

文章编号: 1001-0920(2013)09-1341-08

基于精英学习的量子行为粒子群算法

章国勇, 伍永刚, 顾 巍

(华中科技大学 水电与数字化工程学院, 武汉 430074)

摘 要: 在分析量子行为粒子群算法中吸引子指导作用的基础上, 引入两种精英学习策略, 提出了基于精英学习的量子粒子群算法(QPSO-EL). 采用动态逼近学习策略对精英个体进行局部更新, 协助其跳出自身局部极值点, 引导种群进行有效搜索; 借鉴群体早熟判断机制对停滞状态下的精英个体空间进行变尺度混沌扰动, 增大种群全局搜索空间, 有效平衡了算法的局部和全局搜索能力. 典型函数的仿真结果表明, 该算法具有收敛速度快、求解精度高的特点.

关键词: 量子行为粒子群; 精英学习; 动态逼近搜索; 变尺度混沌扰动

中图分类号: TP301.6

文献标志码: A

Quantum-behaved particle swarm optimization algorithm based on elitist learning

ZHANG Guo-yong, WU Yong-gang, GU Wei

(College of Hydropower and Information Engineering, Huazhong University of Science and Technology, Wuhan 430074, China. Correspondent: WU Yong-gang, E-mail: ygangwu@163.com)

Abstract: The local attractor point in the quantum-behaved particle swarm optimization algorithm plays an important role in determining the convergence process of population. Therefore, a quantum-behaved particle swarm optimization algorithm based on two elitist learning strategies(QPSO-EL) is presented. In this method, the dynamic-approximation search strategy is exerted on the elitist particles to avoid them running into local optima and provides a good guidance for the population. While the algorithm is found to be in a dead state according to the premature judgment mechanism, the mutative-scale chaotic perturbation is used to exhibit a wide range exploration and keep the balance of exploration and exploitation. The experiment results on classic functions demonstrate the global convergence ability and the search accuracy of the proposed method.

Key words: quantum-behaved particle swarm optimization; elitist learning; dynamic-approximation research; mutative-scale chaotic perturbation

0 引 言

粒子群优化(PSO)算法是一种全局优化进化算法^[1], 由于其具有流程简单、可调参数少、收敛速度快等优点, 已成功应用于数值优化、图像处理等领域. 但标准 PSO 算法已被证明在求解复杂优化问题时并不能收敛于全局最优解^[2], 这极大地限制了 PSO 的应用前景. 受量子力学的启发, Sun 等^[3]提出一种具有量子行为的粒子群优化算法(QPSO). 相比传统的 PSO 算法, QPSO 算法利用种群中所有粒子的位置建立分布概率模型, 通过随机采样操作完成了对群体的更新, 使得粒子能够以某一概率出现在整个可行搜索空间的任意位置^[4], 因而, QPSO 算法的全局搜索性能远远

优于标准 PSO 算法.

与 PSO 算法一样, QPSO 算法在搜索过程中也容易因迭代后期粒子多样性减小而早熟收敛^[4-17]. 对此, 国内外已有许多针对 QPSO 算法的改进工作. 文献 [5-6] 通过对参数进行自适应控制来提高算法优化性能; 文献 [7-10] 提出采用多种变异操作方法来增强 QPSO 跳出局部最优点的能力; 文献 [11-13] 通过引入多种其他优化机制来改善 QPSO 的寻优能力. 基于 QPSO 中吸引子的指导作用^[4], 文献 [14] 引入遗传算法中的选择操作来产生吸引子坐标, 增加了种群多样性; 文献 [15] 引入微分进化算子来增加粒子随机性; 文献 [16] 提出基于高斯分布的方法来产生局部吸引点; 文献 [17] 采用基于动态随机变量和自身最优粒

收稿日期: 2012-05-03; 修回日期: 2012-07-16.

基金项目: 国家科技支撑计划项目(2009BAC56B03).

作者简介: 章国勇(1989-), 男, 博士生, 从事智能计算、电力系统优化调度的研究; 伍永刚(1963-), 男, 教授, 博士生导师, 从事电力系统优化调度、自动发电优化控制等研究.

子加权平均的策略更新吸引子坐标.

上述针对吸引子的改进均是基于改变吸引子表达式来改善 QPSO 的寻优性能,一定程度上降低了吸引子位置的随机性. 本文在保持吸引子位置多样性不变的前提下,通过引入精英学习策略更新吸引子位置,引导粒子在保持多样性的同时向全局最优值进行有效靠拢;同时针对进化过程中不同种群状态提出了两种精英学习策略,并对精英个体的选取和学习策略的参数进行了分析,以求达到算法在局部搜索和全局空间搜索的平衡.

1 QPSO 算法

在一个 D 维的目标搜索空间中,由 M 个粒子组成一个群体,在第 t 时刻将第 i 个粒子的位置表示为一个 D 维向量 $(x_{i1}(t), x_{i2}(t), \dots, x_{iD}(t))$. 个体最好的位置为 $P_i(t) = (P_{i1}(t), P_{i2}(t), \dots, P_{iD}(t))$, 群体全局最好的位置为 $P_g(t) = (P_{g1}(t), P_{g2}(t), \dots, P_{gD}(t))$, 其中 g 为全局最优位置下标. 根据 Clerc 等^[18]对 PSO 算法中粒子收敛分析,必然存在以点 P_d 为中心的某种形式的吸引势,且满足以下条件:

$$P_d = \varphi_{1d}P_{id} + \varphi_{2d}P_{gd}/(\varphi_{1d} + \varphi_{2d}), \quad (1)$$

