文章编号: 1001-0920(2008)08-0863-06

一种多样性控制的粒子群优化算法

方 伟, 孙 俊, 须文波
(江南大学 信息工程学院, 江苏 无锡 214122)

摘 要: 针对粒子群优化(PSO)算法的早熟收敛问题,提出一种新的基于群体多样性控制的 PSO 算法(DCPSO). 该方法使得粒子在收缩状态下充分搜索,在发散状态下能够飞离群体的聚集位置,不断的收缩-发散过程保证了群体能在较大的空间进行搜索,减少了粒子群算法的早熟收敛现象. 通过对多个标准测试函数的实验结果表明,DCPSO 算法在复杂优化问题中具有较强的全局搜索能力,而且比现有的多样性指导的 PSO 算法(ARPSO)具有更好的性能.

关键词: 粒子群优化; 早熟收敛; 多样性; 全局收敛
中图分类号: TP18 **文献标识码**: A

Diversity-controlled particle swarm optimization algorithm

FANG Wei, SUN Jun, XU Wen-bo
(School of Information Technology, Jiangnan University, Wuxi 214122, China. Correspondent: FANG Wei, E-mail: wxfangwei@hotmail.com)

Abstract: Aiming at the premature convergence problem in particle swarm optimization (PSO) algorithm, a novel diversity-controlled PSO (DCPSO) algorithm is proposed. Guided by the controlled swarm's diversity, the particles search in the attractive phase sufficiently and adjust themselves by moving away from the center of the swarm quickly in the repulsive phase. The attractive-repulsive procedure can guarantee the population search in a wide space and help to avoid trapping into the local minima. Experimental results on several well-known benchmark functions show that DCPSO has strong global optimization ability in the complicated problems and outperforms the existing diversity-guided PSO (ARPSO).

Key words: Particle swarm optimization; Premature convergence; Diversity; Global convergence

1 引 言

粒子群优化(PSO)[1]算法是一种新的基于群智能的优化方法,其模拟了鸟群觅食过程中的迁徙和聚集行为,通过鸟群之间的集体协作来搜索最优解. 与其他基于群体智能的优化算法,如遗传算法(GA),进化算法(EA)等类似,PSO 算法也是通过初始化一组随机解之后,经过一系列的迭代过程来搜寻最优值. 与 GA 和 SA 相比,PSO 算法具有运算简便、易于实现、控制参数少等特点,已得到国内外众多学者的关注和研究[2-5],并已成功应用于函数优化、布局优化[6,7]等科学和工程领域. PSO 算法存在的一个主要问题是在解决多峰优化的问题中易出现早熟收敛现象,原因在于 PSO 算法是以全体最优解为中心的结构,这使得群体中的信息交流速度快且方向单一,使得粒子快速地聚集在一个较小的搜索区域内,群体多样性因此降低,导致出现早熟收敛现象. 群体的多样性是影响 PSO 算法全局收敛性能的一个重要因素. 在群智能优化算法中,群体的高多样性表明群体能够在较大的空间内进行搜索,这有助于跳出局部最优和减少早熟现象的产生. 但是,若一直保持较高的多样性会减慢算法的收敛速度,甚至导致不收敛. 群体的低多样性表明,群体在一个较小的空间内进行搜索,可以提高解的精度. 因此可以认为,在 PSO 算法中通过合理地控制群体的多样性,并且利用多样性来指导算法的搜索过程可以提高算法的性能. 在文献[8]中,Riget 等将个体位置到群体中心点距离的多样性度量方法以及群体发散行为的思想[9]应用到标准 PSO 中,提出了多样性指导的 PSO(ARPSO)算法.

本文首先从理论上分析了 PSO 算法的收敛和

收稿日期: 2007-06-14; **修回日期**: 2007-09-24.
基金项目: 国家自然科学基金项目(60474030).
作者简介: 方伟(1980—),男,江苏高邮人,博士生,从事智能优化算法的研究; 须文波(1946—),男,江苏无锡人,教授,博士生导师,从事智能算法、自动化控制等研究.

