

量子粒子群优化算法的收敛性分析及控制参数研究^{*}

方 伟[†] 孙 俊 谢振平 须文波

(江南大学信息工程学院, 无锡 214122)

(2009 年 7 月 5 日收到; 2009 年 12 月 17 日收到修改稿)

通过分析粒子群优化算法的特点, 将粒子放在量子空间来描述, 建立粒子的量子势能场模型, 并结合群体的群集性推导了量子粒子群优化 (QPSO) 算法. 在随机算法全局收敛定理的框架下, 讨论了 QPSO 算法的收敛性, 证明 QPSO 算法是一种全局收敛的算法. 针对 QPSO 算法的唯一控制参数, 提出了三种控制策略, 结合标准测试函数的仿真结果给出了具有实际指导意义的控制参数选择方法.

关键词: 量子系统, 势阱, 全局收敛, 粒子群优化

PACC: 0365

1. 引 言

粒子群优化 (PSO) 算法是一种新的群体智能优化算法^[1, 2], 其思想来源于对鸟类群集行为的模拟. PSO 算法自提出以来, 由于算法本身对目标函数的性质没有依赖以及算法的易于实现、参数相对较少等特点和良好的优化性能引起了国内外众多学者的关注和研究^[3-10], 已经成为当前群体智能优化方法研究的一个热点. 对 PSO 算法的性能改进是研究的一个重要方向, 这是由于 PSO 算法存在着以下的主要缺陷: (1) 当迭代次数趋于无穷大时, 算法不能以概率 1 收敛于全局最优解^[11]; (2) PSO 算法是模拟鸟类的群体行为而设计的, 其进化方程是一组简单的速度和位置 (代表决策变量) 状态方程, 这种模型的随机性和群集智能性比较低; (3) PSO 算法的收敛性在很大程度上依赖于粒子群速度的上限值或参数的选取^[12], 而上限值的设置与粒子的初始位置 and 最优解的距离有关, 控制参数的选取往往需要丰富的经验, PSO 算法的鲁棒性被降低. 为了提高 PSO 算法中粒子的随机性和 PSO 算法的全局搜索性能, 文献^[13]提出将每个个体都用量子空间中的一个粒子来描述, 这是因为人类的学习过程与粒子的量子行为极为相似, 即具有很大的不确定性, 并根据群体智慧的聚集性建立量子势能场模型, 然后

根据人类群体自组织性和协同性等特点, 开发了新的群体智能算法——量子粒子群优化 (QPSO) 算法. QPSO 算法也以其易于实现、参数较少等特点在多个应用领域以及算法改进上得到了一定的关注^[14, 15]. 尽管 QPSO 算法在应用中体现了较好的全局优化性能, 但是对算法本身的全局收敛性还没有做出过理论研究.

本文将在随机算法全局收敛的框架下对 QPSO 算法的全局收敛性做出证明, 还将对 QPSO 算法控制参数的取值策略加以研究, 以寻找更为有效的控制参数使用方法.

2. QPSO 算法思想

2.1. 从 PSO 算法到 QPSO 算法

在文献^[16]中, Clerc 和 Kennedy 对 PSO 算法中粒子的运行轨道作了分析, 指出在 PSO 算法中如果每个粒子都能够收敛到它的局部吸引点 $p_i = (p_{i1}, p_{i2}, \dots, p_{id}, \dots, p_{iD})$, 那么 PSO 算法可能收敛. 这里

$$p_{id}(t) = \frac{c_1 r_{1d}(t) P_{id}(t) + c_2 r_{2d}(t) P_{gd}(t)}{c_1 r_{1d}(t) + c_2 r_{2d}(t)} \quad (1 \leq d \leq D, 1 \leq i \leq N), \quad (1)$$

或者

^{*} 国家自然科学基金 (批准号: 60474030) 和江南大学科研基金 (批准号: 1055210322090270, 1055211542080210) 资助的课题.

[†] E-mail: wxfangwei@hotmail.com

$$p_{id}(t) = \varphi_d(t)P_{id}(t) + [1 - \varphi_d(t)]P_{gd}(t), \quad (2)$$

$$\varphi_d(t) = c_1 r_{1d}(t) / [c_1(t)r_{1d}(t) + c_2 r_{2d}(t)], \quad (3)$$

式中 t 是算法的当前迭代次数, D 是粒子的维数, N 是群体的规模, $r_{1d}(t)$ 和 $r_{2d}(t)$ 是 $(0, 1)$ 间的随机数, P_i 为粒子 i 的当前最优位置, P_g 为群体的全局最优位置. 在 PSO 算法中, 学习因子 c_1 和 c_2 通常设置为相等的值, 即 $c_1 = c_2$. 因此, (3) 式可以重新写为

$$\varphi_d(t) = r_{1d}(t) / [r_{1d}(t) + r_{2d}(t)]. \quad (4)$$

由此可知, $\varphi_d(t)$ 是均匀分布在 $(0, 1)$ 上随机数, 将 (2) 式改写为

$$p_{id}(t) = \varphi_d(t)P_{id}(t) + [1 - \varphi_d(t)]P_{gd}(t). \quad (5)$$

从 (5) 式可以看出, p_i 的位置位于由点 P_i 和 P_g 为顶点构成的超矩形中, 随着点 P_i 和 P_g 的变化而变化. 当粒子收敛到它们的局部吸引点的时候, 粒子的自身最优位置、局部吸引点和全局最优位置将会收敛到同一点, 从而使得 PSO 算法收敛.