$$\varphi_{ij}(t) = c_1r_{1ij}(t)/(c_1r_{1ij}(t) + c_2r_{2ij}(t)). \quad (2)$$

将粒子移至量子力学空间中,采用波函数 $|\psi(r, t)|$ 来描述粒子的位置,由薛定谔方程决定粒子的状态变化过程. 假设每个粒子都在一个以各自吸引子为中心的 δ 势阱中移动,通过求解薛定谔方程得出粒子在各自 δ 势阱中的概率密度函数为

$$Q(y) = |\psi(y)|^2 = |\psi(x-p)|^2 = \frac{1}{L}e^{-2|x-p|/L}, \quad (3)$$

其中 L 为 Delta 势阱的特征长度,它决定了粒子的搜索范围. 在势场中,粒子的运动位置服从上述概率密度函数,其位置是不确定的,但在实际应用中,任意确定时刻粒子必须具有确定的位置,故采用蒙特卡罗进行随机模拟得到运动方程,即粒子的进化方程如下:

$$x = P_d \pm \frac{L}{2} \ln(1/u), u \in U(0, 1), \quad (4)$$

$$L = 2\beta \cdot |m_{\text{best}} - x_{ij}(t)|. \quad (5)$$

其中: m_{best} 为所有个体最好位置平均,即

$$m_{\text{best}} = \left(\sum_{i=1}^M P_{i1}/M, \sum_{i=1}^M P_{i2}/M, \dots, \sum_{i=1}^M P_{iD}/M \right); \quad (6)$$

参数 β 为收缩-扩张系数. 为保证算法收敛, β 必须满足 $\beta < 1.782$ ^[3], 通常采用随进化代数 t 线性改变的方式控制,如下所示:

$$\beta = \frac{(\beta_1 - \beta_2) \times (G_{\text{max}} - t)}{G_{\text{max}}} + \beta_2. \quad (7)$$

其中: β_1 和 β_2 分别为 β 的开始值和结束值, G_{max} 为最大进化代数.

2 基于精英学习的量子行为粒子群算法

2.1 算法机理描述

通过分析 QPSO 进化方程(4)可知,在 QPSO 算法中,粒子个体的每一维变量 x_{ij} 服从一个以吸引子 P_d 为中心、 L 为范围的概率分布^[4]. 将式(1)代入(4),得到粒子进化方程为

$$x_{id} = \varphi_{id}(P_{id} - P_{gd}) + P_{gd} \pm \frac{L}{2} \cdot \ln(1/u). \quad (8)$$

分析式(8)可以看出,由于粒子对 P_{gd} 的收敛性,当 P_{id} 和 P_{gd} 非常接近时粒子将集中在全局最优解附近,出现群体聚集现象. 若此时 P_{gd} 在全局最优解的领域内,则粒子向 P_{gd} 的收敛可增强对该领域的局部搜索,从而提高最优解的精度^[14];但如果 P_{gd} 位于局部最优解领域内,并且距离全局最优解较远时,粒子以较大的概率出现在局部最优解附近,很容易导致算法早熟. 由此可以看出,吸引子 P_d 的坐标直接决定了该粒子分布区域的中心位置.

为克服上述早熟收敛的现象,本文充分利用吸引子的引导作用,针对不同种群状态引入两种精英学习策略来更新种群中吸引子位置,形成基于精英学习的量子行为粒子群(QPSO-TL)算法,算法结构如图1所示. 通过精英选择策略选取个体进入精英个体空间,将动态逼近学习策略施加于精英个体空间,提高算法局部搜索能力,改善个体进化中心位置,指引种群有效搜索;当种群出现聚集现象时,通过变尺度混沌扰动使得 P_{id} 和 P_{gd} 保持一定距离,扩大种群的搜索范围,降低算法早熟收敛概率,最终寻求算法在全局和局部搜索的有效平衡.

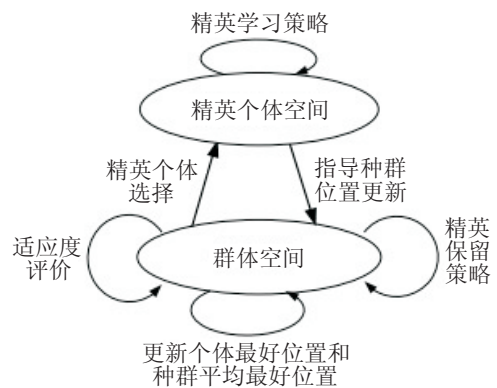


图 1 QPSO-TL 算法结构图

2.2 混沌初始化种群

混沌是自然界中广泛存在的一种非线性现象,利用混沌序列具有的遍历性、随机性和规律性特点^[19]进行种群的初始化分布,既不改变 QPSO 算法初始化时具有的随机性本质又可大大加强算法搜索的多样性. 本文采用的帐篷映射表达式如下:

$$\begin{cases} Z_{i+1} = 2Z_i, 0 \leq Z_i \leq 0.5; \\ Z_{i+1} = 2(1 - Z_i), 0.5 \leq Z_i \leq 1. \end{cases} \quad (9)$$

设种群规模为 n , 变量的维数为 D 维. 首先随机给出 D 个初始参数序列 $r^k = [r_1^k, r_2^k, \dots, r_D^k]$, 将其代入式 (9), 经过 n 次迭代运算产生 D 条运动轨迹, 每条运动轨迹有 n 个序列, 构成了 $n \times D$ 的初始种群.

2.3 精英学习策略

精英学习策略的第1步是精英个体的选择. 由 QPSO 进化方程知, 个体的迭代过程取决于自身概率密度函数中心吸引子的位置和种群分布范围 L . 为改善分布中心吸引子坐标, 将个体按适应度值进行排序, 选取当前种群最优前 m 个粒子和全局最优向量一起进入精英个体空间进行动态逼近学习更新. 当种群出现聚集时, 已有研究表明, 利用混沌特有的特性进行小空间扰动具有较好的效果^[10,15], 但当搜索空间较大时其效果却不能令人满意. 为此, 本文利用基于帐篷映射的变尺度混沌扰动机制对精英个体空间进行扰动, 扩大个体搜索空间, 增加种群多样性.