发散条件;然后提出了新的多样性控制的 PSO 算法,给出了控制参数的选择依据. 在标准测试函数中的实验结果表明了所提算法的有效性和可行性.

2 PSO 算法与 ARPSO 算法

2.1 PSO 算法

在 PSO 算法中,每一个粒子都被理解为相应问题的解空间的一个可选解. 在一个 D 维的搜索空间中,每个粒子的速度和位置分别表示为 D 维的矢量,即 $\vec{v}_i = (v_{i1}, \cdots, v_{id}, \cdots, v_{iD})$ 和 $\vec{x}_i = (x_{i1}, \cdots, x_{id}, \cdots, x_{iD})$;粒子位置 x_i 对应的目标函数值为粒子的适应度,用来衡量粒子位置优劣程度;粒子的速度 v_i 决定粒子运动方向和距离;粒子个体自身位置的最优值为 $\vec{p}_i = (p_{i1}, \cdots, p_{id}, \cdots, p_{iD})$,通常也称为 p_{best},粒子群中最好的位置为 $\vec{p}_g$,通常称为 g_{best}. 粒子通过追踪这两个值来更新自己的速度和位置,更新公式如下[1,4]:

$$v_{id}(t+1) = w \cdot v_{id}(t) + c_1 \cdot \text{rand}() \cdot (p_{id}(t) - x_{id}(t)) + c_2 \cdot \text{Rand}() \cdot (p_{gd}(t) - x_{id}(t)), \quad (1)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1). \quad (2)$$

其中:w 是惯性因子;c_1 和 c_2 是学习因子,通常都取值为 2;rand() 和 Rand() 是 (0,1) 之间的随机数;$v_{id}(t)$ 和 $x_{id}(t)$ 是粒子 i 在第 t 次迭代中第 d 维的速度和位置;$p_{id}(t)$ 是粒子 i 在第 t 次迭代中第 d 维的个体极值的位置;$p_{gd}(t)$ 是群体在第 t 次迭代中第 j 维的全局极值的位置.

2.2 ARPSO 算法

Riget 等认为标准 PSO 算法模型中,粒子群在自身最优解和全局最优解的信息指导下,一直处于收缩过程,当找到一个最优解后粒子就基本处于停滞状态,不再更新适应度函数的值. 因此作者提出的 ARPSO 算法,是通过修改 PSO 算法的更新方程 (1),增加了粒子群发散运动方程[8],如下式:

$$v_{id}(t+1) = w \cdot v_{id}(t) - c_1 \cdot \text{rand}() \cdot (p_{id}(t) - x_{id}(t)) - c_2 \cdot \text{Rand}() \cdot (p_{gd}(t) - x_{id}(t)). \quad (3)$$

方程 (3) 是将方程 (1) 的两个学习因子的系数改成了负数形式,粒子群在这个方程的指导下的运动方向是远离当前搜索到的最优解的位置,扩大群体的搜索空间,群体的多样性会相应增加,因此提高了算法搜索到全局最优解的概率. 从文献[8]中可以看出,ARPSO 模型在多峰问题的优化上取得了较好的效果.

3 新的多样性控制的 PSO 算法

为了设计新的多样性控制方法,首先在文献[2]的理论基础上重新分析了 PSO 算法的粒子收敛行为.

3.1 PSO 算法粒子收敛行为分析

考虑标准 PSO 算法的进化方程 (1),令 $\phi_1 = c_1 r_1$,$\phi_2 = c_2 r_2$,则速度更新方程变为

$$v_{id}(t+1) = wv(t) + \phi_1 (p_{id} - x_{id}) + \phi_2 (p_{gd} - x_{id}). \quad (4)$$

定义

$$z_{id} = \frac{\phi_1 p_{id} + \phi_2 p_{gd}}{\phi_1 + \phi_2}, \quad (5)$$

于是速度方程 (4) 可改写为

$$v_{id}(t+1) = wv(t) + \phi (z_{id} - x_{id}), \quad (6)$$

其中 $\phi = \phi_1 + \phi_2$.

