

PSO 算法的稳定性分析及算法改进

朱小明¹ 张慧斌²

(北京师范大学信息科学与技术学院 北京 100875)¹ (忻州师范学院计算机科学与技术系 忻州 034000)²

摘 要 种群多样性的缺失是导致 PSO 算法易陷入早熟早收敛的重要原因,因此对基于线性定常离散系统的 PSO 算法的稳定性作了理论分析,并分析了种群多样性缺失的原因,根据此特性提出了一种惯性权重因子在一定范围内随机取值且学习因子取恒定常数的改进 PSO 算法,该算法可以使粒子速度具有一定的概率发散,以保持种群的多样性。通过对 3 个约束优化问题的仿真实验表明,该算法跳出局部极值的概率很大,可有效地避免早熟早收敛。

关键词 PSO 算法,线性定常离散系统,稳定性分析,早熟早收敛,种群多样性

中图法分类号 TP273 **文献标识码** A

Stability Analysis of Particle Swarm Optimization Algorithm and its Improved Algorithm

ZHU Xiao-ming¹ ZHANG Hui-bin²

(College of Information Science and Technology, Beijing Normal University, Beijing 100875, China)¹

(Department of Computer Science and Technology, Xinzhou Teachers College, Xinzhou 034000, China)²

Abstract The loss of population diversity is an important reason which leads to the premature convergence of the PSO algorithm. Therefore, the stability of the PSO algorithm based on the linear time-invariant discrete system was analyzed theoretically and the possible reasons of the lack of the population diversity were discussed in this paper. Based on the stability of the algorithm, an improved PSO algorithm was presented in which the inertia weight factor value is got randomly within a certain range and the learning factor value is a constant. In the algorithm the population diversity can be maintained by the character that the particle speed has certain probability. The simulation experiments of three constraint optimization problems show that the algorithm has great probability to jump out of local extremum, and avoids the precocious premature convergence effectively.

Keywords PSO algorithm, Linear time-invariant discrete systems, Stability analysis, Premature convergence, Population diversity

粒子群优化算法(PSO)是一种基于群体智能的随机启发式搜索算法,作为一种高效并行优化算法,它可用于解决大量非线性、不可微和多峰值的复杂优化命题,而且算法程序本身实现简洁,需要调整的参数相对较少,因而发展很快,并成功应用于多个学科和工程领域。但是,同其它群体智能优化算法一样,PSO 算法容易产生早熟早收敛、全局收敛性能差的问题。

早熟早收敛导致算法陷入局部极值而无法实现全局优化。当前学者们对导致早熟早收敛的原因有一个共同的认识:算法多样性不足使算法易陷入局部极值从而引起早熟早收敛^[1]。因此很多学者通过增加群体多样性来提高算法的性能,从而增加收敛到全局最优的概率。文献[2]通过多样性的衡量标准来控制种群的多样性,从而实现粒子间的吸引和排斥,以避免算法的早熟;文献[3]采用互斥多种群和反收敛来实现多样性的动态调节;文献[4]提出全信息 PSO 算法,它通过向邻近粒子学习来降低共享信息的流速,从而增加粒子的

多样性;还有文献通过引入突变因子^[5]、随机扰动^[6]、混沌算子^[7]和引入模拟退火的思想^[8],使粒子有一定的概率逃离局部极值区域。上述方法和策略虽然在一定程度上提高了种群的多样性,但是对 PSO 算法的稳定性理论分析不够,因此在种群多样性的分析上不够精确。

本文在线性定常离散系统下对 PSO 算法的稳定性(即收敛性)条件作了严格的数学理论推导,得出了 PSO 算法的收敛性和发散性的条件,在此基础上,提出了一种改进的 PSO 算法。该算法通过使粒子的速度有一定概率发散来保持种群的多样性,从而增强其搜索全局最优值的能力。

1 PSO 算法的数学模型

PSO 算法通过跟踪个体极值及全局极值来搜索解空间的最优值,在一维空间里,粒子的速度位置更新公式表示为:

$$V(t+1) = \omega V(t) + c_1 r_1(t)(P_i(t) - X(t)) + c_2 r_2(t)(P_g(t) - X(t)) \quad (1)$$