2.3.1 动态逼近学习策略

引入动态逼近搜索机制对当前进化过程中精英个体进行局部寻优, 将更新后的最优解向量作为量子粒子群算法的全局最优解, 种群在新的最优位置的引导下进行有效搜索, 动态逼近学习策略表达式如下:

$$x_j^{k+1} = x_j^k \pm A \cdot U \cdot r x_j, \quad (10)$$

$$U = e^{-m \cdot \text{Gen} / G_{\max}}, \quad (11)$$

$$r x_j = (x_{j \max} - x_{j \min}) \cdot N(0, 1). \quad (12)$$

其中: A 为动态逼近系数, 用于控制精英个体在自身更新过程中不断缩小搜索步长, 初始值取 1; U 为自适应系数, 用于确定主循环迭代过程中每次调用动态逼近学习策略时精英个体搜索的初始步长值; m 为进化系数; $r x_j$ 为搜索步长; $x_{j \max}$ 和 $x_{j \min}$ 为变量搜索区间最大值和最小值.

为提高精英个体的局部搜索能力, 改善种群进化方程中吸引子的位置, 选取了局部搜索能力较强的高斯分布函数来控制个体搜索步长. 考虑到算法的计算效率, 在进化初期, 由于算法的收敛速度较快, 以小概率对精英个体进行局部搜索; 而在进化后期, 以接近 1 的概率调用动态逼近搜索, 调用精英学习策略的概率 y 如下:

$$y = 1 - e^{-m \cdot \text{Gen} / G_{\max}}. \quad (13)$$

综上所述, 动态逼近学习策略实现流程如下:

Step 1: 按式 (11), (12) 计算当前逼近搜索步长值.

Step 2: 对精英个体中变量 x_j^k 按式 (10) 进行局部更新, 其余的 $D - 1$ 个变量保持不变.

Step 3: 评价 $x^{k+1} = [x_1^k, x_2^k, \dots, x_D^k]$ 的适应度, 若优于 x^k , 则将 x^k 替换为 x^{k+1} , 并保存该搜索步长转 Step 2 继续运行, 直到最大逼近次数, 否则转 Step 4.

Step 4: $j = j + 1$, 若 $j = D$, 则结束本次局部搜索转 Step 5, 否则转 Step 2 继续对下一变量进行搜索.

Step 5: $k = k + 1$, 若 k 大于指定的搜索次数则退出程序, 否则按下式:

$$A = \alpha \cdot A \quad (14)$$

缩小搜索步长 (α 为动态逼近因子), 并按新的搜索步长转 Step 1 继续执行.

由上述分析可以发现, 随迭代次数的增加, 搜索也越来越精细. 当 α 取值较小时, 算法搜索精度较高但不能保证跳出局部最优解; 而当 α 取值较大时, 搜索步长变幅较小, 局部搜索步长较大, 算法易跳过最好解所在区域. 本文采用自适应线性减小的方式确定 α 的值, 表达式如下:

$$\alpha = \frac{(\alpha_1 - \alpha_2) \times (G_{\max} - t)}{G_{\max}} + \alpha_2. \quad (15)$$

2.3.2 种群停滞判断与变尺度混沌学习策略

进化停滞判断是早熟处理的基础, 通常采用平均适应度方差或平均粒距来判断, 相应表达式如下:

$$\delta^2 = \frac{1}{n} \sum_{i=1}^n \left(\frac{f_i - f_{\text{avg}}}{f} \right)^2, \quad (16)$$

$$D(t) = \frac{1}{n \cdot l} \cdot \sum_{i=1}^n \sqrt{\sum_{j=1}^D (p_{ij} - \bar{p}_j)^2}. \quad (17)$$

其中: n 为种群规模; l 为搜索空间最大对角长度; f_i 为个体 i 的适应度值; f_{avg} 为平均适应度值; f 为归一化因子, 且有 $f = \max[1, \max |f_i - f_{\text{avg}}|]$; p_{ij} 为个体 i 第 j 维变量的值; \bar{p}_j 表示所有个体第 j 维变量的平均值.

由式 (16) 和 (17) 可以看出, 种群平均适应度方差是从函数值方面反映粒子分布情况, 而平均粒距是从空间角度反映各个个体相互之间的分布离散程度^[19]. 当粒子群陷入早熟收敛或者达到全局收敛, 粒子将聚集在搜索空间的一个或几个特定的位置, 当这几个极小点相差不大时种群的适应度方差将很小而平均粒距却很大, 因此, 采用平均粒距来描述种群多样性是不够完善的. 本文采用平均适应度方差 δ^2 对种群聚集现象进行判断, 采用变尺度混沌策略对精英个体空间进行扰动变异操作, 具体流程如下:

Step 1: 利用混沌对初值敏感的特点, 取 D 个有微小变异的初值 $z_j^k \in [0, 1]$. 其中 D 为个体中的变量维数, k 为第 k 次混沌搜索次数.

Step 2: 将 z_j^k 映射到 x_j 的取值区间 $[x_{j \min}^k, x_{j \max}^k]$, 初始取值 $x_{j \min}^0 = x_{j \min}$, $x_{j \max}^0 = x_{j \max}$, 得

到混沌搜索变量值

$$x_j^k = x_{j \min}^k + z_j^k(x_{j \max}^k - x_{j \min}^k). \quad (18)$$

Step 3: 评价 $x^k = [x_1^k, x_2^k, \dots, x_D^k]$ 的适应度值, 若优于 x , 则将 x^k 替代 x , 否则继续.