不失一般性,现考虑单粒子系统的一维情况. 这时粒子速度和位置的迭代方程可进一步简化为

$$v(t+1) = wv(t) + \phi (z - x(t)), \quad (7)$$

$$x(t+1) = x(t) + v(t+1), \quad (8)$$

其中将 w,z,ϕ 固定为常数. 该系统还可描述为

$$v_{t+1} = wv_t + \phi y_t, \quad (9)$$

$$y_{t+1} = -wv_t + (1-\phi) y_t, \quad (10)$$

其中 $y_t = z - x_t$. 令

$$P_t = \begin{bmatrix} v_t \\ y_t \end{bmatrix}, \quad (11)$$

则

$$M = \begin{bmatrix} w & \phi \\ -w & 1-\phi \end{bmatrix} \quad (12)$$

为系统的系数矩阵;系统的动力方程可写为 $P_{t+1} = MP_t$ 或 $P_t = M^t P_0$,亦即系统完全由 M 定义. M 的特征根为

$$\begin{cases} e_1 = \dfrac{1 + w - \phi + \sqrt{(\phi - w - 1)^2 - 4w}}{2}, \\ e_2 = \dfrac{1 + w - \phi - \sqrt{(\phi - w - 1)^2 - 4w}}{2}. \end{cases} \quad (13)$$

当 $(\phi - w - 1)^2 - 4w \neq 0$ 时,M 有相异的特征根,这时存在可逆矩阵

$$A = \begin{bmatrix} \phi + w - 1 + \sqrt{(\phi - w - 1)^2 - 4w} & 2\phi \\ \phi + w - 1 - \sqrt{(\phi - w - 1)^2 - 4w} & 2\phi \end{bmatrix}, \quad (14)$$

使得

$$AMA^{-1} = L = \begin{bmatrix} e_1 & 0 \\ 0 & e_2 \end{bmatrix}. \quad (15)$$

如果定义 $Q_t = AP_t$,可以得到

$$Q_{t+1} = LQ_t, \quad (16)$$

或

$$Q_{t+1} = L^t Q_0 = \begin{bmatrix} e_1^t & 0 \\ 0 & e_2^t \end{bmatrix} Q_0. \tag{17}$$

因此系统收敛的充分必要条件是,特征根 e_1 和 e_2 的模均小于 1,即 $|e_1| < 1$ 和 $|e_2| < 1$. 下面分别进行讨论.

1) 当 $(\varphi - w - 1)^2 - 4w = 0$ 时,系统有两个相同的特征根,$e_1 = e_2 = \frac{1 + w - \varphi}{2}$,因此系统收敛的充要条件为

$$\begin{cases} \left|\frac{1 + w - \varphi}{2}\right| < 1, \\ (\varphi - w - 1)^2 - 4w = 0; \end{cases} \tag{18}$$

2) 当 $(\varphi - w - 1)^2 - 4w > 0$ 时,系统有两个相异的实根,系统收敛的充要条件为

$$\begin{cases} -2 < 1 + w - \varphi + \sqrt{(\varphi - w - 1)^2 - 4w} < 2, \\ -2 < 1 + w - \varphi - \sqrt{(\varphi - w - 1)^2 - 4w} < 2, \\ (\varphi - w - 1)^2 - 4w > 0, \end{cases} \tag{19}$$

即

$$\begin{cases} -2 < 1 + w - \varphi < 2, \\ (\varphi - w - 1)^2 - 4w > 0; \end{cases} \tag{20}$$

3) 当 $(\varphi - w - 1)^2 - 4w < 0$,系统有复数根,其模为

$$|e_1| = |e_2| = \sqrt{\frac{(1 + w - \varphi)^2}{4} + \frac{4w - (\varphi - w - 1)^2}{4}} = \sqrt{w}, \tag{21}$$

因此系统收敛的充要条件为

$$\begin{cases} w < 1, \\ (\varphi - w - 1)^2 - 4w < 0. \end{cases} \tag{22}$$

综合 1) ～ 3) 三种情况,在 $\varphi - w$ 平面中,保证系统收敛的 φ 与 w 的取值范围如图 1 中阴影部分所示.