从动力学的角度看, PSO 算法中粒子的收敛过程是以局部吸引点 p 为吸引子, 随着速度的减小不断地接近 p 点, 最后跌落到 p 点. 因此在整个过程中, 在 p 点处实际上存在某种形式的吸引势能场吸引着粒子. 这正是整个粒子群能保持聚集性的原因. 由于在经典的 PSO 系统中粒子的收敛是以轨道的形式实现的, 并且由于粒子的速度总是有限的, 因此在搜索过程中粒子的搜索空间是一个有限的区域, 不能覆盖整个可行空间, 这是 PSO 算法本身所存在的局限性, 它已经被 Van den Bergh^[11] 证明不是一个全局收敛算法. 因此, 我们提出将每个个体都用量子空间中的一个粒子来描述, 以体现粒子的不确定性, 并根据群体智慧的聚集性建立量子势能场模型, 然后根据人类群体自组织性和协同性等特点提出了 QPSO 算法^[13].

2.2. QPSO 算法

假设 PSO 系统是一个量子系统, 在量子空间中粒子的速度和位置不能同时确定, 每个粒子的状态都由波函数 ψ 来确定, $|\psi|^2$ 是粒子位置的概率密度函数. 由文献^[16]对 PSO 系统中粒子的收敛分析可知, 假定在第 t 次迭代, 粒子 i 在 D 维空间运动, 该粒子在第 d 维的势阱为 $p_{id}(t)$, 则在第 $(t+1)$ 次迭代可以得到粒子 i 的波函数为

$$\psi[x_{id}(t+1)] = \frac{1}{\sqrt{L_{id}(t)}} \exp \left[-\frac{|x_{id}(t+1) - p_{id}(t)|}{L_{id}(t)} \right]. \quad (6)$$

粒子的概率密度函数 Q 表示为

$$\begin{aligned} Q[x_{id}(t+1)] &= |\psi[x_{id}(t+1)]|^2 \\ &= \frac{1}{L_{id}(t)} \exp \left[\frac{-2|x_{id}(t+1) - p_{id}(t)|}{L_{id}(t)} \right]. \end{aligned} \quad (7)$$

粒子的概率分布函数 T 表示为

$$T[x_{id}(t+1)] = \exp \left[\frac{-2|x_{id}(t+1) - p_{id}(t)|}{L_{id}(t)} \right]. \quad (8)$$

在 (6)–(8) 式中, $L_{id}(t)$ 是双指数分布的标准偏差. 应用 Monte Carlo 方法可以得到在第 $(t+1)$ 次迭代时, 第 i 个粒子第 d 维的位置为

$$x_{id}(t+1) = p_{id}(t) \pm \frac{L_{id}(t)}{2} \ln[1/u_{id}(t)], \quad (9)$$

式中 $u_{id}(t)$ 是均匀分布在 $(0, 1)$ 上的随机数. $L_{id}(t)$ 的值由下式确定:

$$L_{id}(t) = 2\alpha(t) |C_d(t) - x_{id}(t)|, \quad (10)$$

式中参数 C 称为平均最优位置, 它是所有粒子自身最优位置的中心点, 可由下式计算得到:

$$\begin{aligned} C(t) &= (C_1(t), C_2(t), \dots, C_D(t)) \\ &= \frac{1}{N} \sum_{i=1}^N P_i(t) \\ &= \left(\frac{1}{N} \sum_{i=1}^N P_{i1}(t), \frac{1}{N} \sum_{i=1}^N P_{i2}(t), \dots, \frac{1}{N} \sum_{i=1}^N P_{iD}(t) \right). \end{aligned} \quad (11)$$

综合 (9), (10) 式可以得到粒子的位置更新方程为

$$\begin{aligned} x_{id}(t+1) &= p_{id}(t) \pm \alpha(t) |C_{id}(t) - x_{id}(t)| \\ &\quad \times \ln[1/u_{id}(t)], \end{aligned} \quad (12)$$

式中参数 α 称为压缩-扩张因子.

粒子的当前最优位置 P_i 与全局最优位置 P_g 的更新方式与 PSO 算法中相应参数的更新方式完全相同, 即

$$P_i(t+1) = \begin{cases} x_i(t+1) & (f[x_i(t+1)] < f[P_i(t)]), \\ P_i(t) & (f[x_i(t+1)] \geq f[P_i(t)]), \end{cases} \quad (13)$$

$$P_g(t+1) = \arg \min_{1 \leq i \leq N} \{f[P_i(t)]\}. \quad (14)$$

式中 $f(\cdot)$ 为目标函数. 将 (12) 式作为粒子位置更新公式的 PSO 算法称为 QPSO 算法^[13]. 下面给出 QPSO 算法的执行步骤.

步骤 1 在问题空间中初始化粒子群中粒子的位置.

步骤 2 根据 (11) 式计算粒子群的平均最优

位置.

步骤 3 计算粒子的当前适应值并与前一次迭代的适应值比较,如果当前适应值小于前一次迭代的适应值,则根据粒子的位置更新为粒子当前的最优位置,即如果 $f[x_i(t+1)] < f[P_i(t)]$, 则 $P_i(t+1) = x_i(t+1)$.

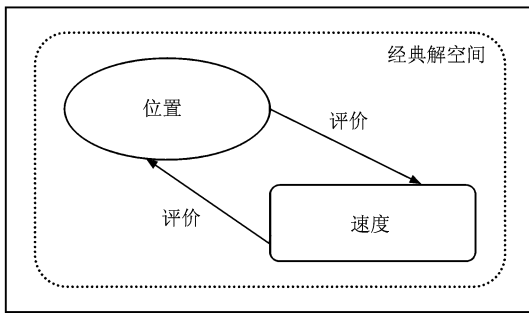
步骤 4 计算群体当前的全局最优位置,即 $P_g(t+1) = \arg \min_{1 \leq i \leq N} \{f[P_i(t)]\}$.

步骤 5 比较当前全局最优位置与前一次迭代的全局最优位置,如果当前全局最优位置较好,则群体的全局最优位置更新为它的值.

