RLPSO 在 5 个测试函数上具有较明显的优势,且在 F_{10} 上的成绩也只是略差于 CLPSO,而 CPSO 则对其他 3 个函数具有更好的效果. 因此可以说明 RLPSO 算法在高维函数优化时性能较为突出.

此外,为了验证数据的置信度,本文还进一步对 RLPSO 在 $D=100, N=30$ 时与其他 PSO 算法的优

化结果进行了双尾 t -检验,置信水平 $\alpha=0.05$. 检验结果如表 5 所示. 表中, Better、Same 和 Worse 值分别表示 RLPSO 比相应的 PSO 算法在进行函数优化时具有显著更优、相同和显著更劣的函数个数. 从检验结果可以看出, RLPSO 算法比其他算法具有更好的性能. 此外, CPSO 算法也表现出了较好的性能.

表 5 RLPSO 与对比算法的置信 t -检验 ($\alpha=0.05$, 双尾)

		FDR	CPSO	wFIPS	CLPSO	SPSO	DEPSO	RCPSO
F_1	t	143.332	17.909	39.724	109.341	644.420	20.722	121.773
	p	0.000	0.000	0.000	0.000	0.000	0.000	0.000
F_2	t	49.798	-4.660	7.298	27.886	118.525	-1.529	42.996
	p	0.000	0.000	0.000	0.000	0.000	0.132	0.000
F_3	t	66.921	-4.939	95.088	86.203	184.616	21.666	94.685
	p	0.000	0.000	0.000	0.000	0.000	0.000	0.000
F_4	t	5.569	-41.268	1.649	-33.952	17.001	-2.140	25.221
	p	0.000	0.000	0.105	0.000	0.000	0.037	0.000
F_5	t	41.542	1.343	130.671	52.156	161.165	6.116	39.342
	p	0.000	0.018	0.000	0.000	0.000	0.000	0.000
F_6	t	43.801	6.040	-2.097	21.336	64.007	2.873	29.661
	p	0.000	0.000	0.040	0.000	0.000	0.006	0.000
F_7	t	32.462	0.821	23.662	38.934	121.653	5.230	37.861
	p	0.000	0.415	0.000	0.000	0.000	0.000	0.000
F_8	t	25.411	1.550	18.314	21.259	61.989	4.874	27.348
	p	0.000	0.127	0.000	0.000	0.000	0.000	0.000
F_9	t	68.233	37.238	60.231	74.978	156.537	2.291	9.388
	p	0.000	0.000	0.000	0.000	0.000	0.026	0.000
F_{10}	t	5.382	1.629	2.140	-2.480	14.424	6.520	4.763
	p	0.000	0.109	0.037	0.016	0.000	0.000	0.000
	1 (Better)	10	4	8	8	10	8	10
	0 (Same)	0	3	1	0	0	1	0
	-1 (Worse)	0	3	1	2	0	1	0

4.3.1 时间复杂度

本文实验均采用最大评价次数 $maxFEs$ 作为停机条件,其原因在于在算法迭代过程中,对粒子个体的评价是整个算法中最耗时的部分. 因此,以 $maxFEs$ 作为停机条件对各个算法更公平. 需要指出的是,尽管对粒子个体的评价操作最为耗时,但种群在每次迭代时进行其他操作的耗时也会对整个算法的速度有一些影响. 如 RCPSO 每代操作时都要从当前粒子群中选择较优粒子对精英子种群进行替换,而为了保持精英个体多样性而进行的排异距离计算消耗了较多的计算时间,例如,若精英子种群的规模为 M ,则每代更新该子种群最少需要进行 $M \times (M-1)/2 - 1$ 次粒子间的欧式距离计算,此外,RCPSO 每代还对排异半径都进行了计算,因此该算法的耗时最多;DEPSO 的耗时比 SPSO2007 略多,一方面是统计和学习要耗时,但更主要的原因在于其中的差分演化算法 (Differential Evolution Algorithm, DEA) 本身就比 SPSO2007 耗时要多;

FDR-PSO 中每次为粒子选择学习对象时均要计算粒子间的欧式距离,这也会耗费一定的运行时间. RLPSO 除了基本学习外,还有反向学习和最优粒子的局部搜索过程,即每次迭代时,种群历史最优解 P_{g1} 和 P_{g2} 会进行局部搜索,以及 RLPSO 算法要构建用于部分粒子进行反向学习的最差粒子种群. 但与 RCPSO 构建精英子种群不同的是,RLPSO 只在种群初始化时构建一次,在种群进化过程中,该子种群是不变的. 而反向学习的部分粒子只是改变了学习方式,却并未增加额外的时间消耗. 因此,RLPSO 比 DEPSO、RCPSO 和 FDR-PSO 耗时要少. 其他几种算法在每次迭代中所进行的额外操作很少,因此速度都与 RLPSO 相当.