到稿日期:2012-05-30 返修日期:2012-09-20 本文受国家社科基金项目(BHA100068)资助。

朱小明(1959—),男,硕士,高级工程师,主要研究方向为计算机网络应用,E-mail:zhuxm@elec.bnu.edu.cn;张慧斌(1971—),男,硕士,讲师,主要研究方向为人工智能和密码学。

$$X(t+1)=X(t)+V(t+1) \quad (2)$$

式中, c_1, c_2 为正常数, 称为加速因子; r_1, r_2 为 $[0, 1]$ 之间的随机数; ω 是惯性权重因子, 如果是变量则设为 $\omega(t)$ 。为了便于叙述和分析, 设 $\varphi_1(t) = c_1 r_1(t)$ 、 $\varphi_2(t) = c_2 r_2(t)$, 粒子的速度式(1)就演化为:

$$V(t+1) = \omega(t)V(t) + \varphi_1(t)(P_i(t) - X(t)) + \varphi_2(t)(P_g(t) - X(t)) \quad (3)$$

令 $\varphi(t) = \varphi_1(t) + \varphi_2(t)$, 文献[9]对式(2)、式(3)作线性变换, 把 PSO 算法的粒子速度和位置变化公式写成矩阵的形式:

$$\begin{bmatrix} V(t+1) \\ X(t+1) \end{bmatrix} = \begin{bmatrix} \omega(t) & -\varphi(t) \\ \omega(t) & 1-\varphi(t) \end{bmatrix} \begin{bmatrix} V(t) \\ X(t) \end{bmatrix} + \begin{bmatrix} \varphi_1(t) & \varphi_2(t) \\ \varphi_1(t) & \varphi_2(t) \end{bmatrix} \begin{bmatrix} P_i(t) \\ P_g(t) \end{bmatrix} \quad (4)$$

设 $Z(t) = \begin{bmatrix} V(t) \\ X(t) \end{bmatrix}$, $P(t) = \begin{bmatrix} P_i(t) \\ P_g(t) \end{bmatrix}$, $A(t) = \begin{bmatrix} \omega(t) & -\varphi(t) \\ \omega(t) & 1-\varphi(t) \end{bmatrix}$, $B(t) = \begin{bmatrix} \varphi_1(t) & \varphi_2(t) \\ \varphi_1(t) & \varphi_2(t) \end{bmatrix}$, 则式(4)改写成:

$$Z(t+1) = A(t)Z(t) + B(t)P(t) \quad (5)$$

显然, 式(5)是一个线性时变离散系统的数学模型, 也就是标准的 PSO 的粒子速度和位置变化公式演化成线性时变离散系统的数学模型。

假设 $\omega(t)$ 、 $P_i(t)$ 、 $P_g(t)$ 、 $\varphi_1(t)$ 、 $\varphi_2(t)$ 均为迭代过程中不变的常数, 设:

$$\begin{aligned} \varphi_1 &= \varphi_1(t), \varphi_2 = \varphi_2(t), \\ \varphi &= \varphi(t) = \varphi_1(t) + \varphi_2(t) = \varphi_1 + \varphi_2; \\ P_i &= P_i(t), P_g = P_g(t), \omega = \omega(t), \end{aligned}$$

$$A = A(t) = \begin{bmatrix} \omega & -\varphi \\ \omega & 1-\varphi \end{bmatrix}, B = B(t) = \begin{bmatrix} \varphi_1 & \varphi_2 \\ \varphi_1 & \varphi_2 \end{bmatrix};$$

则式(5)由线性时变离散系统转化为线性定常离散系统, 方程如下:

$$Z(t+1) = AZ(t) + BP \quad (6)$$

令 $p = \frac{\varphi_1 P_i + \varphi_2 P_g}{\varphi}$, $y(t) = p - X(t)$, 则式(4)演变为:

$$\begin{bmatrix} V(t+1) \\ y(t+1) \end{bmatrix} = \begin{bmatrix} \omega & -\varphi \\ -\omega & 1-\varphi \end{bmatrix} \begin{bmatrix} V(t) \\ y(t) \end{bmatrix} \quad (7)$$