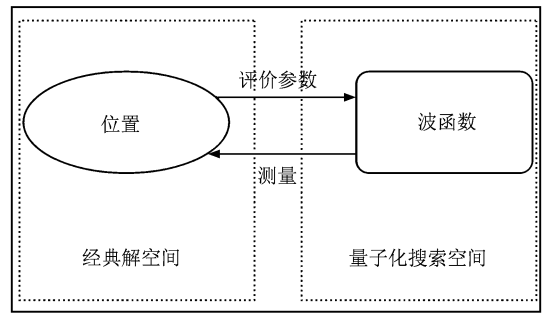
步骤 6 分别根据(5),(12)式计算随机点的位置和粒子的新位置.

步骤 7 重复步骤 2—步骤 6,直至满足一定的结束条件.

QPSO 算法是在 PSO 算法的基础上结合量子理论提出的,它不仅具有 PSO 算法的优点,同时在五个方面做出了进一步的改进. PSO 算法采用速度-位移模型,而 QPSO 算法采用了仅有位移的模型. PSO 算法与 QPSO 算法在状态转换上的差异如图 1 所示. 对于 QPSO 算法,量子系统的粒子在测量之前没有既定的规程,而以一定的概率分布出现在任何位置,从而可以达到全局搜索. PSO 算法的控制参数比遗传算法少, QPSO 算法的控制参数较 PSO 算法更少. 对相同的标准测试函数进行测试, QPSO 算法比 PSO 算法表现出较好的求解性能^[13,17].



(a)



(b)

图 1 PSO 算法和 QPSO 算法的状态转换 (a) PSO 算法, (b) QPSO 算法

3. QPSO 算法的收敛性分析

下面将证明 QPSO 算法是一个全局搜索算法. 在证明之前,首先给出判别算法收敛的定义^[18].

定义 1 局部搜索算法是一种只能保证搜索到目标函数的局部最优解的算法.

定义 2 全局搜索算法是一种能够保证搜索到目标函数的全局最优解的算法.

在分析 QPSO 算法收敛性时还需要使用到一些其他的假设和定理,因此为了分析的方便,简要给出了它们的内容.

3.1. 全局搜索算法的收敛准则

定义 3 给定一个目标函数 f , 它的解空间是从 \mathbb{R}^n 到 \mathbb{R} , S 是 \mathbb{R}^n 的一个子集. 在 S 中寻找一个点 z , 它能够使得函数 f 的值最小化或者至少能够生成一个函数 f 在 S 上的可接受的下确界.

假设 1 $f(H(z, \xi)) \leq f(z)$, 如果 $\xi \in S$, 则 $f(H(z, \xi)) \leq f(\xi)$.

假设 1 中的 H 是指可以在待求解问题的空间产生一个解的函数, 应能够保证 H 所产生的新个体优于当前个体. 不同的 H 就代表了不同的具体算法, 但是都必须满足假设 1 以保证优化算法的正确运行. z 是存在于解的子集 S 中的某个最小值, ξ 是根据相应算法在子集 S 中所得的一系列的可解.

任何算法的全局收敛意味着序列 $\{f(z_t)\}_{t=1}^{\infty}$ 收敛于函数 f 在 S 上的下确界. 为了避免一些病态的情况^[18], 将搜索目标变为搜索本质下确界 ϕ ,

$$\phi = \inf(m : v[z \in S | f(z) < m] > 0), \quad (15)$$

式中 $v[A]$ 是在集合 A 上的 Lebesgue 测度.

假设 2 对于 S 的任意 Borel 子集 A , 若其测度 $v[A] > 0$, 则有

$$\prod_{i=0}^{\infty} (1 - \mu_i[A]) = 0, \quad (16)$$

式中 $\mu_i[A]$ 是由测度 μ_i 所得到 A 的概率.

假设 2 是指对于位置测度为 v 的任意一个 A 的子集, 如果采用随机取样的方法, 那么它重复错过集合 A 的概率必定为零. 由于算法的 ε 可接受区域

$R_\varepsilon \subset S(R_\varepsilon = \{z \in S \mid f(z) < \phi + \varepsilon\}, \varepsilon > 0)$, 所有在可接受区域取得点的概率肯定是非零值。

定理 1(全局收敛的充要条件)^[18] 假设目标函数 f 为可测函数, 区域 S 是 \mathfrak{R}^n 的可测子集, 能够满足假设 1 和假设 2, 设 $\{z_k\}_{k=0}^\infty$ 为算法生成的解序列, 可得

$$\lim_{t \rightarrow +\infty} B[z_t \in R_\varepsilon] = 1, \quad (17)$$

式中 $B[z_t \in R_\varepsilon]$ 是指在第 t 步由算法生成的解 $z_t \in R_\varepsilon$ 的概率。

3.2. QPSO 算法的全局收敛性证明

下面将利用定理 1 对 QPSO 算法的全局收敛性进行证明, 也就是证明 QPSO 算法能同时满足假设 1 和假设 2。

引理 1 QPSO 算法满足假设 1。

证明 从(13), (14)式可以得出, 函数 H 在 QPSO 算法中的描述可以定义为

$$H(P_{g,t}, x_{i,t}) = \begin{cases} P_{g,t} & (f(F_{\text{app}}(x_{i,t})) \geq f(P_{g,t})), \\ F_{\text{app}}(x_{i,t}) & (f(F_{\text{app}}(x_{i,t})) < f(P_{g,t})), \end{cases} \quad (18)$$

式中 $F_{\text{app}}(x_{i,t})$ 表示具体应用的函数, 通过执行 QPSO 算法来更新. 将 $x_{i,t+1}$ 作为成功应用函数 F_{app} 得到的计算结果, 即

$$x_{i,t+1} = F_{\text{app}}(x_{i,t}). \quad (19)$$

用 $F_{\text{app}}(x_{i,t})_d$ 表示 F_{app} 的第 d 维, 具体表达方式如下:

$$F_{\text{app}}(x_{i,t})_d = \frac{c_1 r_{1d}(t) P_{i,d,t} + c_2 r_{2d}(t) P_{g,d,t}}{c_1 r_{1d}(t) + c_2 r_{2d}(t)} + \alpha_t |C_{d,t} - x_{i,d,t}| \ln\left(\frac{1}{u_{i,d,t}}\right), \quad (20)$$