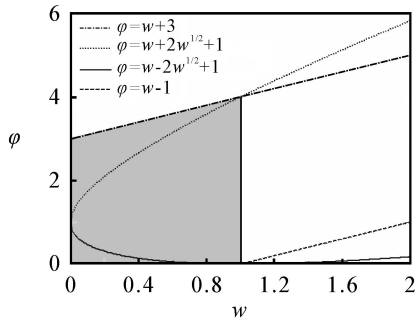

图 1　PSO 算法的收敛区域示意图

令阴影区域为 A,则当 $(w, \varphi) \in A$ 时粒子收敛,当 $(w, \varphi) \in \overline{A}$ 时粒子发散,这是本文所提算法的控制方式的理论基础.

3.2　算法思想

ARPSO 算法增强了在多峰优化问题中寻找全局最优解的概率,但是对于多样性的控制却显得比较简单. 算法并没用充分发挥多样性控制方法的潜力,并且 ARPSO 算法对其他参数如粒子群规模、迭代次数等的灵敏度太大,不易调节. 因此在本文提出的算法中增强了对多样性的控制,充分利用多样性信息来控制算法的搜索过程. 首先给出了粒子群多样性的度量公式如下[8,9]:

$$\text{diversity}(S) = \frac{1}{|S| \cdot |L|} \cdot \sum_{i=1}^{|S|} \sqrt{\sum_{j=1}^{D} (p_{ij} - \overline{p_j})^2}. \tag{23}$$

其中:S 表示粒子群,$|S|$ 表示粒子群的大小,$|L|$ 表示搜索空间的最大对角距离,D 表示问题的维数,p_{ij} 表示第 i 个粒子的第 j 维的值,$\overline{p_j}$ 表示当前所有粒子在第 j 维的平均值.

在所提新算法中,与 ARPSO 算法一样,定义了粒子群的两种状态,即收缩状态和发散状态;还定义了多样性的上限(d_{high}) 和下限(d_{low});当粒子群的多样性 diversity 小于 d_{low} 时,粒子就开始作远离粒子群中心的发散运动,直到多样性大于 d_{high},一旦大于 d_{high},粒子停止发散运动,开始向着中心作收缩运动;粒子群在多样性的指导下作收缩 - 发散运动,同时使得粒子群的多样性不会出现标准 PSO 中随搜索进程而逐渐减小的情况,可以避免 PSO 易陷入局部最优.

对于粒子群收缩和发散状态的控制,本文采用如下的方法:

首先,将粒子的速度更新公式写为

$$v_{id} = W \cdot v_{id} + C_1 \cdot \text{rand}() \cdot (p_{id} - x_{id}) + C_2 \cdot \text{Rand}() \cdot (p_{gd} - x_{id}), \tag{24}$$

其中:参数 rand() 和 Rand() 的含义与式(1) 和式(3) 中的对应参数完全相同,但 W,C_1,C_2 的取值范围是根据粒子群所处的阶段来定. 根据分析结果,它们按照如下要求取值:在发散阶段,取值区域为 $(W, \varphi) \in \overline{A}$;在收缩阶段,取值区域为 $(W, \varphi) \in A$. 其中 $\varphi = C_1 \cdot \text{rand}() + C_2 \cdot \text{Rand}()$. 亦即当参数 $(W, \varphi) \in \overline{A}$ 时,单个粒子是发散的,这样导致整个粒子群发散,从而使多样性增加. 当 $(W, \varphi) \in A$ 时,单个粒子是收敛的,粒子群的范围收缩,从而多样性就减少. 这样通过不断地使粒子群在收缩和发散阶段之间转换,可以使之持续地搜索下去直到找到最优解,本文称所提算法为 DCPSO. 算法流程如图 2 所示.