令 $A^* = \begin{bmatrix} \omega & -\varphi \\ -\omega & 1-\varphi \end{bmatrix}$, $Z^* = \begin{bmatrix} V(t) \\ y(t) \end{bmatrix}$, 则式(7)演变为:

$$Z^*(t+1) = A^* Z^*(t) \quad (8)$$

则有:

$$Z^*(t) = (A^*)^t Z^*(0) \quad (9)$$

显然, 式(10)的特征多项式和特征多项式方程分别为:

$$f(\lambda) = \lambda^2 + (\varphi - 1 - \omega)\lambda + \omega \quad (10)$$

$$\lambda^2 + (\varphi - 1 - \omega)\lambda + \omega = 0 \quad (11)$$

2 基于线性定常离散系统的稳定性分析

对于线性定常离散系统, 系统矩阵的全部特征根在复平面的单位圆内是线性定常离散系统稳定的充分必要条件。

求解特征多项式方程(式(11)), 得到迭代矩阵 A^* 的特征根为:

$$\lambda_{1,2} = \frac{(\omega + 1 - \varphi) \pm \sqrt{(\omega + 1 - \varphi)^2 - 4\omega}}{2} \quad (12)$$

线性定常离散系统稳定性的充分必要条件是 $|\lambda_{1,2}| < 1$,

即:

$$\left| \frac{(\omega + 1 - \varphi) \pm \sqrt{(\omega + 1 - \varphi)^2 - 4\omega}}{2} \right| < 1 \quad (13)$$

解不等式(13), 得: $\varphi < 2\omega + 2$ 和 $|\omega| < 1$; 由 $p = \frac{\varphi_1 P_i + \varphi_2 P_g}{\varphi}$, 可得 $\varphi > 0$ 。

于是, 得到基于线性定常离散系统的 PSO 算法的稳定性判定条件的定理 1。

定理 1 基于线性定常离散系统的 PSO 算法的稳定性充要条件是:

$$0 < \varphi < 2(\omega + 1) \text{ 且 } -1 < \omega < 1, \varphi > 0 \quad (14)$$

由以上分析可知, 当 $-1 < \omega < 1$ 时, 基于线性定常离散系统的 PSO 算法的稳定性如下:

当 $0 < \varphi < 2(\omega + 1)$ 时, 算法是稳定的(即收敛);

当 $0 < \varphi = 2(\omega + 1)$ 时, 算法是临界稳定的;

当 $\varphi > 2(\omega + 1)$ 时, 算法是不稳定的(即发散)。

3 PSO 算法的改进

文献[10]指出, 当局部极值的测度远大于粒子当前迭代的粒子的飞行速度时, 收敛速度快的 PSO 算法的粒子一旦进入局部极值区域就很难从中跳出。这是标准 PSO 算法容易陷入局部极值区域的根本原因。

观察基本 PSO 算法的式(1)可以看出, 粒子速度收敛速度快, 粒子飞行速度 $V(t)$ 快速下降趋于 0, 即 $|V(t+1)| > |V(t)|$ 的概率太小甚至没有, 这就是各类文献所说的粒子活性缺失, 使得粒子很难跳出局部极值区域, 这是 PSO 算法易陷入早熟早收敛的根本原因。

当粒子飞行速度不稳定时, $|V(t+1)| > |V(t)|$ 的概率比较大, 粒子的收敛速度慢, 因此粒子的活性也较大, PSO 算法本身的稳定性也差, 粒子的速度具有一定的多样性, 可以使 PSO 算法有很大概率跳出局部极值区域。

文献[2]提出的 ARPSO 算法根据一定的评判标准来判断粒子活性的大小, 根据评判标准认为粒子活性很小时, 就让粒子发散。但是, 粒子活性评判标准的计算量远远大于粒子速度和位置变化公式的计算量, 因此很不实用。

要让粒子速度有较大概率发散, 必须降低粒子收敛速度, 保持粒子活性, 保持算法的多样性。根据上述线性定常离散系统下的 PSO 算法的稳定性可知, 当 $\varphi > 2(\omega + 1)$ 时, 算法是不稳定的、发散的。

标准的 PSO 算法是线性时变离散系统, 为了让其更接近于线性定常离散系统, 本文提出一种惯性权重因子在一定范围内随机取值并且去掉学习因子的随机参数 r_1, r_2 (即 φ_1, φ_2 取常数) 的改进 PSO 算法, 称之为 WPSO 算法, 其速度更新公式如下:

$$V(t+1) = \omega(t)V(t) + c_1(P_i(t) - X(t)) + c_2(P_g(t) - X(t)) \quad (15)$$

$\omega(t)$ 是一定范围内的随机值,使其有一定概率: $\varphi > 2(\omega + 1)$, 这样粒子的速度有可能收敛也有可能发散,保持了种群的多样性。