式中 $C_{d,t}$ 是在第 t 次迭代第 d 维的平均最优位置。

以上所述表明粒子位置对迭代次数 t 的依赖关系. 序列 $\{P_{g,t}\}_{t=0}^\infty$ 是所有粒子从开始到第 t 次迭代步(包括第 t 次迭代)所到达最优位置序列。

按照 QPSO 算法的定义, 序列 $P_{g,t}$ 是单调的, 因此 H 的定义是显然符合假设 1. 证毕。

引理 2 QPSO 算法满足假设 2。

证明 在任意一个迭代步 t , 粒子 i 的第 d 维的概率密度函数为

$$Q(x_{i,d,t}) = \frac{1}{L_{i,d,t}} \exp\left(\frac{-2|x_{i,d,t} - p_{i,d,t}|}{L_{i,d,t}}\right), \quad (21)$$

粒子 i 的概率密度函数可以表示为

$$Q(x_{i,t}) = \prod_{d=1}^D \frac{1}{L_{i,d,t}} \exp\left(\frac{-2|x_{i,d,t} - p_{i,d,t}|}{L_{i,d,t}}\right). \quad (22)$$

定义 $\mu_{i,t}$ 为对应于 D 维双指数分布的概率测度, 对于 S 的任意 Borel 子集 A , 满足 $v[A] > 0$, 可以得到

$$\mu_{i,t}[A] = \int_A \left[\prod_{d=1}^D \frac{1}{L_{i,d,t}} \exp\left(\frac{-2|x_{i,d,t} - p_{i,d,t}|}{L_{i,d,t}}\right) \right] \times dx_{i,1,t} dx_{i,2,t} \cdots dx_{i,D,t}, \quad (23)$$

$$M_{i,t} = \mathfrak{R}^D \supset S, \quad (24)$$

式中 $M_{i,t}$ 是 $\mu_{i,t}$ 在样本空间的支撑, 并且 $A \supset M_{i,t}$. 因此, 可以得到

$$0 < \mu_{i,t}[A] < 1. \quad (25)$$

粒子的支撑联合为

$$M_t = \bigcup_{i=1}^N M_{i,t} = \mathfrak{R}^D \supset S, \quad (26)$$

式中 M_t 是分布 μ_t 的支撑. 由 μ_t 生成的概率 A 可以由下式计算得到:

$$\mu_t[A] = 1 - \prod_{i=1}^N (1 - \mu_{i,t}[A]). \quad (27)$$

通过(25)式可得

$$0 < \mu_t[A] < 1. \quad (28)$$

因此

$$\prod_{t=0}^{\infty} (1 - \mu_t[A]) = 0. \quad (29)$$

这就表明 QPSO 算法满足假设 2. 证毕。

引理 3 QPSO 算法是一个全局收敛算法。

证明 由于 QPSO 算法满足假设 1 和假设 2, 通过定理 1 可以得出 QPSO 算法是一个全局收敛算法. 证毕。

4. QPSO 算法的控制参数研究

算法参数是影响算法性能和效率的关键, 如何确定最优参数使算法性能最佳是一个极其复杂的优化问题^[19]. 在 QPSO 算法中, 除了群体大小、问题维数与迭代次数外的唯一控制参数为(12)式中的压缩-扩张因子 α , 它在算法中起到的作用是能够调节算法的收敛过程. 尽管算法只有一个控制参数, 但是尚未对其进行过系统的研究. 在文献[20]中, 通过仿真结果得知 QPSO 算法的压缩-扩张因子必须满足 $\alpha < 1.781$ 才能保证粒子的收敛, 否则算法

将肯定不能保证收敛. 在这样的前提条件下, 如何选择 α 的取值及取值方式已成为一个重要的研究课题. 本文研究了 α 在以下三种取值策略下的 QPSO 算法的优化能力: 第一种是采用固定取值策略, 第二种是采用动态线性递减策略, 第三种是动态非线性递减策略. 通过算法对多个不同特征的标准测试函数的求解来说明 QPSO 算法在 α 采用不同取值策略下的性能. 这三种取值策略的共同特点在于不需要进行复杂的数学运算和方法来决定参数的值, 这样可以在不增加算法复杂度的情况下研究 α 的控制

方法.

4.1. 实验设置

实验中采用 Sphere, Rosenbrock, Rastrigin, Griewank 和 Ackley 5 个标准测试函数来测试不同取值策略下 QPSO 算法的性能. 5 个测试函数的数学表达式及自变量取值范围列于表 1. 5 个测试函数的维数都分别考虑了 $D = 10, 20, 30$ 的情况, 对应维数的最大迭代次数 $t_{\max} = 1000, 1500, 2000$, 对应维数的粒子数 $N = 20, 40, 80$.

表 1 标准测试函数

函数名称	函数表达式	自变量取值范围
Sphere	$f_1(x) = \sum_{d=1}^D x_d^2$	$-100 \leq x_d \leq 100$
Rosenbrock	$f_2(x) = \sum_{d=1}^D [100(x_{d+1} - x_d^2) + (x_d - 1)^2]$	$-30 \leq x_d \leq 30$
Rastrigin	$f_3(x) = \sum_{d=1}^D [x_d^2 - 10\cos(2\pi x_d) + 10]$	$-5.12 \leq x_d \leq 5.12$
Griewank	$f_4(x) = \frac{1}{4000} \sum_{d=1}^D x_d^2 - \prod_{d=1}^D \cos\left(\frac{x_d}{\sqrt{d}}\right) + 1$	$-600 \leq x_d \leq 600$
Ackley	$f_5(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{d=1}^D x_d^2}\right) - \exp\left(\frac{1}{D} \sum_{d=1}^D \cos 2\pi x_d\right) + 20 + e$	$-32 \leq x_d \leq 32$

4.2. 控制参数的固定取值策略

设置 QPSO 算法的控制参数 α 的值分别为 1.2, 1.0, 0.8, 0.6, 0.4, 0.2 来求解测试函数.