3.3　算法参数选择

下面给出PSO算法中在单个粒子一维情况下各

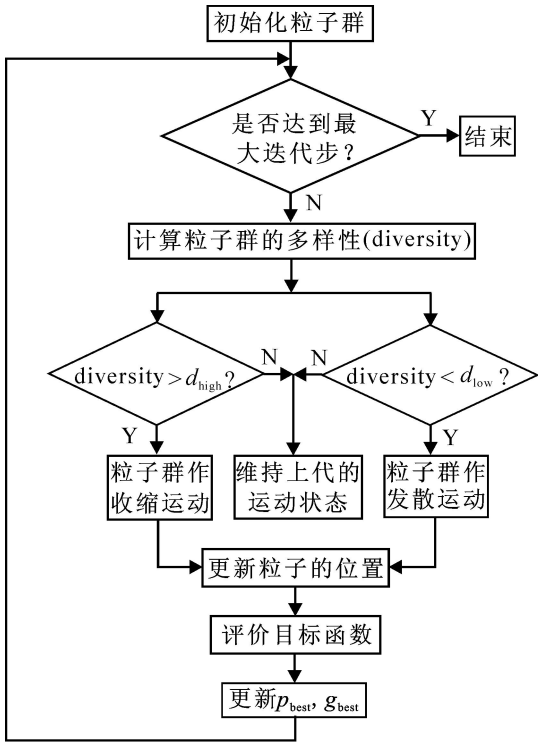

图 2 DCPSO 算法流程图

组参数的不同运动状态,迭代公式为式(9)和(10);其仿真条件为:$x_0 = 2$, $v_0 = -0.1$, $z = 0$,迭代次数为 100.

图 3 中,X 轴表示迭代次数,Y 轴表示函数值. 从图 3(a) ~ 3(l) 可以看出,当 $W > 1$ 时,无论怎样设置 φ 的值,粒子始终是发散的;当 $W = 1$ 时,粒子处于振荡状态;当 $W < 1$ 时,粒子的收敛行为还要视 φ 的值而定.

从图中还可看出,当 $W = 0.65$, $\varphi = 0.1$ 时,粒子以较快的速度渐近收敛;当 $W = 0.65$, $\varphi = 1.5$ 时,粒子振荡收敛;而当 $W = 0.65$, $\varphi = 4$ 时,粒子振荡发散;当 $W = 3$ 时,粒子均为发散的. 因此以上结果表明,粒子收敛性的仿真实验与 3.1 节的理论分析完全一致. 选择算法的具体参数值时,可以参考这个仿真实验. 在选择收缩阶段的参数值时,一般应使粒子收敛不能太快,又不能太慢;发散阶段的参数选择,W 值只要大于 1,就能保证发散,而且 W 与 C_1 和 C_2 的值越大,发散越快.

4 实验与讨论

4.1 实验设置与结果

本文使用与文献[8]中相同的标准测试函数,具体描述如表 1. Rosenbrock 函数是一个比较难优化的单峰函数,它的函数曲面上有一段狭长的区域,具有非凸的病态特征;Rastrigin 函数是非线性的多峰函数,有大量的局部最优点. 两个测试函数的全局最优值都为 0.

表 1 标准测试函数

测试函数	公式	搜索空间
Rosenbrock	$f_3(\vec{x}) = \sum_{i=1}^{n-1}(100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	[-100, 100]
Rastrigin	$f_4(\vec{x}) = \sum_{i=1}^{n}(x_i^2 + 10 - 10\cos(2\pi x_i))$	[-5.12, 5.12]