Step 4: $k = k + 1$, 采用式 (9) 更新混沌变量.

Step 5: 若 k 大于最大混沌搜索迭代次数, 则终止迭代, 否则重复 Step 2~Step 4, 直到一定次数内 (本文取 10 次) 最优解保持不变为止, 执行 Step 6.

Step 6: 通过下式:

$$x_{j \min}^{k+1} = x_j^* - \gamma \cdot (x_{j \max}^k - x_{j \min}^k), \quad (19)$$

$$x_{j \max}^{k+1} = x_j^* + \gamma \cdot (x_{j \max}^k - x_{j \min}^k), \quad (20)$$

动态缩小各变量的搜索范围, 转 Step 2 继续执行. 其中: x_j^* 为变量 x_j 的当前最优位置, γ 为变尺度系数. 为使新搜索范围不致越界, 需做以下处理:

当 $x_{j \min}^{k+1} > x_{j \min}^k$ 时,

$$x_{j \min}^{k+1} = x_{j \min}^k;$$

当 $x_{j \max}^{k+1} < x_{j \max}^k$ 时,

$$x_{j \max}^{k+1} = x_{j \max}^k.$$

相比传统的混沌量子行为粒子群, 上述变尺度混

沌扰动通过动态改变混沌扰动步长可以有效地扩大个体进化空间范围, 使得 p_{id} 和 p_{gd} 保持一定距离, 减少因多样性下降而出现早熟收敛的可能性.

3 仿真实验结果与分析

3.1 仿真设计

仿真中引入 6 个标准函数 (见表 1) 来测试 QPSO-EL 的总体性能, 其理论最小值均为 0. 为了测试和比较算法的性能, 本文设计了 7 组算法对比测试实验, 包括标准 QPSO^[14] 和现有的相关改进 QPSO 算法, 如: CMQPSO^[20]、RQPSO^[8]、IGSO^[12]、CQPSO-DE^[15] 和 QPSO-RS^[14]. 各算法中的参数参考相应文献, 本文 QPSO-EL 算法的参数设置如下: 扩张-收缩因子 β 从 1.0 到 0.5 线性减小, 动态逼近因子 α 从 0.3 到 0.1 线性减小, 变尺度因子 γ 取 0.1, 多样性度量值阈值 $\Delta\delta$ 取 0.015, 自适应进化系数取 3.0, 精英个体数取 1, 变尺度混沌扰动 100 次, 动态逼近迭代搜索 8 次. 参照文献 [14] 的实验设置, 分别测试 10, 20, 30 维的情况, 对应算法的最大迭代次数为 1 000, 1 500, 2 000, 粒子个体为 20, 40, 80. 对于 Schaffer 函数, 维数固定为 2, 最大迭代次数为 2 000. 每个实例运行 30 次所得到的平均最优值如表 2~表 7 所示.

表 1 标准测试函数

测试函数	函数表达式	搜索范围	初始化范围	函数类型
Sphere	$f_1(x) = \sum_{i=1}^n x_i^2$	[-100,100]	[50,100]	单峰
Tablet	$f_2(x) = 10^6 x_1^2 + \sum_{i=2}^n (x_i^2)^2$	[-100,100]	[50,100]	单峰
Rosenbrock	$f_3(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	[-30,30]	[15,30]	单峰
Rastrigrin	$f_4(x) = \sum_{i=1}^n (x_i^2 - 10 \cos(2\pi x_i) + 10)$	[-5.12,5.12]	[2.56,5.12]	多峰
Ackley	$f_5(x) = 20 + e - 20e^{-\frac{1}{5} \sqrt{\frac{1}{n} \sum_{i=1}^n x_i^2} - \frac{1}{n} \sum_{i=1}^n \cos(2\pi x_i)}$	[-32,32]	[16,32]	多峰
Schaffer	$f_6(x) = 0.5 + \frac{\sin^2(\sqrt{x^2 + y^2}) - 0.5}{(1.0 + 0.001(x^2 + y^2))^2}$	[-100,100]	[30,100]	多峰

表 2 Sphere 函数测试结果

m	Dim	G_{\max}	QPSO ^[14]	RQPSO ^[8]	CQPSO-DE ^[15]	CMQPSO ^[20]	QPSO-RS ^[14]	IQPSO ^[12]	QPSO-EL
20	10	1 000	3.197 9e-043	1.741 2e-7	-	5.542e-28	4.947 8e-44	4.68e-30	1.666 2e-208
	20	1 500	1.919 7e-024	1.851 7e-6	-	2.255e-24	1.815 5e-24	1.57e-12	2.327 8e-28
	30	2 000	7.373 6e-015	6.011 8e-6	-	3.459e-15	4.172 5e-15	1.53e-10	1.037 1e-22
40	10	1 000	2.760 0e-076	1.503 5e-7	3.475 8e-113	7.483e-19	9.904 9e-76	3.48e-3	9.678 5e-191
	20	1 500	3.197 9e-043	1.345 4e-6	1.533 1e-75	1.970e-18	3.067 3e-44	6.37e-2	1.474 2e-213
	30	2 000	7.373 6e-015	6.011 8e-6	-	3.459e-15	4.172 5e-15	1.65e-20	1.224 1e-238
80	10	1 000	2.560 7e-103	1.077 9e-7	1.104 1e-129	4.565e-17	2.389 6e-103	7.33e-41	2.596 6e-177
	20	1 500	1.411 3e-068	1.345 4e-6	-	3.763e-16	1.555 8e-68	3.17e-24	4.940 1e-202
	30	2 000	6.576 4e-050	4.172 3e-6	-	4.903e-12	4.157 4e-50	2.16e?21	3.350 2e-227