表 2 列出了在控制参数 α 取不同固定值的情况下应用 QPSO 算法求解 5 个标准测试函数的结果, 表中列出的计算值都是经过 50 次运算后得到的平均最优值. 从计算结果可以看出, 在 $\alpha = 0.8$ 的情况下, QPSO 算法在 80% 的测试中取得了最好的结果; 其次是 $\alpha = 0.6$ 的情况, 在 13.3% 的测试中取得了最好的结果; $\alpha = 1.0$ 的情况下也有 6% 的测试取得了最好的结果. 从表 2 还可以看出, 对于 Rosenbrock 和 Griewank 函数, 所有对它们的测试都是在控制参数 $\alpha = 0.8$ 时取得了最好的结果. 可以看出 QPSO 算法的控制参数 α 在采用固定取值的策略下, α 的值取为 0.6 或 0.8 时能够对多数的函数取得较好的优化结果.

4.3. 控制参数的线性减小取值策略

该策略是使控制参数 α 的值随迭代次数的变化而线性减小, 即用如下的方程来取值:

$$\alpha(t) = (\alpha_0 - \alpha_1)(t_{\max} - t)/t_{\max} + \alpha_1, \quad (30)$$

式中 $\alpha_1 < \alpha_0$, α_0 和 α_1 分别是控制参数 α 的初始值和终止值. 本文的实验中控制参数 α 设置了 9 组不同的线性减小范围.

表 3—表 7 列出了控制参数 α 采用线性减小取值策略的优化结果. 从统计结果可知, QPSO 算法采用线性减小取值策略能够较平稳地得到优化结果, 与线性减小的范围没有形成很显著的联系, 这是与固定取值策略最大的不同之处. 从表 3—表 7 可以看出: 在控制参数的线性减小范围为 0.8→0.6 的情况下, QPSO 算法在 60% 的测试中取得了最好的结果; 其次是在控制参数的线性减小范围为 1.0→0.5 的情况下, QPSO 算法在 20% 的测试中取得了最好的结果; 其余的几组参数设置也能够在个别的测试中取得较好的结果. Rastrigin 函数是 5 个测试函数中较难求解的多峰函数, 在控制参数的线性减小范围为 1.0→0.5 的情况下所得结果最好. 可以看出, QPSO 算法在控制参数 α 采用线性减小取值策略情况下, 当 α 线性减小的取值范围在 $[0.8, 0.6]$ 和 $[1.0, 0.5]$ 时能够对绝大多数的函数取得较好的优化结果.

表 2 QPSO 算法的控制参数 α 取固定值时求解测试函数的平均最优结果

粒子数	维数	迭代数	$\alpha = 1.2$	$\alpha = 1.0$	$\alpha = 0.8$	$\alpha = 0.6$	$\alpha = 0.5$	$\alpha = 0.4$	$\alpha = 0.2$
Sphere 函数									
20	10	1000	5.09×10^{-2}	1.41×10^{-17}	2.98×10^{-61}	1.69×10^{-3}	1.69×10^1	2.41×10^2	5.70×10^2
40	20	1500	6.78×10^2	1.20×10^{-3}	3.33×10^{-51}	9.76×10^{-11}	16.70×10^{-1}	1.79×10^2	1.35×10^3
80	30	2000	6.55×10^3	59.10×10^{-1}	2.70×10^{-45}	6.55×10^{-72}	1.47×10^{-2}	3.89×10^1	1.65×10^3
Rosenbrock 函数									
20	10	1000	7.31×10^1	84.50×10^{-1}	57.20×10^{-1}	8.66×10^{-1}	3.20×10^3	1.13×10^4	9.47×10^4
40	20	1500	1.18×10^5	6.81×10^1	2.13×10^1	8.22×10^1	1.03×10^3	2.50×10^4	2.93×10^5
80	30	2000	6.65×10^6	2.47×10^3	3.92×10^1	7.29×10^1	4.63×10^2	3.71×10^3	3.31×10^5
Rastrigin 函数									
20	10	1000	3.64×10^1	1.90×10^1	52.80×10^{-1}	1.06×10^1	1.16×10^1	1.43×10^1	1.79×10^1
40	20	1500	1.42×10^2	9.56×10^1	2.11×10^1	2.33×10^1	2.74×10^1	3.30×10^1	5.22×10^1
80	30	2000	2.63×10^2	1.91×10^2	6.18×10^1	3.53×10^1	4.23×10^1	4.73×10^1	8.58×10^1
Griewank 函数									
20	10	1000	7.12×10^{-1}	4.05×10^{-1}	7.23×10^{-2}	2.07×10^{-1}	7.35×10^{-1}	32.9×10^{-1}	88.90×10^{-1}
40	20	1500	66.40×10^{-1}	6.34×10^{-1}	2.35×10^{-2}	5.97×10^{-2}	10.10×10^{-1}	40.9×10^{-1}	1.83×10^1
80	30	2000	5.43×10^1	10.20×10^{-1}	8.12×10^{-3}	2.18×10^{-2}	4.16×10^{-1}	23.3×10^{-1}	2.31×10^1
Ackley 函数									
20	10	1000	1.38×10^{-1}	1.45×10^{-9}	4.62×10^{-2}	15.60×10^{-1}	45.40×10^{-1}	61.20×10^{-1}	93.50×10^{-1}
40	20	1500	81.70×10^{-1}	2.92×10^{-3}	4.80×10^{-15}	25.40×10^{-1}	53.30×10^{-1}	73.90×10^{-1}	97.20×10^{-1}
80	30	2000	1.43×10^1	11.30×10^{-1}	6.22×10^{-15}	22.40×10^{-1}	47.50×10^{-1}	71.20×10^{-1}	93.20×10^{-1}