4.3.2 收敛速度

从前面的耗时统计与分析可以知道,RLPSO 算法具有较低的时间复杂性. 而收敛速度是另一个衡量算法求解速度的重要指标,它在解决实际问题时具有重要的意义. 图 1 给出了不同算法在 10 个

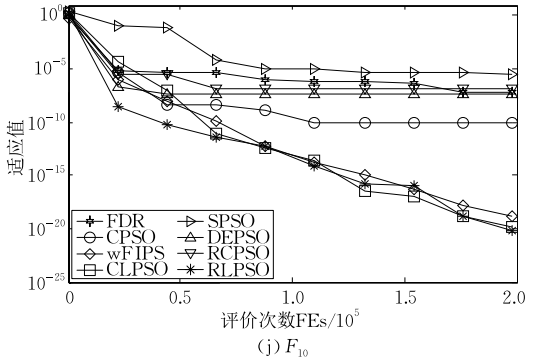
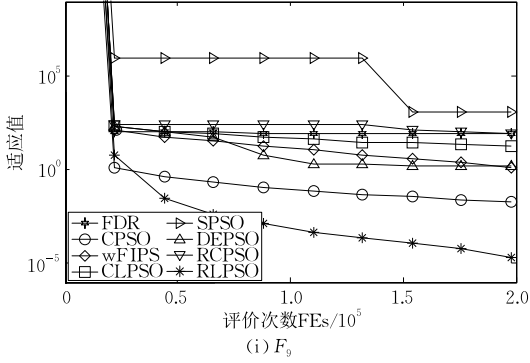
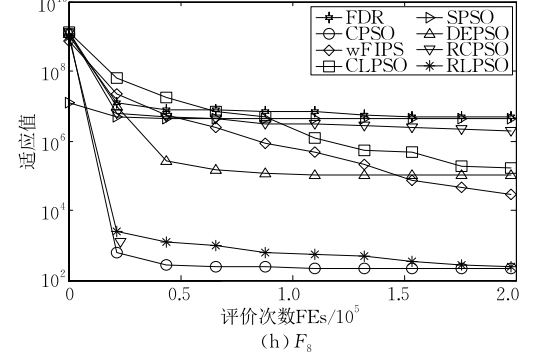
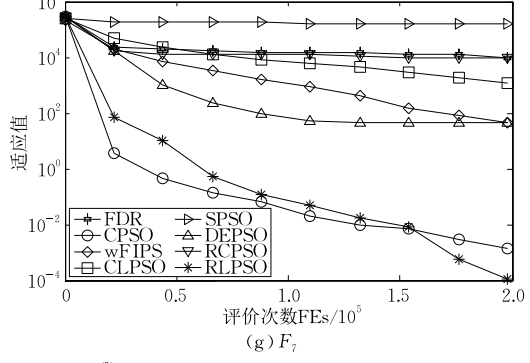
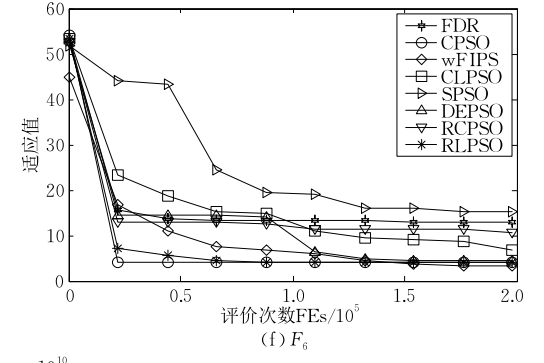
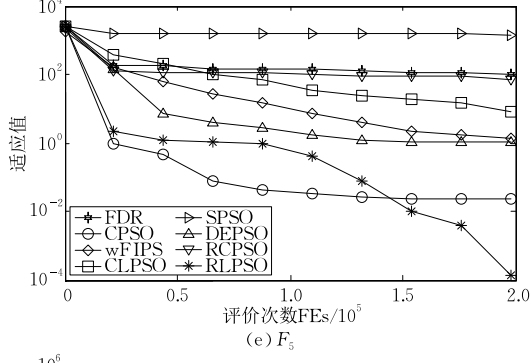
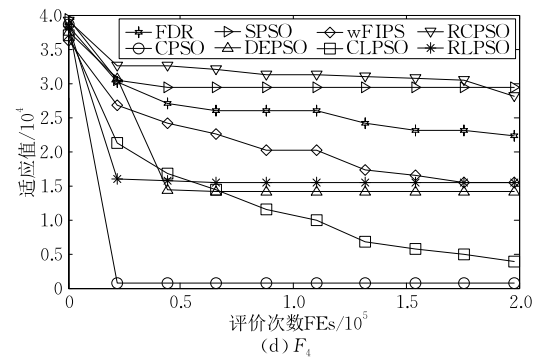
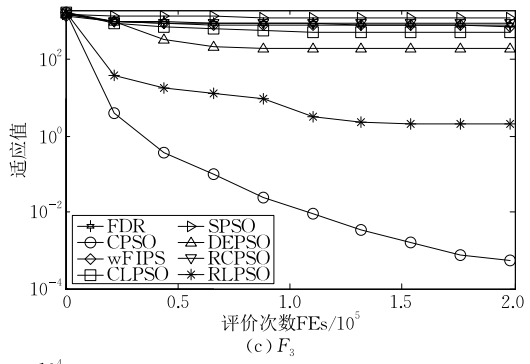
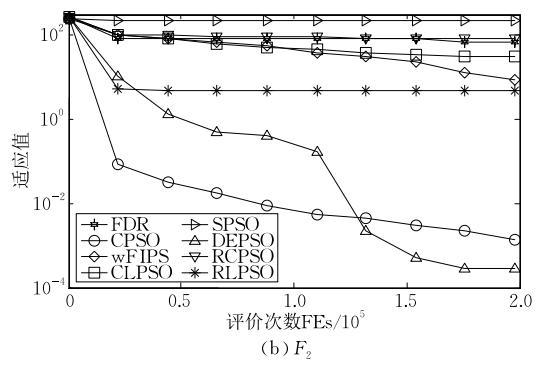
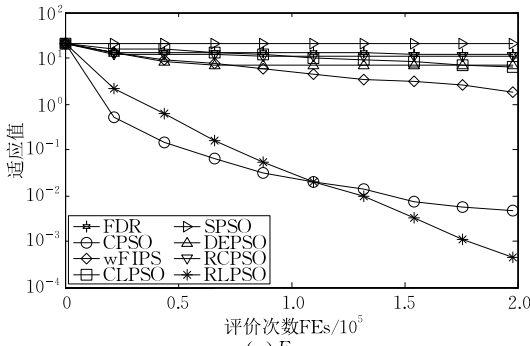