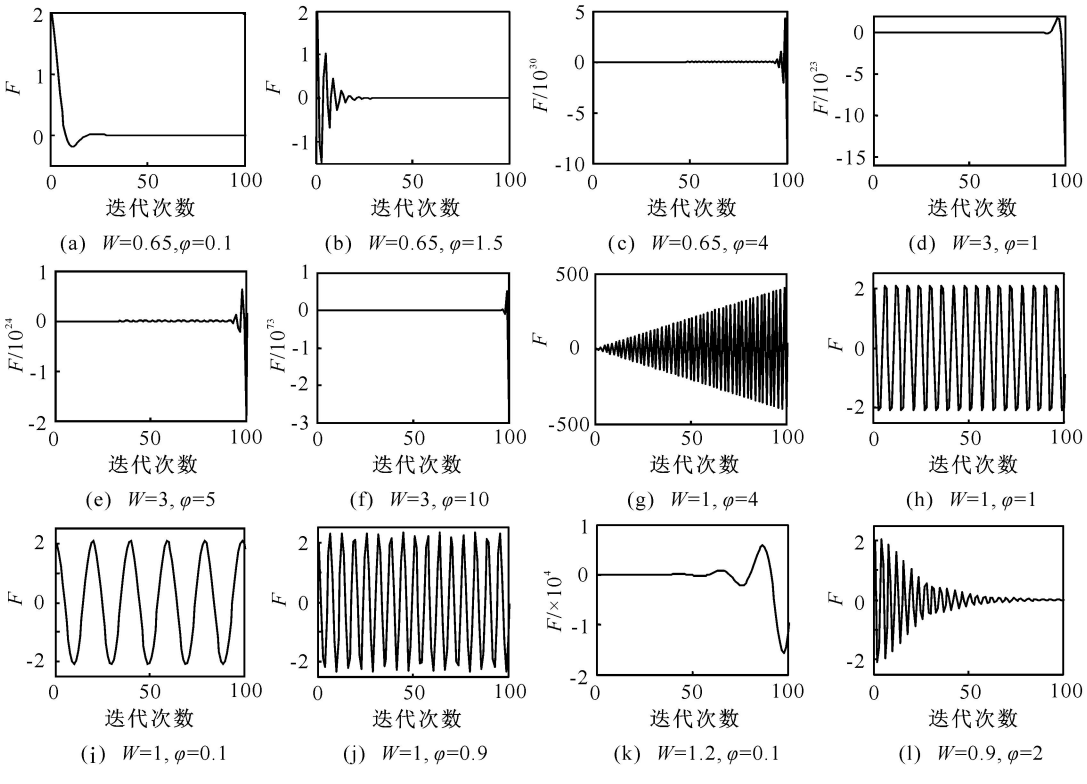

图 3 各种参数下单个粒子一维的运动状态

表2　实　验　结　果

算法	维数 测试函数	20	50	100	迭代次数 / %		g_{best} 更新率 / %	
					收缩阶段	发散阶段	收缩阶段	发散阶段
DCPSO	Ackley	7.976E-10			92.56	7.44	97.67	2.33
			4.36E-08		93.40	6.60	99.07	0.93
				1.991	94.40	5.60	99.73	0.27
ARPSO[8]		3.30E-08			76.90	23.10	98.50	1.50
			0.027		78.40	21.60	98.90	1.10
				0.218	81.10	18.90	98.90	1.10
DCPSO	Rosenbrock	0.87708			99.99	0.01	100.00	0.00
			1.3497		99.55	0.45	100.00	0.00
				83.289	96.76	3.24	99.97	0.03
ARPSO[8]		2.34			69.20	30.80	88.80	11.20
			10.43		71.30	28.70	89.30	10.70
				103.46	79.20	20.80	95.40	4.60

为了与文献[8]的结果比较,采用了相同的实验,即对每个函数进行 3 组实验,每组实验的粒子数均为 20;问题维数分别为:20,50,100;最大迭代次数为问题维数的 1000 倍;每组实验都应用 DCPSO 算法随机独立运行 50 次.