表 3 QPSO 算法的控制参数 α 值线性减小时求解

Sphere 函数的平均最优结果

α	粒子数, 维数, 迭代数		
	20, 10, 1000	40, 20, 1500	80, 30, 2000
0.8→0.6	1.02×10^{-65}	1.28×10^{-76}	2.71×10^{-94}
1.0→0.6	1.52×10^{-52}	4.07×10^{-49}	6.34×10^{-55}
1.2→0.6	1.86×10^{-39}	1.50×10^{-34}	8.48×10^{-36}
0.8→0.5	2.37×10^{-42}	2.37×10^{-54}	8.01×10^{-72}
1.0→0.5	7.70×10^{-44}	5.58×10^{-43}	2.25×10^{-49}
1.2→0.5	5.38×10^{-34}	2.31×10^{-31}	1.53×10^{-36}
0.8→0.4	9.30×10^{-33}	9.23×10^{-41}	2.60×10^{-55}
1.0→0.4	1.80×10^{-37}	2.65×10^{-36}	7.02×10^{-42}
1.2→0.4	4.87×10^{-29}	2.46×10^{-28}	4.45×10^{-31}

表 4 QPSO 算法的控制参数 α 值线性减小时求解

Rosenbrock 函数的平均最优结果

α	粒子数, 维数, 迭代数		
	20, 10, 1000	40, 20, 1500	80, 30, 2000
0.8→0.6	18.30	1.73×10^1	3.67×10^1
1.0→0.6	5.89	2.19×10^1	3.35×10^1
1.2→0.6	5.18	2.91×10^1	4.06×10^1
0.8→0.5	14.00	2.58×10^1	3.81×10^1
1.0→0.5	7.57	2.95×10^1	3.58×10^1
1.2→0.5	5.72	2.35×10^1	3.55×10^1
0.8→0.4	25.90	3.07×10^1	3.23×10^1
1.0→0.4	7.46	2.84×10^1	2.99×10^1
1.2→0.4	15.00	2.58×10^1	3.09×10^1

表 5 QPSO 算法的控制参数 α 值线性减小时求解

Rastrigin 函数的平均最优结果

α	粒子数, 维数, 迭代数		
	20, 10, 1000	40, 20, 1500	80, 30, 2000
0.8→0.6	5.25	12.20	1.89×10^1
1.0→0.6	6.30	22.00	2.98×10^1
1.2→0.6	5.00	12.30	1.76×10^1
0.8→0.5	3.81	9.47	1.58×10^1
1.0→0.5	4.10	10.60	1.62×10^1
1.2→0.5	4.98	12.50	1.97×10^1
0.8→0.4	3.91	10.50	1.61×10^1
1.0→0.4	4.44	9.97	1.59×10^1
1.2→0.4	5.25	12.20	1.89×10^1

表 6 QPSO 算法的控制参数 α 值线性减小时求解

Griewank 函数的平均最优结果

α	粒子数, 维数, 迭代数		
	20, 10, 1000	40, 20, 1500	80, 30, 2000
0.8→0.6	5.47×10^{-2}	1.93×10^{-2}	6.50×10^{-3}
1.0→0.6	1.08×10^{-1}	2.52×10^{-2}	1.13×10^{-2}
1.2→0.6	1.67×10^{-1}	2.77×10^{-2}	8.20×10^{-3}
0.8→0.5	6.32×10^{-2}	1.76×10^{-2}	1.11×10^{-2}
1.0→0.5	6.10×10^{-2}	1.83×10^{-2}	7.34×10^{-3}
1.2→0.5	1.18×10^{-2}	2.05×10^{-2}	7.87×10^{-3}
0.8→0.4	6.37×10^{-2}	1.63×10^{-2}	9.30×10^{-3}
1.0→0.4	5.81×10^{-2}	2.33×10^{-2}	6.69×10^{-3}
1.2→0.4	6.04×10^{-2}	1.73×10^{-2}	6.79×10^{-3}

表 7 QPSO 算法的控制参数 α 值线性减小时求解 Ackley 函数的平均最优结果

α	粒子数, 维数, 迭代数		
	20, 10, 1000	40, 20, 1500	80, 30, 2000
0.8→0.6	2.95×10^{-15}	6.22×10^{-15}	7.07×10^{-15}
1.0→0.6	3.30×10^{-15}	6.22×10^{-15}	1.05×10^{-14}
1.2→0.6	3.45×10^{-15}	1.21×10^{-14}	12.20×10^{-1}
0.8→0.5	3.29×10^{-2}	6.86×10^{-15}	1.09×10^{-14}
1.0→0.5	3.59×10^{-15}	7.85×10^{-15}	1.26×10^{-14}
1.2→0.5	3.99×10^{-1}	1.65×10^{-14}	8.12×10^{-1}
0.8→0.4	5.29×10^{-15}	1.07×10^{-14}	1.21×10^{-14}
1.0→0.4	3.45×10^{-15}	1.03×10^{-14}	1.55×10^{-14}
1.2→0.4	4.94×10^{-14}	4.01×10^{-1}	4.06×10^{-1}

4.4. 控制参数的非线性减小取值策略

在这里采用了两种方式的非线性减小策略. 第一种方式为开口向下的抛物线, 方程形式如下:

$$\alpha = (\alpha_0 - \alpha_1)(t/t_{\max})^2 + \alpha_0. \quad (31)$$

第二种方式为开口向上的抛物线, 方程形式如下:

$$\alpha = (\alpha_0 - \alpha_1)(t/t_{\max})^2 + (\alpha_1 - \alpha_0)(2t/t_{\max}) + \alpha_0. \quad (32)$$

(31), (32) 式中, $\alpha_0 = 1.0, \alpha_1 = 0.5$.