图1 不同 PSO 算法在测试函数上的收敛曲线 ($D=100, N=30$)

测试函数优化上的收敛曲线. 从图中可以看出, RLPSO 和 CPSO 在大部分函数中均具有更快的收敛性, RLPSO 除了在 F_3 上明显劣于 CPSO 外, 在其他函数上均取得了最佳或近似最佳的收敛速度. 因此, 若在求解实际应用问题时, 将算法终止条件改为达到设定精度, 则 RLPSO 具有更好的性能. RLPSO 这种快速的收敛性得益于每次迭代时, 种群历史最优解 P_{g1} 和 P_{g2} 都会进行局部学习. 这说明 RLPSO 中局部学习机制有效地增强了算法的收敛速度.

4.3.3 反向学习机制

保持种群多样性一直是在改进群智能算法性能时的一个关键问题, 因为种群多样性直接关系到算法能否跳出局部最优, 是克服“早熟”收敛的一个关键. RLPSO 算法的反向学习过程的目的就是通过最差粒子位置的牵引, 使一部分可能已陷入局部最优的粒子飞离出该区域. 之所以利用最差粒子位置进行牵引是考虑多峰函数中的峰尖附近总有峰谷存在, 而峰谷可视为一个最差位置, 若能将粒子牵引至峰谷附近, 则该粒子在后续学习过程中寻找到更优