表 3 Tablet 函数测试结果

m	Dim	G_{\max}	QPSO ^[14]	RQPSO ^[8]	CQPSO-DE ^[15]	CMQPSO ^[20]	QPSO-RS ^[14]	IQPSO ^[12]	QPSO-EL
20	10	1000	2.399 8e-21	-	-	7.665e-43	1.963 5e-53	-	5.838 5e-56
	20	1500	3.060 3e-10	-	-	9.820e-29	1.484 2e-29	-	6.690 1e-43
	30	2000	2.395 3e-08	-	-	3.045e-14	5.846 7e-19	-	2.225 9e-43
40	10	1000	2.778 3e-27	-	-	4.672e-47	1.002 5e-97	-	4.356 2e-270
	20	1500	6.801 5e-17	-	-	5.831e-35	1.180 2e-56	-	1.009 0e-55
	30	2000	1.326 3e-11	-	-	9.873e-27	9.017 1e-38	-	1.211 1e-52
80	10	1000	2.589 5e-28	-	-	5.643e-66	8.710 0e-132	-	3.954 3e-266
	20	1500	3.999 6e-17	-	-	1.064e-52	6.924 0e-90	-	5.442 3e-75
	30	2000	7.617 7e-12	-	-	7.886e-41	2.582 7e-64	-	4.356 9e-59

表 4 Rosenbrock 函数测试结果

m	Dim	G_{\max}	QPSO ^[14]	RQPSO ^[8]	CQPSO-DE ^[15]	CMQPSO ^[20]	QPSO-RS ^[14]	IQPSO ^[12]	QPSO-EL
20	10	1000	9.565 7	47.6904	-	9.552	6.912 0	15.04	7.399 2
	20	1500	82.429 4	70.7450	-	30.982	35.504 8	38.95	1.499 9
	30	2000	98.794 8	103.7322	-	113.366	72.915 0	25.97	2.343 2
40	10	1000	8.998 3	20.1872	5.5812	8.031	5.193 6	4.17	0.039 3
	20	1500	40.744 9	46.8270	15.6854	19.553	19.482 1	3.79	9.484 6e-3
	30	2000	43.558 2	78.5551	-	31.982	36.737 3	11.07	1.024 8e-4
80	10	1000	6.831 2	13.4288	5.2616	5.673	5.099 0	0.76	5.417 7e-3
	20	1500	33.528 7	29.8257	-	17.036	17.096 8	2.89	7.274 8e-5
	30	2000	44.594 6	54.1137	-	25.775	27.173 7	10.46	4.024 3e-6

表 5 Rastrigrin 函数测试结果

m	Dim	G_{\max}	QPSO ^[14]	RQPSO ^[8]	CQPSO-DE ^[15]	CMQPSO ^[20]	QPSO-RS ^[14]	IQPSO ^[12]	QPSO-EL
20	10	1000	4.003 2	4.4489	-	4.995	3.736 5	3.55	0
	20	1500	15.064 8	15.9715	-	11.521	14.963 1	9.27	0
	30	2000	28.302 7	27.6414	-	23.994	23.762 2	12.47	6.938 8e-19
40	10	1000	2.645 2	3.2081	0	3.017	2.015 4	1.38	0
	20	1500	11.310 9	10.5817	0	15.680	11.927 1	6.08	0
	30	2000	18.927 9	20.9748	-	21.025	15.442 8	3.47	0
80	10	1000	2.261 7	2.0922	0	1.396	1.469 2	0.67	0
	20	1500	8.412 1	8.4794	-	8.096	7.747 6	4.77	0
	30	2000	14.857 4	16.4016	-	14.271	14.670 5	8.79	0

表 6 Ackley 函数测试结果

m	Dim	G_{\max}	QPSO ^[14]	RQPSO ^[8]	CQPSO-DE ^[15]	CMQPSO ^[20]	QPSO-RS ^[14]	IQPSO ^[12]	QPSO-EL
20	10	1000	15.203 3	-	-	4.907	1.601 1	-	7.993 6e-15
	20	1500	18.608 3	-	-	10.092	3.667 8	-	5.223 8e-15
	30	2000	20.148 7	-	-	15.663	3.987 6	-	2.270 6e-14
40	10	1000	14.824 2	-	4.263 3e-15	5.986	3.128 3e-12	-	3.552 6e-16
	20	1500	9.523	-	1.421 1e-15	8.023	5.390 5e-9	-	6.012 1e-16
	30	2000	20.528 1	-	-	12.890	3.511 2e-7	-	2.228 5e-16
80	10	1000	12.375 3	-	1.421 1e-15	1.053	3.028 7e-13	-	2.664 5e-16
	20	1500	17.408 3	-	-	1.995	7.211 1e-6	-	2.792 1e-15
	30	2000	18.615 9	-	-	5.257	1.142 5e-7	-	6.029 7e-15

表 7 Schaffer 函数测试结果

m	Dim	G_{\max}	QPSO ^[14]	RQPSO ^[8]	CQPSO-DE ^[15]	CMQPSO ^[20]	QPSO-RS ^[14]	IQPSO ^[12]	QPSO-EL
20	10	2000	0.001 6	0.001 433	-	2.207e-4	3.642 1e-6	-	1.775 6e-13
40	20	2000	5.830 8e-4	2.130 3e-8	-	1.620e-7	2.097 0e-8	-	3.093 7e-18
80	30	2000	7.569 5e-9	3.597 9e-9	-	9.433e-9	6.948 7e-9	-	4.322 1e-19