从文献[8]中可以知道 ARPSO 算法的实验结果及算法思想,但是作者没有具体地给出其所得结果对应的详细参数,故无法复原其中的实验. 为了比较的合理性,本文直接采用[8]中的实验结果作比较,并且对于非算法相关的具体参数采用了与[8]中相同的参数设置,即 $d_{low} = 5E\text{-}6$, $d_{high} = 0.25$;另外,设置 DCPSO 算法在发散状态时 $W = 3$, $C_1 = C_2 = 10$;收缩状态时 $W = 0.65$, $C_1 = C_2 = 1.5$,这种参数选择与 Clerc 的具有压缩因子的 PSO 算法[10]的参数设置比较接近.

表 2 中列出了文献[8]的实验结果以及本文每组实验的最优解的平均值、各阶段的迭代次数的百分比和 g_{best} 在两个阶段的更新率. 从表 2 的结果可以看出,DCPSO 算法在这两个测试函数中所取得的结果基本上都比 ARPSO 算法好,尤其是 20 维和 50 维的测试中,结果都远远优于 ARPSO 算法;在 ARPSO 算法和 DCPSO 算法中,粒子最优位置的更新基本都发生在收缩阶段,在发散阶段更新的很少或者几乎没有;但在 ARPSO 算法中,由于粒子从发散状态跳出的速度较慢而使得发散阶段在整个搜索过程中占了较高的比例;在 DCPSO 算法中发散阶段相对所占比例非常小.

4.2　实验结果讨论

对于 DCPSO 算法,算法一开始以较快的速度收敛,这样粒子群将比 ARPSO 算法中的粒子群更快收敛到多样性度量的下限. 如果将粒子群从初始化开始到多样性值第 1 次小于下限值的进化阶段称为阶段 1,则这段时间(即迭代次数) 若用 T1 表示,那么对于DCPSO算法,其 $T1_{DCPSO}$ 小于ARPSO算法的 $T1_{ARPSO}$. 也就是当两个算法同时进化到 $T1_{DCPSO}$ 时,DCPSO 算法已经收敛到多样性的下限值,而 ARPSO 算法还处于阶段 1. 对于一般的进化类算法,在群体初始化后的进化初期,多样性的迅速减小对于算法能够很快找到比较好的解是有利的. 因此在 $T1_{DCPSO}$ 时刻,DCPSO 算法找到的解(g_{best}) 一般优于 ARPSO 算法找到的解. 在 $T1_{DCPSO}$ 以后,DCPSO 算法的粒子群开始进入扩散 - 收缩的相互转化阶段,即阶段 2,这一阶段的时间用 T2 表示,则 T2 = MAX_Iteration - T1. 因此 $T2_{DCPSO}$ 远大于 $T2_{ARPSO}$.

在阶段 2,DCPSO 算法以较快的速度扩散,以适当的速度收敛. ARPSO 算法由于粒子群是通过粒子的反向运动发散的,发散速度较慢,并以较快的速度收敛. 从仿真实验结果的分析可知,不论哪种算法,在发散过程中目标函数更新的可能性很小,在收缩阶段更新的可能性很大. 这是因为,在进入阶段 2 后,目标函数值已取得了一个相对较好的值,粒子的 p_{best} 可更新的区域已经比较小,因此 g_{best} 的更新区域也比较小. 随着 p_{best} 和 g_{best} 的不断更新,它们的更新区域越来越小,而且一般距粒子群中心较近的区域总存在可更新区域. 当然,在距粒子群比较远的距离也可能存在可更新区域,但不如距中心区域的可能性大. 在发散过程中,如果粒子错过了更新区域,它在接下来的发散过程中基本不会再落入更新区域中,它只会越来越远离更新区域;如果在发散过程中,粒子恰好落入更新区域,则接下来更新区域会减小. 因为以前的更新区域会包含它,所以在以后的发散过程中,随着粒子的远离,粒子很难再落入新的更新区域中. 而在收缩过程中,粒子向粒子群的中心靠拢,粒子很容易进入更新区域,并且如果在某一时刻进入更新区域,使得以后的更新区域变小;但由于粒

子不断地向中心靠拢,其再次落入更新区域的可能性依然很大. 以上讨论可以解释为:在阶段 2,粒子在发散过程中目标函数的更新少,而在收缩阶段目标函数的更新多.