解的可能性会大大增加. 图 2 演示了 F_4 ($D=30$) 在反向学习过程中粒子的飞行过程. 图 2 中, “+”表示种群粒子, “○”表示初始种群中的较劣粒子, 其中“●”表示最劣粒子. 可以看出, 在初始种群中, 尽管这些较劣粒子具有较差的适应值, 但其中一些粒子在某些维上的取值比当前种群最优解更接近于真实最优解 $x=420.9687^D$, 如图 2 中最劣粒子以及与其最邻近的劣粒子在横坐标这一维的取值比其他粒子在该维的取值更优. 因此可以利用这个有益信息牵引一部分粒子跳出局部最优, 改善在该维上的取值. 图 2(a)~(d) 分别表示反向学习开始、反向学习 5 代、15 代及结束时的种群粒子位置分布情况. 可以看出, 在反向学习过程中, 较劣粒子有效地将部分粒子吸引到了其周围. 为了进一步从数值上验证这一思想, 下面将 RLPSO 与移除反向学习机制的算法 (这里称为 LPSO) 进行了比较. 由于反向学习的主要作用是牵引一部分粒子飞离局部最优, 因此这里只选取了部分多峰函数 $F_1 \sim F_4$ 进行测试. 实验参数设置为: $N=30, D=20$, 独立运行 30 次. 实验结果如表 6 所示.

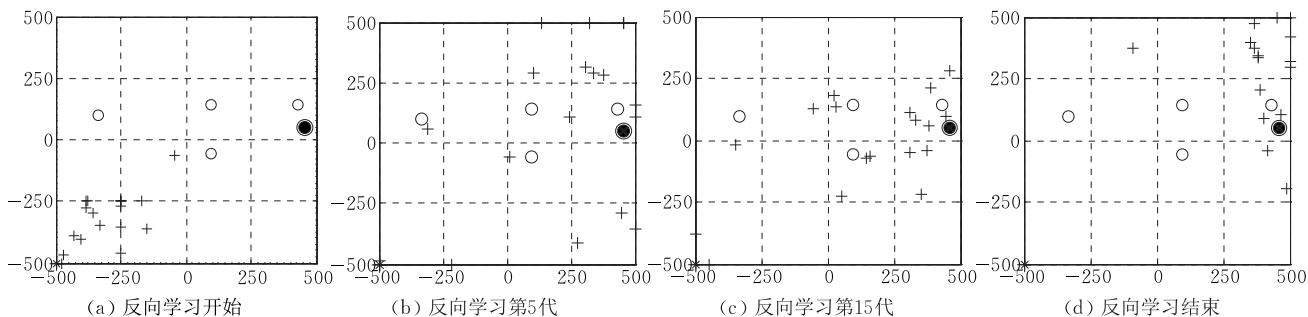


图 2 反向学习过程的粒子分布情况

表 6 LPSO 与 RLPOS 的测试结果

		F_1	F_2	F_3	F_4
LPSO	Mean	3.30e-06	1.20e-06	1.07e+01	1.07e+04
	Std. Dev	9.70e-06	5.61e-06	1.48e+01	2.26e+04
RLPSO	Mean	4.61e-08	3.16e-08	2.71e+00	4.37e+03
	Std. Dev	2.94e-08	5.97e-08	1.13e+00	8.97e+02

从表 6 及前文分析可以看出, 反向学习机制在不增加算法复杂性的基础上不同程度地改善了算法的性能 (一般为 1~2 个数量级). 因此可以保守地认为 PSO 算法在进行多峰函数优化时, 该机制是可以帮助种群跳出局部最优的. 同时, 我们也看出, 该学习机制对于算法的改善还是很有限的, 其原因在于学习机制过于简单, 对较劣粒子所隐含的有益信息的提取和利用还不够, 后续将进一步针对该问题展开研究.

5 结 论

为了缓解 PSO 算法的收敛速度与“早熟”问题之间的矛盾, 进一步提高其优化性能, 本文提出了 RLPSO 算法. 该算法保留了种群历史最优解和历史次最优解, 算法在整个优化过程中依靠这 2 个较优解的差分结果来强化算法的局部搜索能力, 并通过优化进程动态调节差分结果的权重, 以提高搜索精度; 当算法出现“早熟”现象时便会启动反向学习过程, 在该过程中, 部分粒子利用其个体历史最劣解以及初始种群的多个最劣解的吸引合力而逃逸出局部最优, 改善种群的多样性; 当反向学习结束后, 算法再次进入正常的迭代优化过程. 实验结果表明, 本文提出的 RLPSO 算法的优化能力要优于对比算法,