以最大迭代次数 1000 为例, 在图 2 中给出了包括线性减小取值策略和非线性减小取值策略的控制参数 α 与迭代次数 t 的关系, 三种方式的最大值与最小值相同.

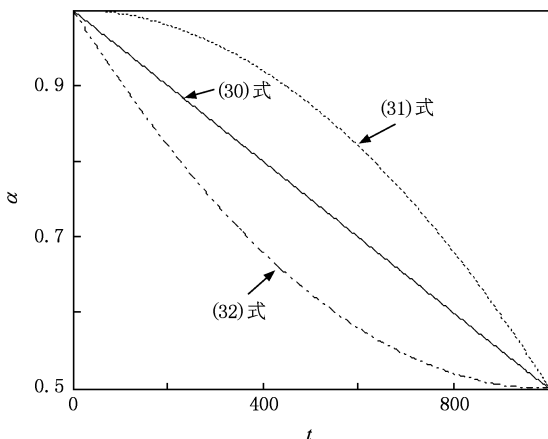


图 2 利用 (30), (31), (32) 式得到的控制参数 α 与迭代次数 t 的关系

表 8 列出了 QPSO 算法的控制参数采用非线性减小策略以及线性减小策略求解 5 个标准测试函数

的优化结果. Sphere 函数的曲面是单峰且线性的, 比较容易求解, 从表 8 的数据可知, 采用线性减小取值策略的 QPSO 算法取得了很好的效果, 因此对于该类函数而言, 控制参数线性减小的取值方法更加有效. Rastrigin 函数在 10 维的情况下, 控制参数采用 (32) 式的取值方式可以得到比另两种方式更好的解. 但是, 当问题的维数增加到 20 以及 30 的时候, 控制参数采用 (30) 式的取值方式则显示出了更强的求解能力. 对于 Ackley 函数而言, 控制参数采用 (30) 式的取值方式对它的求解是比较可靠的. 从表 8 的结果可知, QPSO 算法在控制参数采用 (31) 和 (32) 式的求解中取得较好优化结果的次数基本相同, 但是少于控制参数采用 (30) 式的取值方式. 由此可知, 在本文所研究的参数值中, QPSO 算法采用控制参数线性减小的取值方式会在多数情况下取得较好的结果, 而非线性减小的取值策略也能够取得一些较好的结果.

表 8 QPSO 算法的控制参数 α 按照 (30), (31), (32) 式的取值方式求解测试函数的平均最优结果

粒子数	维数	迭代数	(30)式	(31)式	(32)式
Sphere 函数					
20	10	1000	7.70×10^{-44}	5.96×10^{-40}	8.65×10^{-31}
40	20	1500	5.58×10^{-43}	2.06×10^{-34}	1.06×10^{-34}
80	30	2000	2.25×10^{-49}	8.98×10^{-35}	1.46×10^{-43}
Rosenbrock 函数					
20	10	1000	75.70×10^{-1}	58.20×10^{-1}	1.21×10^1
40	20	1500	2.95×10^1	2.60×10^1	3.18×10^1
80	30	2000	3.58×10^1	3.24×10^1	3.49×10^1
Rastrigin 函数					
20	10	1000	38.10×10^{-1}	47.70×10^{-1}	35.80×10^{-1}
40	20	1500	94.70×10^{-1}	1.30×10^1	1.13×10^1
80	30	2000	1.58×10^1	1.73×10^1	1.65×10^1
Griewank 函数					
20	10	1000	6.10×10^{-2}	1.11×10^{-1}	5.78×10^{-2}
40	20	1500	1.83×10^{-2}	3.13×10^{-2}	1.62×10^{-2}
80	30	2000	7.33×10^{-3}	1.03×10^{-2}	9.90×10^{-3}
Ackley 函数					
20	10	1000	3.59×10^{-15}	3.23×10^{-15}	9.20×10^{-15}
40	20	1500	7.85×10^{-15}	1.18×10^{-14}	1.72×10^{-14}
80	30	2000	1.26×10^{-14}	2.33×10^{-14}	4.07×10^{-1}

4.5. 三种取值策略的比较

为了能够定量分析所提出的三种取值策略的求解鲁棒性, 表 9 列出了文中每种取值策略在求解

表 9 QPSO 算法的控制参数 α 取值策略分别为固定、线性减小和非线性减小时求解测试函数平均最优值的标准差

粒子数	维数	迭代数	固定	线性减小	非线性减小
Sphere 函数					
20	10	1000	2.18×10^2	1.62×10^{-29}	6.11×10^{-31}
40	20	1500	5.20×10^2	8.19×10^{-29}	7.07×10^{-35}
80	30	2000	2.45×10^3	1.48×10^{-31}	6.34×10^{-35}
Rosenbrock 函数					
20	10	1000	3.51×10^4	7.14	4.45
40	20	1500	1.10×10^5	4.29	4.13
80	30	2000	2.50×10^6	3.47	1.77
Rastrigin 函数					
20	10	1000	9.95	0.80	0.84
40	20	1500	45.93	3.81	1.23
80	30	2000	88.27	4.43	0.53
Griewank 函数					
20	10	1000	3.21	3.88×10^{-2}	3.78×10^{-2}
40	20	1500	6.59	3.93×10^{-2}	1.07×10^{-2}
80	30	2000	20.62	1.84×10^{-2}	2.51×10^{-4}
Ackley 函数					
20	10	1000	3.66	1.30×10^{-1}	4.22×10^{-15}
40	20	1500	3.95	1.30×10^{-1}	3.82×10^{-15}
80	30	2000	5.08	4.50×10^{-1}	2.88×10^{-1}

同一个问题时采用不同参数值所得到的平均最优值的标准差. 从表 9 的结果可以看出, 在所有的求解结果中, 采用固定取值策略的标准差总是远大于另外两种取值策略. 因此, 可以认为在本文研究的这些参数值中, 固定取值策略的鲁棒性较差, 而线性减小和非线性减小两种取值策略则体现了求解相对较稳定的特点.