3.2 仿真结果分析

从上述测试结果可以看出,除了 Tablet 函数测试中部分结果不如 QPSO-RS 外,在大多数实例中 QPSO-TL 的结果都优于其他算法. 针对函数 f_1, f_2 , 由于函数形式比较简单, 函数最优值所在的区域比较平坦, 大多数算法能获得较好的收敛效果, 但随着测试函数维数的增加, 部分算法出现寻优精度不高的问题. 对于函数 f_3 而言, 虽然也是单峰问题, 但由于函数最优值附近有陡峭的区域, 使得算法极易陷入局部最优. 随着维数的增加, 大多算法出现早熟收敛的情况, QPSO-EL 算法的测试结果表现出了较好的全局寻优性能. 对于多峰函数 $f_4 \sim f_6$ 而言, 标准的 QPSO 算法表现出较差的全局寻优能力, 本文改进算法与文献 [15] 中 CQPSO-DE 相当, 且随着函数维数的增加仍能表现出较好的收敛精度, 充分说明了 QPSO-EL 算法求解复杂多峰问题的优越性.

为进一步分析两种精英学习策略的必要性, 本文选取了 3 个标准函数: Tablet、Rosenbrock、Rastrigrin 对标准的量子行为粒子群 (QPSO)、单独基于动态逼近学习的 QPSO-DA 和包含两种精英学习策略的 QPSO-EL 算法的性能进行测试. 图 2 给出了在粒子数为 40、变量维数为 30、迭代 1000 次的情况下 3 种不同学习策略下最优值的进化曲线.

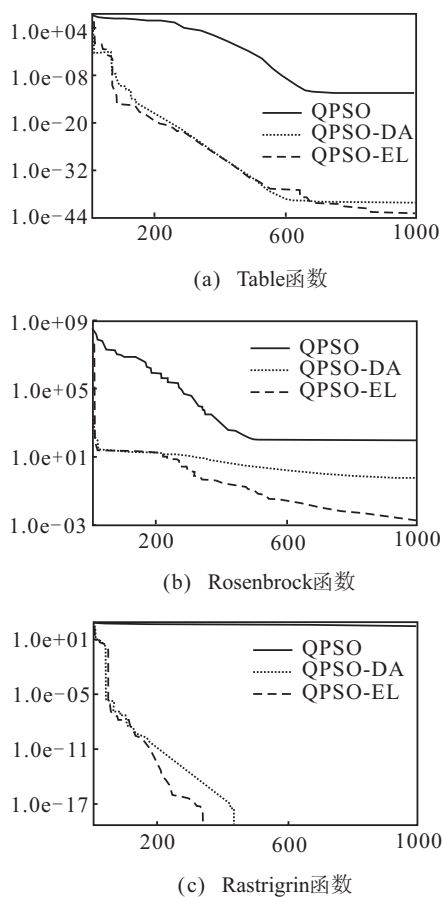


图 2 不同学习策略下算法最优解收敛过程

从图 2 可以看出, 动态逼近学习策略的加入有效提高了算法搜索效率, 相比标准 QPSO 函数最优适应度值得到较大的改善. 但随着迭代过程的推进, 种群的多样性不断减少, 出现聚集现象. 对于较简单的 Tablet 函数, QPSO-DA 的最优适应度值依然得到改善, 但对于存在较多局部收敛点的 Rosenbrock 函数而言, QPSO-DA 进化后期开始出现早熟收敛的情况, 种群最优适应度值衰减缓慢. 此时, 变尺度混沌扰动策略的作用被充分凸显出来. 从图 2 中可以看出, QPSO-EL 种群仍然能够保持较大的搜索空间, 个体进化过程中搜索范围得以保持, 函数最优适应度值较 QPSO-DA 有较大的改进. 总的看来, QPSO-EL 在提高 QPSO 局部搜索能力的同时保持了种群较强的全局搜索能力.

3.3 参数分析

QPSO-EL 中重要的参数包括动态逼近因子 α 、扩张-收缩因子 β 、多样性度量阈值 $\Delta\delta$ 等. β 在相关文献中已有讨论^[5-6], 本文仅讨论 $\Delta\delta$ 和 α 对算法性能的影响.

3.3.1 多样性度量阈值 $\Delta\delta$ 的影响

$\Delta\delta$ 的取值直接影响了 QPSO-TL 调用变尺度混沌变异的时机. 取初始种群规模 40, 保持其他参数不变, 停滞阈值分别取 0.005, 0.01, 0.015, 0.02. 对测试函数 f_2, f_3 和 f_4 分别进行 20 次优化. 函数的维度均为 30, 迭代 1000 次优化结果的最优值、最差值和平均值如表 8 所示. 可以发现, $\Delta\delta$ 较小时算法容易陷入早熟收敛; 而 $\Delta\delta$ 较大时, 则使得粒子失去较优的局部搜索精度. 仿真结果表明, 对于单模函数由于函数较简单, 借助 QPSO 自身较强的全局寻优能力, $\Delta\delta$ 越小最优解精度越高; 而对于多模函数, 由于迭代过程中存在较多的早熟收敛点, 当停滞阈值 $\Delta\delta$ 较小时, 算法易陷入局部最优出现较差解, 但 $\Delta\delta$ 较大也会因此降低搜索精度. 通常取 0.015 左右时优化结果较理想.

3.3.2 动态逼近因子 α 的影响

保持其他参数不变, 动态逼近因子分别取 0.1, 0.3, 0.5 和自适应变化. 函数设置同上, 优化结果的最优值、最差值和平均值如表 9 所示. 可以看出, 当 α 较小时, 动态步长衰减较快, 搜索精度高, 但不能保证算法跳出局部最优. 对于单模函数, 当 $\alpha = 0.1$ 时可以保证较高的收敛精度, 但对于多模函数却容易因此陷入早熟收敛. 当 $\alpha = 0.5$ 时由于步长变幅较小, 局部搜索步长较大, 算法易跳过最优解区域. 本文采用自适应变化方式, 算法初期保持较大的全局搜索步长, 在算法后期自适应减小步长, 增强局部搜索精度. 实例验证了该方法相比单一的系数法具有更高的全局寻优能力和收敛精度.