而且“早熟”现象出现时采取的反向学习过程比补充随机粒子个体更能改善算法性能。后续将继续研究 PSO 算法在多峰函数优化时对较差粒子隐含的有益信息的提取和利用,以提高算法在函数优化,尤其是多峰函数优化中的性能。

参 考 文 献

- [1] Kennedy J, Eberhart R C. Particle swarm optimization//Proceedings of the IEEE International Conference on Neural Networks. Piscataway, USA, 1995, 4: 1942-1948
- [2] Chi Yu-Hong, Sun Fu-Chun, Wang Wei-Jun, Yu Chun-Ming. An improved particle swarm optimization algorithm with search space zoomed factor and attractor. Chinese Journal of Computers, 2011, 34(1): 115-130(in Chinese)
(迟玉红, 孙富春, 王维军, 喻春明. 基于空间缩放和吸引子的粒子群优化算法. 计算机学报, 2011, 34(1): 115-130)
- [3] Zhang Jia-Jun. A novel particle swarm optimization for image fusion//Proceedings of the 2nd International Asia Conference on Informatics in Control, Automation and Robotics. Wuhan, China, 2010: 126-129
- [4] Wang Jian, Zhang Xian-Zhou, Zhang Yong, Li Wei. Research on ambiguity resolution of GPS short baseline by using improved particle swarm optimization. Journal of Geodesy and Geodynamics, 2012, 32(4): 148-151(in Chinese)
(王建, 张献州, 张勇, 李伟. 改进粒子群算法求解 GPS 短基线整周模糊度的研究. 大地测量与地球动力学, 2012, 32(4): 148-151)
- [5] Shi Y, Eberhart R C. Fuzzy adaptive particle swarm optimization//Proceedings of the IEEE Congress on Evolutionary Computation. Seoul, Korea, 2001: 1011-1016
- [6] Trelea I C. The particle swarm optimization algorithm: Convergence analysis and parameter selection. Information Processing Letters, 2003, 85(6): 317-325
- [7] Jiang M, Luo Y P, Yang S Y. Stochastic convergence analysis and parameter selection of the standard particle swarm optimization algorithm. Information Processing Letters, 2007, 102(1): 8-16
- [8] Zhang Wei, Wang Hua-Kui. Analysis on parameter selection for stability of particle swarm optimization algorithm. Journal of System Simulation, 2009, 21(14): 4339-4344, 4350 (in Chinese)
(张玮, 王华奎. 粒子群算法稳定性的参数选择策略分析. 系统仿真学报, 2009, 21(14): 4339-4344, 4350)
- [9] Kennedy J. Small worlds and mega-minds: Effects of neighborhood topology on particle swarm performance//Proceedings of the IEEE Congress on Evolutionary Computation. Washington DC, USA, 1999: 1931-1938
- [10] Kennedy J, Mendes R. Population structure and particle swarm performance//Proceedings of the IEEE Congress of Evolutionary Computation. Honolulu, HI, 2002: 1671-1676
- [11] Mendes R, Kennedy J, Neves J. The fully informed particle swarm: Simpler, maybe better. IEEE Transactions on Evolutionary Computation, 2004, 8(3): 204-210
- [12] Suganthan P N. Particle swarm optimizer with neighborhood operator//Proceedings of the IEEE Congress on Evolutionary Computation. Washington DC, USA, 1999: 1958-1962
- [13] Peram T, Veeramachaneni K, Mohan C K. Fitness-distance-ratio based particle swarm optimization//Proceedings of the Swarm Intelligence Symposium. Indianapolis, USA, 2003: 174-181
- [14] Tan Yang, Tang De-Quan, Quan Hui-Yun. Rejection of competition particle swarm optimization. Journal of System Simulation, 2011, 23(12): 2635-2640, 2646(in Chinese)
(谭阳, 唐德权, 全惠云. 一种排斥竞争的粒子群优化算法. 系统仿真学报, 2011, 23(12): 2635-2640, 2646)
- [15] Liang J J, Qin A K, Suganthan P N, et al. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions. IEEE Transactions on Evolutionary Computation, 2006, 10(3): 281-295
- [16] Zeng Yi, Zhu Xu-Sheng, Liao Guo-Yong. Hybrid particle swarm optimization based on neighborhood space. Journal of East China Jiaotong University, 2013, 30(3): 44-49 (in Chinese)
(曾毅, 朱旭生, 廖国勇. 一种基于邻域空间的混合粒子群优化算法. 华东交通大学学报, 2013, 30(3): 44-49)
- [17] Angeline P J. Using selection to improve particle swarm optimization//Proceedings of the IEEE International Conference on Evolutionary Computation. Alaska, USA, 1998: 84-89
- [18] Higashi H, Iba H. Particle swarm optimization with Gaussian mutation//Proceedings of the IEEE Swarm Intelligence Symposium. Indianapolis, USA, 2003: 72-79
- [19] Xin B, Chen J, Peng Z H, et al. An adaptive hybrid optimizer based on particle swarm and differential evolution for global optimization. Science China (Information Sciences), 2010, 53(5): 980-989
- [20] Liu Jun-Fang, Zhang Xue-Ying, Ning Ai-Ping. Hybrid optimization algorithm of PSO and ABC. Computer Engineering and Application, 2011, 47(35): 32-34, 44(in Chinese)
(刘俊芳, 张雪英, 宁爱平. PSO 和 ABC 的混合优化算法. 计算机工程与应用, 2011, 47(35): 32-34, 44)
- [21] Xia Xue-Wen, Liu Jing-Nan, Hu Zhong-Bo. An improved particle swarm optimizer based on tabu detecting and local learning strategy in a shrunk search space. Applied Soft Computing, 2014, 23(1): 76-90
- [22] Parsopoulos K E. Parallel cooperative micro-particle swarm optimization: A master-slave model. Applied Soft Computing, 2012, 12(11): 3552-3579
- [23] van den Bergn F, Engelbrecht A P. A cooperative approach to particle swarm optimization. IEEE Transactions on Evolutionary Computation, 2004, 8(2): 225-239