5. 结 论

本文分析了 PSO 算法的特点, 在此基础上结合量子空间的粒子行为提出了 QPSO 算法的思想, 给出了算法的执行步骤, 并对 PSO 算法和 QPSO 算法的特点作了比较. 通过随机算法全局收敛的定理, 证明了 QPSO 算法是一种全局优化算法. 由于控制参数 α 的变化方式会影响到 QPSO 算法的优化性能, 因此本文提出了 α 的三种控制策略. 对这些控制策略应用标准测试函数进行了数值仿真, 并对仿真结果作了分析. 本文工作对于 QPSO 算法的应用具有一定的实际意义.

- [1] Kennedy J, Eberhart R C 1995 *IEEE International Conference on Neural Networks* (Perth: IEEE) p1942
- [2] Poli R, Kennedy J, Blackwell T, Freitas A 2008 *J. Artif. Evol. Appl.* **2008** 1
- [3] Clerc M, Kennedy J 2002 *IEEE Trans. Evolut. Comput.* **6** 58
- [4] Gao H B, Zhou C, Gao L 2005 *Chin. J. Comput.* **28** 1980 (in Chinese) [高海兵、周 驰、高 亮 2005 计算机学报 **28** 1980]
- [5] Gao F, Tong H Q 2006 *Acta Phys. Sin.* **55** 577 (in Chinese) [高 飞、童恒庆 2006 物理学报 **55** 577]
- [6] Wang D F, Han P 2006 *Acta Phys. Sin.* **55** 1644 (in Chinese) [王东风、韩 璞 2006 物理学报 **55** 1644]
- [7] Pan F, Chen J, Gan M G 2006 *Acta Autom. Sin.* **32** 368 (in Chinese) [潘 峰、陈 杰、甘明刚 2006 自动化学报 **32** 368]
- [8] Zeng J C, Cui Z H 2006 *J. Comput. Res. Dev.* **43** 96 (in Chinese) [曾建潮、崔志华 2006 计算机研究与发展 **43** 96]
- [9] Zhao Z J, Xu S Y, Zheng S L, Yang X N 2009 *Acta Phys. Sin.* **58** 5118 (in Chinese) [赵知劲、徐世宇、郑仕链、杨小牛 2009 物理学报 **58** 5118]
- [10] Wang X F, Xue H J, Si S K, Yao Y T 2009 *Acta Phys. Sin.* **58** 3729 (in Chinese) [王校锋、薛红军、司守奎、姚跃亭 2009 物

- 理学报 **58** 3729]
- [11] Van den Bergh F 2001 *Ph. D. Dissertation* (Pretoria: University of Pretoria)
- [12] Eberhart R C, Shi Y 2001 *IEEE Conference on Evolutionary Computation* (Seoul: IEEE) p81
- [13] Sun J, Feng B, Xu W B 2004 *IEEE Congress on Evolutionary Computation* (Hawaii: IEEE) p325
- [14] Dos Santos Coelho L, Alotto P 2008 *IEEE Trans. Magn.* **44** 1074
- [15] Dos Santos Coelho L 2008 *Chaos Solitons Fract.* **37** 1409
- [16] Clerc M, Kennedy J 2002 *IEEE Trans. Evolut. Comput.* **6** 58
- [17] Sun J, Feng B, Xu W B 2004 *IEEE Conference on Cybernetics and Intelligent Systems* (Singapore: IEEE) p111
- [18] Zeng J C, Jie J, Cui Z H 2004 *Particle Swarm Algorithm* (Beijing: Science Press) (in Chinese) [曾建潮、介 婧、崔志华 2004 微粒群算法 (北京: 科学出版社)]
- [19] Wang L 2001 *Intelligent Optimization Algorithms with Applications* (Beijing: Tsinghua University Press) (in Chinese) [王 凌 2001 智能优化算法及其应用 (北京: 清华大学出版社)]
- [20] Sun J, Xu W B, Feng B 2005 *IEEE International Conference on Systems, Man and Cybernetics* (Hawaii: IEEE) p3049

Convergence analysis of quantum-behaved particle swarm optimization algorithm and study on its control parameter*

Fang Wei[†] Sun Jun Xie Zhen-Ping Xu Wen-Bo

(*School of Information Technology, Jiangnan University, Wuxi 214122, China*)

(Received 5 July 2009; revised manuscript received 17 December 2009)

Abstract

Based on the analysis of particle swarm optimization algorithm, the particle is described in the quantum space and the potential energy field model is created. And then according to the swarm's gregariousness, the quantum-behaved particle swarm optimization (QPSO) algorithm is derived. Within the framework of random algorithm's global convergence theorem, the convergence of QPSO algorithm is discussed and is proved to be a kind of global convergence algorithm. Three kinds of control strategy are proposed for the unique parameter of QPSO algorithm and they are tested on five benchmark functions. According to the test results, some conclusions concerning the selection of the parameter are drawn.

Keywords: quantum system, potential well, global convergence, particle swarm optimization

PACC: 0365

* Project supported by the National Natural Science Foundation of China (Grant No. 60474030) and the Scientific Research Foundation of Jiangnan University, China (Grant Nos. 1055210322090270, 1055211542080210).

[†] E-mail: wfangwei@hotmail.com