粒子的发散过程是粒子接下来的收缩过程的前提,即只有先将多样性增大,才能使多样性减小. 既然发散过程目标函数值很少更新,那么就应该使该过程耗费的时间(迭代次数) 尽量少,留下尽可能多的时间(迭代次数) 给收缩过程. 从表 2 的实验数据可以看到,DCPSO 算法做到了这一点,因此它搜索到的最终结果优于 ARPSO 算法.

DCPSO 算法和 ARPSO 算法先通过阶段 1 的基本收敛过程找到一个次优解,在此基础上,通过阶段 2 的发散和收缩两个过程,使多样性维持在一定的水平(下限值以上) ,粒子的更新域不断缩小. 而因为最优解邻域始终是包含于粒子更新域中的,从而使粒子越来越接近最优解邻域,这就是在 DCPSO 算法和 ARPSO 算法中引入多样性控制策略的目的,只不过 DCPSO 算法的控制策略比 ARPSO 算法更有效,性能更好.

5 结 论

本文在分析了 PSO 算法收敛条件的基础上,提出了多样性控制的 PSO 算法. 通过对多个标准测试函数的实验结果可以看出,DCPSO 算法在复杂优化问题中具有较强的全局优化性能,而且也比 ARPSO 算法具有更好的优化效果,表明了本文所提的控制方式的有效性和可行性. 接下来的研究工作是将 DCPSO 算法应用于实际的工程优化中.

参考文献(References)

[1] Kennedy J, Eberhart R. Particle swarm optimization [C]. Proc of IEEE Int Conf of Neural Networks. Perth, 1995: 1942-1948.

[2] Clerc M, Kennedy J. The particle swarm — Explosion, stability and convergence in a multidimensional complex space[J]. IEEE Trans on Evolutionary Computation, 2002, 6(2): 58-73.

[3] Mendes R, Kennedy J, Neves J. The fully informed particle swarm: Simpler, maybe better[J]. IEEE Trans on Evolutionary Computation, 2004, 8: 204-210.

[4] Shi Y, Eberhart R C. A modified particle swarm[C]. Proc of IEEE Int Conf on Evolutionary Computation. Anchorage, 1998: 1945-1950.

[5] 曾建潮, 崔志华. 微粒群算法的统一模型及分析[J]. 计算机研究与发展, 2006, 43(1): 96-100.
(Zeng J C, Cui Z H. A new unified model of particle swarm optimization and its theoretical analysis[J]. J of Computer Research and Development, 2006, 43(1): 96-100.)

[6] 刘洪波, 王秀坤, 谭国真. 粒子群优化算法的收敛性分析及其混沌改进算法[J]. 控制与决策, 2006, 21(6): 636-645.
(Liu H B, Wang X K, Tan G Z. Convergence analysis of particle swarm optimization and its improved algorithm based on chaos[J]. Control and Decision, 2006, 21(6): 636-645.)

[7] 周驰, 高亮, 高海兵. 基于粒子群优化算法的约束布局优化[J]. 控制与决策, 2005, 20(1): 37-40.
(Zhou C, Gao L, Gao H B. Particle swarm optimization based algorithm for constrained layout optimization[J]. Control and Decision, 2005, 20 (1): 37-40.)

[8] Riget J, Vesterstrφm J S. A diversity-guided particle swarm optimizer — The ARPSO [R]. Denmark: University of Aarhus, 2002.

[9] Rasmus K Ursem. Diversity-guided evolutionary algorithms[C]. Proc of Parallel Problem Solving from Nature VII(PPSN-2002). Heidelberg: Springer-Verlag, 2002: 462-471.

[10] Clerc M. The swarm and the queen: Towards a deterministic and adaptive particle swarm optimization [C]. Proc of the Congress on Evolutionary Computation. Washington: IEEE Service Center, 1999: 1951-1957.