表 8 $\Delta\delta$ 对测试函数结果影响

函数	$\Delta\delta = 0.005$	$\Delta\delta = 0.01$	$\Delta\delta = 0.015$	$\Delta\delta = 0.02$
	[最优解, 最差解, 平均解]	[最优解, 最差解, 平均解]	[最优解, 最差解, 平均解]	[最优解, 最差解, 平均解]
f_2	[1.102e-51, 2.812e-41, 2.816e-42]	[1.127e-50, 1.077e-39, 1.579e-40]	[1.723e-47, 1.411e-37, 1.473e-38]	[3.253e-47, 6.159e-37, 4.359e-38]
f_3	[0.017, 1.071, 0.363]	[8.173 e-03, 0.883, 0.312]	[6.319e-03, 0.419, 0.167]	[0.024, 1.012, 0.252]
f_4	[0, 0.994, 0.198]	[0, 0.994, 0.099]	[0, 1.734 e-18, 4.336 e-19]	[0, 2.602e-18, 1.041e-18]

表 9 α 对测试函数结果影响

函数	$\alpha = 0.1$	$\alpha = 0.3$	$\alpha = 0.5$	α 自适应变化
	[最优解, 最差解, 平均解]	[最优解, 最差解, 平均解]	[最优解, 最差解, 平均解]	[最优解, 最差解, 平均解]
f_2	[4.265e-217, 8.266e-59, 1.796e-59]	[1.348e-50, 1.138e-40, 1.099e-41]	[1.705e-35, 5.860e-46, 1.721e-36]	[8.853e-52, 5.593e-47, 1.204e-47]
f_3	[0.021, 10.49, 1.790]	[0.002, 2.042, 0.456]	[1.363, 7.761, 4.297]	[7.065e-4, 0.699, 0.271]
f_4	[0, 0, 0]	[0, 0, 0]	[0, 8.673e-19, 2.602e-19]	[0, 0, 0]

表 10 精英个数 m 对测试函数结果影响

函数	$m = 1$		$m = 0.05x$		$m = 0.1x$	
	[最优解, 最差解, 平均解]	运行时间	[最优解, 最差解, 平均解]	运行时间	[最优解, 最差解, 平均解]	运行时间
f_2	[7.891e-47, 2.206e-42, 1.82e-44]	1.931	[4.990e-48, 5.353e-44, 9.549e-45]	2.828	[6.634e-51, 4.612e-46, 3.485e-48]	4.066
f_3	[2.795e-5, 0.739, 0.140]	3.797	[9.506e-7, 1.481e-4, 5.681e-5]	5.704	[1.831e-7, 6.275e-6, 1.728e-6]	9.813
f_4	[0, 8.673e-19, 8.673e-20]	2.807	[0, 0, 0]	4.375	[0, 0, 0]	8.086

3.3.3 精英个体数量的影响

针对不同的精英数保持其他控制参数不变, 考虑到运行时间的影响, 本文讨论了 3 种不同精英数量下函数的运行结果, 如表 10 所示. 可以看出, 增加精英学习个体数可以改善个体自身最优位置坐标, 但同时平均运行时间也大幅度增加. 对于多峰复杂问题的求解可通过增加精英个体数来增强个体的全局寻优能力, 提高算法收敛精度; 而对于求解过程较简单的函数而言, 已有的学习机制已经能够达到要求. 因此, 实际工程问题可根据具体情况调整精英个体数量获取最优解.

4 结 论

基于对QPSO算法易陷入局部最优的原因分析, 考虑到算法中局部吸引子的指导作用, 论文通过精英个体选择策略对不同进化状态下的种群引入两种精英学习策略, 提出了基于精英学习的量子行为粒子群(QPSO-EL), 并分析了动态逼近因子、多样性度量阈值和精英个体数量对算法性能的影响. 在该算法中, 动态逼近局部更新策略提高了算法的局部搜索能力, 改善了个体概率进化中心点位置; 变尺度混沌扰动的引入, 拓宽了种群聚集时个体的寻优空间, 提高了算法全局寻优几率. 通过多个测试函数的仿真结果对比分析, QPSO-EL较其他算法具有更高的全局寻优能力和收敛精度, 适用于对多峰函数的优化求解.

参考文献(References)

[1] Kennedy J, Eberhart R. Particle swarm optimization[C]. Proc of IEEE Int Conf on Neural Networks. Perth: IEEE,

1995: 1942-1948.

[2] Van den Bergh F. A new locally convergent particle swarm optimizer[C]. Proc of the IEEE Int Conf on Systems, Man and Cybernetics. Tunisia: IEEE, 2002: 94-99.

[3] Sun J, Feng B, Xu W B. Particle swarm optimization with particles having quantum behavior[C]. Proc of 2004 Congress on Evolutionary Computation. Portland: IEEE, 2004: 325-331.

[4] 周頔, 孙俊, 须文波. 具有量子行为的协同粒子群优化算法[J]. 控制与决策, 2011, 26(04): 582-586.
(Zhou D, Sun J, Xu W B. Quantum-behaved particle swarm optimization algorithm with cooperative approach[J]. Control and Decision, 2011, 26(4): 582-586.)