文献[11]提出了一种惯性权重因子 ω 随机取值的 PSO 算法来增大粒子的活性,但是其 φ_1 、 φ_2 取随机数,因此不能确定粒子的发散概率。

对于 WPSO 算法,一般来说, $\omega(t)$ 数学期望值 $E(\omega(t))$ 只要满足 $c_1 + c_2 < 2(E\omega(t) + 1)$ 且 $-1 < \omega(t) < 1$, WPSO 算法就是依概率收敛稳定的。

例如,当 $\omega(t)$ 在 0.5 和 1.0 之间随机取值, $c_1 + c_2 = 3.4$ 时, $\omega(t)$ 落在 0.5 和 0.7 之间的概率为 40%, 此时 WPSO 算法是发散的;同理 $\omega(t)$ 落在 0.5 和 0.7 之间的概率为 60%, 此时 WPSO 算法是依概率收敛的。

用一个仿真实例 $f(x) = x^2$ (搜索空间是 $[-100, 100]$, 粒子数为 10 个)来分别观察基于线性定常离散系统的 PSO 算法、标准的 PSO 算法和 WPSO 算法的粒子的收敛和发散性。

对于线性定常离散系统的 PSO 算法,取 $\omega = 0.7$, φ_1 、 φ_2 为恒定常数。

当取 $\varphi = \varphi_1 + \varphi_2 < 3.4$ 时, PSO 算法的所有粒子都收敛;当取 $\varphi = \varphi_1 + \varphi_2 > 3.4$ 时,所有粒子都发散;当 $\varphi = \varphi_1 + \varphi_2 = 3.4$ 时,粒子的速度和位置变化过程初期是振荡发散,振幅逐渐变大,最后进入等幅周期振荡状态,粒子处于线性定常离散系统,这称为临界稳定状态。随机地取一个粒子,在临界稳定状态下其速度和位置的变化过程如图 1、图 2 所示。

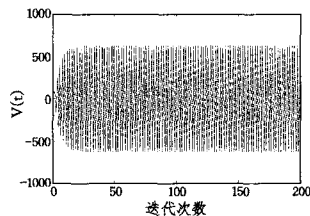


图 1 当 $\varphi = 2(\omega + 1)$ 时粒子的速度变化情况

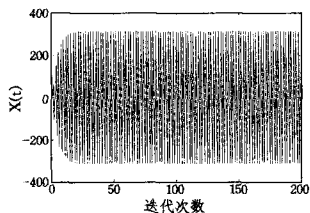


图 2 当 $\varphi = 2(\omega + 1)$ 时粒子的位置变化情况

对于 PSO 算法,取 $\omega = 0.7$, $c_1 = c_2 = 1.7$,虽然 $\varphi = 2(\omega + 1)$,但整个算法依然稳定收敛,随机地取一个粒子,其速度变化过程如图 3 所示。

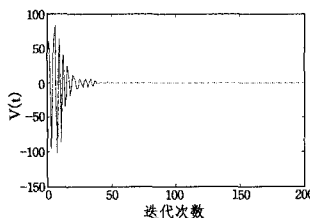


图 3 PSO 算法一个粒子的速度变化过程

对于 WPSO 算法, ω 在 0.5 和 1.0 之间随机地取值,取 $c_1 = c_2 = 1.7$,算法收敛,随机地取一个粒子,其速度变化过程如图 4 所示。

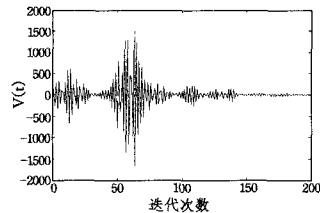


图 4 WPSO 算法一个粒子的速度变化过程

由图 3 可以看出, PSO 算法粒子的收敛速度非常快,在迭代后期粒子没有活性,因此在求解多极值的优化问题和多极值的约束优化问题时,很容易陷入局部极值,容易导致早熟早收敛现象,但是实验表明 PSO 算法适合求解单极值的优化问题。

由图 4 可以看出, WPSO 算法粒子的收敛速度比较慢,粒子有很大的概率发散,即 $|V(t+1)| > |V(t)|$ 的概率比较大,因此粒子速度变化较大,在迭代后期粒子仍具有较大的活性,显然有较大概率跳出局部极值区域,但是实验表明在求解单极值优化问题时,在同样的迭代次数下,收敛精度明显比 PSO 差,因此 WPSO 算法不适合求解单极值的优化问题,适合求解多极值优化问题。