XIA Xue-Wen, born in 1974, Ph.D., associate professor. His research interests include computational intelligence and its applications.

LIU Jing-Nan, born in 1943, professor, Ph.D. supervisor, academican of Chinese Academy of Engineering. His research

interests include GPS theories, method and data processing.

GAO Ke-Fu, born in 1981, Ph.D., lecturer. His research interests include information security and location based services.

LI Yuan-Xiang, born in 1962, Ph.D., professor, Ph.D. supervisor. His research interests include parallel computation, computational intelligence and its applications.

ZENG Hui, born in 1980, Ph.D. candidate. His research interests include computational intelligence and its applications.

Background

The research work was jointly supported by the National Natural Science Foundation of China (Grant Nos. 41231174, 61165004), the Project Supported by Natural Science Foundation of Jiangxi, China (Grant No. 20151BAB207022, and the Project supported by the East China Jiaotong University (Grant No. 14JG003). The purpose of this paper is to improve the convergence accuracy and speed of PSO algorithm for several optimization issues derived from these projects.

Optimization plays a critical role in human modern life. Nowadays, optimization is used in many aspects of human modern life including engineering, medicine, agriculture and economy. Due to the growing number of optimization problems and their growing complexity, we need to improve and develop theoretical and practical optimization methods. PSO is a simple and computationally inexpensive algorithm inspired by the social behavior of bird flocks and fish schools. However, PSO suffers from premature convergence, especially in high dimensional multimodal functions. These weaknesses have limited wider applications of PSO, so that lots of variant PSO algorithms have been proposed. The improvement strategies, which give birth to more advanced PSO variants, can be largely categorized into following five types: (1) parameter tuning, (2) topology tuning, (3) tuning of learning patterns, (4) multiple-swarm strategies and

(5) hybridization strategies. Note that some complicated PSO variants may adopt multiple strategies above simultaneously.

Based on the above studies, this paper proposes a novel PSO, called RLPSO, where introduces two strategies. The one is reverse-learning behavior in which some useful information of inferior particles that often been ignored by other variant PSO algorithms are used to attract particles to leap out of local optimums and then improve RLPSO's exploration ability. Another one is local exploitation of the best particle. In this behavior, the difference between the best particle and the second-best particle in every generation is adopted to guide the best one to carry out local search which can enhance RLPSO's exploitation ability. The results of experiments show that RLPSO has better global searching ability and higher convergence speed especially in high dimension function. RLPSO will be used in high dimension integer ambiguity resolution of GPS baseline.

Another papers entitled "Image Segmentation with Two-Dimension Threshold Based on Adaptive Particle Swarm Optimization and Data Field" and "Particle Swarm Optimization Algorithm Based on Stable Strategy", belonging to the same project, have been publish on *Computer Science* and *Journal of Computer-Aided Design & Computer Graphics* respectively.