[5] 黄泽霞, 俞攸红, 黄德才. 惯性权自适应调整的量子粒子群优化算法[J]. 上海交通大学学报, 2012, 46(2): 228-232.
(Huang Z X, Yu Y H, Huang D C. Quantum-behaved particle swarm algorithm with self adapting adjustment of inertia weight[J]. J of Shanghai Jiaotong University, 2012, 46(2): 228-232.)

[6] 方伟, 孙俊, 谢振平, 等. 量子粒子群优化算法的收敛性分析及控制参数研究[J]. 物理学报, 2010, 59(6): 3686-3694.
(Fang W, Sun J, Xie Z P, et al. Convergence analysis of quantum-behaved particle swarm optimization algorithm and study on its control parameter[J]. Acta Physica Sinica, 2010, 59(6): 3686-3694.)

[7] Coelho L S. Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering

- design problems[J]. *Expert Systems with Applications*, 2010, 37(2): 1676-1683.
- [8] Sun J, Lai C H, Xu W B, et al. A modified quantum-behaved particle swarm optimization [C]. *Proc of IEEE Conf on Computational Science*. Beijing: IEEE, 2007: 294-301.
- [9] 龙海侠, 马生全. 基于多样性变异的量子行为粒子群优化算法[J]. *计算机应用研究*, 2011, 28(6): 2064-2066.
(Long H X, Ma S Q. Quantum-behaved particle swarm optimization with diversity guided mutation[J]. *Application Research of Computers*, 2011, 28(6): 2064-2066.)
- [10] 林星, 冯斌, 孙俊. 混沌量子粒子群优化算法[J]. *计算机工程与设计*, 2008, 29(10): 2610-2612.
(Lin X, Feng B, Sun J. Chaos quantum-behaved particle swarm optimization algorithm[J]. *Computer Engineering and Design*, 2008, 29(10): 2610-2612.)
- [11] Liu J, Sun J, Xu W B. Improving quantum-behaved particle swarm optimization by simulated annealing[J]. *Computational Intelligence and Bioinformatics*, 2006, 4115: 130-136.
- [12] Chen D B, Wang J T. An improved group search optimizer with operation of quantum-behaved swarm and its application[J]. *Applied Soft Computing*, 2012, 12(2): 712-725.
- [13] 朱颢东, 李红婵. 新的自适应免疫量子粒子群优化算法[J]. *微电子学与计算机*, 2011, 28(3): 123-125.
(Zhu H D, Li H C. New adaptive immune quantum-behaved particle swarm optimization algorithm[J]. *Micro electronics and Computer*, 2011, 28(03): 123-125.)
- [14] 龙海侠, 须文波, 王小根, 等. 基于选择操作的量子粒子群算法[J]. *控制与决策*, 2010, 25(10): 1499-1506.
(Long H X, Xu W B, Wang X G, et al. Using selection to improve quantum-behaved particle swarm optimization[J]. *Control and Decision*, 2010, 25(10): 1499-1506.)
- [15] 田雨波, 彭涛, 沙莎. 基于微分进化算子和混沌扰动的量子粒子群算法[J]. *江苏科技大学学报: 自然科学版*, 2011, 25(2): 158-162.
(Tian Y B, Peng T, Sha S. Improved quantum-behaved particle swarm optimization algorithm based on differential evolution operator and chaos disturbance[J]. *J of Jiangsu University of Science and Technology: Natural Science Edition*, 2011, 25(2): 158-162.)
- [16] Sun J, Fang W, Palade V, et al. Quantum-behaved particle swarm optimization with Gaussian distributed local attractor point[J]. *Applied Mathematics and Computation*, 2011, 218(7): 3763-3775.
- [17] 李盼池, 王海英, 宋考平, 等. 量子势阱粒子群优化算法的改进研究[J]. *物理学报*, 2012, 61(6): 1-10.
(Li P C, Wang H Y, Song K P, et al. Research on the improvement of quantum potential well-based particle swarm optimization algorithm[J]. *Acta Physica Sinica*, 2012, 61(6): 1-10.)
- [18] Clerc M, Kennedy J. The particle swarm: Explosion, stability, and convergence in a multi-dimensional complex space[J]. *IEEE Trans on Evolutionary Computation*, 2002, 6(1): 58-73.
- [19] 刘军民, 高岳林. 混沌粒子群优化算法[J]. *计算机应用*, 2008, 28(2): 322-325.
(Liu J M, Gao Y L. Chaos particle swarm optimization algorithm[J]. *Computer Applications*, 2008, 28(2): 322-325.)
- [20] 逢珊, 杨欣毅, 张小峰. 混沌映射的多种群量子粒子群优化算法[J]. *计算机工程与应用*, 2012, 48(33): 56-62.
(Pang S, Yang X Y, Zhang X F. Chaotic mapping multi population quantum-behaved particle swarm optimization algorithm[J]. *Computer Engineering and Applications*, 2012, 48(33): 56-62.)

(上接第1334页)

- [17] Lofberg J. Yalmip: A toolbox for modeling and optimization in Matlab[C]. *Proc of 2004 IEEE Int Symposium on Computer Aided Control Systems Design*. Taipei, 2004: 284-289.
- [18] Parrilo P A, Sturmfels B. Minimizing polynomial functions[C]. *Workshop on Algorithmic and Quantitative Aspects of Real Algebraic Geometry in Mathematics and Computer Science 2001*. Providence : American Mathematical Society, 2003: 83-100.
- [19] Reznick B. Some concrete aspects of Hilbert's 17th problem[C]. *Contemporary Mathematics*. Providence : American Mathematical Society, 2000: 251-272.
- [20] 俞立. 鲁棒控制——线性矩阵不等式处理方法[M]. 北京: 清华大学出版社, 2002: 21-22.
(Yu L. Robust control—Linear matrix inequality approach[M]. Beijing: Tsinghua University Press, 2002: 21-22.)