4 仿真实验及分析

为了测试评价 WPSO 算法求解多极值优化问题的性能,下面采用 3 个约束优化问题作为测试函数,并与一些其它改进的 PSO 算法和其它算法的测试结果作了比较。WPSO 算法参数设置为:粒子群规模 $m = 20$, $c_{1,2} = 1.7$, 迭代次数 $K = 200$, 随机惯性系数 $\omega(t)$ 在 0.5 和 1.0 之间随机地取值。

问题 1 Himmelblau 的非线性约束优化问题

$$\text{Min}_i f(X) = 5.3578547x_3^2 + 0.8356891x_1x_5 + 37.293239x_1 - 40792.141 \quad (16)$$

$$\text{s. t. } g_1(X) = 85.334407 + 0.0056858x_2x_5 + 0.00026x_1x_4 - 0.0022053x_3x_5$$

$$g_2(X) = 80.51429 + 0.0071317x_2x_5 + 0.0029955x_1x_2 - 0.0021813x_3^2$$

$$g_3(X) = 9.300961 + 0.0047026x_3x_5 + 0.0012547x_1x_5 + 0.0019085x_3x_4$$

$$0 \leq g_1(X) \leq 92, 90 \leq g_2(X) \leq 110$$

$$20 \leq g_3(X) \leq 25, 78 \leq x_1 \leq 102$$

$$33 \leq x_2 \leq 45, 27 \leq x_3, x_4, x_5 \leq 45$$

WPSO 算法对问题 1 的 10 次独立寻优所求结果均为 -31025.56, 标准偏差为 0。从表 1 可以看出,很多算法找到问题 1 的解是 -30665.5 左右,说明 -30665.5 是问题 1 的一个局部极值, WPSO 算法可以有效地跳出问题 1 的 -30665.5 这个局部极值,并且比其它改进的 PSO 算法和其他算法的收敛精度高。

表1 不同算法求解问题1的最优解

变量	文献[11]的 7种不同算法	PSASM 算法[12]	PSODP[13]	DE, DESQI 算法[14]	WPSO 算法
x_1		78.00	78.00		78.00
x_2		33.00	33.00		33.00
x_3		27.07	27.07		27.07
x_4		44.99	45.00		45.00
x_5		44.96	44.97		44.97
$f(X)$	-30665.539	-31025.02	-31025.53	-30665.54	-31025.56

问题2 压力管设计问题。该问题的数学模型描述如下:

$$\text{Min: } f(X) = 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 + 19.84x_1^2x_3 \quad (17)$$

$$\text{s. t. } g_1(X) = -x_1 + 0.0193x_3 \leq 0$$

$$g_2(X) = -x_2 + 0.00954x_3 \leq 0$$

$$g_3(X) = -\pi x_3^2x_4 - \frac{4\pi}{3}x_3^2 + 1296000 \leq 0$$

$$g_4(X) = x_4 - 240 \leq 0$$

$$0 \leq x_1 \leq 99, 0 \leq x_2 \leq 99, 10 \leq x_3 \leq 200, 10 \leq x_4 \leq 200$$

WPSO算法对问题2的10次独立寻优中,有6次寻找到4348.862935,有4次找到6059左右。从表2可以看出,迄今已知的算法找到问题2的最优解都在6059左右,说明6059是问题2的一个局部极值,很多算法都无法跳出这个局部极值,而WPSO算法可以有较大概率跳出问题2的局部极值,有效地避免陷入早熟收敛。

表2 不同算法求解问题2的最优解

变量	FPSOA[15]	文献[16]	文献[17]	文献[18]	WPSO 算法
x_1	0.812500	0.8125	0.8125	0.8125	1.00
x_2	0.473500	0.4735	0.4375	0.4375	1.00
x_3	42.098445	40.3239	42.0974	42.0913	40.9196
x_4	176.636596	176.6541	176.6541	176.7465	200.00
$f(X)$	6059.7143	6288.7445	6059.9463	6061.0777	4348.8629

问题3 伸缩绳设计问题。该问题的数学模型描述如下:

$$\text{Min: } f(X) = (x_3 + 2)x_2x_1^2 \quad (18)$$

$$\text{s. t. } g_1(X) = 1 - \frac{x_2^3x_3}{71785x_1^4} \leq 0$$

$$g_2(X) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \leq 0$$

$$g_3(X) = 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0$$

$$g_4(X) = \frac{x_1 + x_2}{1.5} - 1 \leq 0$$

$$0.05 \leq x_1 \leq 2, 0.25 \leq x_2 \leq 1.3, 2 \leq x_3 \leq 15$$

从表3可以看出,WPSO算法对问题3的10次独立寻优所求结果最好解为0.01266529,与文献[15]的最优解几乎一样,但是最差解为0.01271905,标准偏差为4.3760E-5,因此远比文献[15]提供的最差解0.0130801、标准偏差8.4611E-5好得多。

表3 不同算法求解问题3的最优解

变量	FPSOA[15]	文献[16]	文献[17]	文献[18]	WPSO 算法
x_1	0.05168069	0.051480	0.051989	0.051728	0.05170806
x_2	0.35651634	0.3561661	0.353965	0.357644	0.35707983
x_3	11.30078272	11.632201	10.890522	11.244543	11.24218903
$f(X)$	0.01266520	0.012705	0.012681	0.012675	0.01266529

结束语 本文对基于线性定常离散系统的PSO算法的稳定性和发散性作了详细的理论分析,根据分析结果提出了一种惯性权重因子在一定范围内随机取值且学习因子取一定恒定常数的改进PSO算法,即WPSO算法,该算法能让粒子有确定的概率值发散,这样保持了种群的多样性,有很大概率搜索到全局最优解,可以有效地避免早熟收敛。

参考文献

- [1] 辛斌,陈杰,窦丽华,等.群搜索优化中基于分布熵的多样性控制[J].模式识别与人工智能,2009,22(3):374-380
- [2] Riget J, Vesterstroem J S. A Diversity-Guided Particle swarm optimizer—The ARPSO[R]. Aarhus, Germany: University of Aarhus, Department of Computer Science, 2002
- [3] Blackwell T, Branke J. Multiswarms, Exclusion, and Anti-Convergence in Dynamic Environments[J]. IEEE Trans on Evolutionary Computation, 2006, 10(4): 459-472
- [4] Mendes R, Kennedy J, Neves J. The fully informed particle swarm: Simpler, maybe better[J]. IEEE Trans on Evolutionary Computation, 2004, 7(8): 204-210
- [5] Niu Q, Jiao B, Gu X S. Particle swarm optimization combined with genetic operators for job shop scheduling problem with fuzzy processing time[J]. Applied Mathematics and Computation, 2008, 205(1): 148-158
- [6] Zhao X C. A perturbed particle swarm algorithm for numerical optimization[J]. Applied Soft Computing, 2010, 10(1): 119-124
- [7] Coelho L D S, Lee C S. Solving economic load dispatch problems in power systems using chaotic and Gaussian particle swarm optimization approaches[J]. Electrical Power and Energy Systems, 2008, 30(5): 297-307
- [8] 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): 47-50
- [9] 张慧斌,王鸿斌,胡志军.基于定常线性迭代法的PSO算法收敛性分析[J].计算机工程与应用, 2011, 47(31): 35-37
- [10] 张慧斌,王鸿斌,胡志军. PSO算法全局收敛性分析[J].计算机工程与应用, 2011, 47(34): 61-63
- [11] 王勇,蔡自兴,周育人,等.约束优化进化算法[J].软件学报, 2009, 20(1): 11-29
- [12] 高显忠,罗文彩,侯中喜.应用改进PSO算法求解待约束优化问题[J].计算机仿真, 2009, 26(10): 212-215
- [13] 李炳宇,萧蕴诗,汪镭. PSO算法在工程优化问题中的应用[J].计算机工程与应用, 2004, 40(1): 74-76
- [14] 李宏.求解几类复杂优化问题的进化算法及其应用[D].西安:西安电子科技大学, 2009
- [15] 庞兴,王勇. PSO与捕鱼策略相结合的优化方法[J].计算机工程与应用, 2011, 47(8): 36-40
- [16] Coello C A C. Use of a self-adaptive penalty approach for engineering optimization problems[J]. Computers in Industry, 2000, 41: 113-127
- [17] Coello C A C, Montes E M. Constraint-handling in genetic algorithms through the use of dominance-based tournament selection[J]. Advanced Engineering Informatics, 2002, 16: 193-203
- [18] He Q, Wang L. An effective coevolutionary particle swarm optimization for constrained engineering design problems[J]. Engineering Applications of Artificial Intelligence, 2007, 20: 89-99