

QPSO 混合算法在 PID 控制器优化中的应用^①

胡 振

(南充职业技术学院 信息与管理工程系, 南充 637000)

摘 要: 为了提高 PID 控制器参数优化的速度和成功率, 提出了一种基于改进量子粒子群优化与非线性规划的混合算法. 在算法中引入了柯西分布随机数扰动项, 以增加粒子群体的多样性, 增强其全局搜索性能; 并将非线性规划函数融入算法, 使其局部搜索能力也得到改善, 从而提高了算法的精度和收敛速度. 将该算法应用于 4 阶对象的 PID 控制器参数优化, 用 MATLAB 编程运行, 结果表明其响应时间短、超调量小、稳定性好, 具有一定的实用性和推广价值.

关键词: 量子行为粒子群优化算法; 柯西分布; 非线性规划; 混合算法; PID 控制器

Application of QPSO Hybrid Algorithm in PID Controller Optimization

HU Zhen

(Information & Management Engineering, Nanchong Professional Technic College, Nanchong 637000, China)

Abstract: A hybrid algorithm based on improving QPSO and nonlinear programming is proposed to improve the speed and success rate of PID controller parameter optimization. The Cauchy random disturbance is introduced to the algorithm to increase the diversity of the particle population and enhance its global search capability. Nonlinear programming function is integrated in the algorithm to improve the capabilities of the local search, thus improving the accuracy and convergence rate. The algorithm is applied to fourth-order object of PID controller parameter optimization. Use MATLAB programming and run it, the results show the response time is short, the overshoot is small, the stability is good, and has certain practicality and promotional value.

Key words: QPSO; cauchy distribution; nonlinear programming; hybrid algorithm; PID controller

比例 - 积分 - 微分 (Proportion Integration Differentiation, PID) 控制器因结构简单、容易实现、鲁棒性强、适应面宽而广泛应用于工业过程控制, 其核心问题是参数的优化与整定. 近年来, 模糊控制、自整定技术及参数优化方法是 PID 控制器研究与应用的热点^[1,2], 而各种文献提出的 PID 控制器参数优化方法大都是粒子群优化 (Particle Swarm Optimization, PSO) 算法^[3,4]. 由于 PSO 算法并非全局收敛算法, 不能确保搜索到最优参数值, 为此本文提出一种基于改进量子行为粒子群优化 (Quantum-behaved Particle Swarm Optimization, QPSO) 与非线性规划的混合算法来解决该问题.

QPSO 算法是一种全局收敛算法, 但仍有可能发

生早熟现象. 在 QPSO 算法中增加柯西分布随机数扰动项, 能够提高粒子群体的多样性, 增强全局搜索能力, 防止陷入局部最优; 在此基础上嵌入非线性规划函数, 则可改善算法的局部搜索性能, 加快其收敛速度. 用标准函数测试该混合算法, 并将其应用于 PID 控制器参数的优化, 结果表明了其有效性.

1 改进 QPSO 与非线性规划混合算法

1.1 QPSO 算法描述

在 N 维搜索空间中, 由 M 个代表潜在问题解的粒子组成群体 $X = \{x_1, x_2, \dots, x_M\}$, 在 t 时刻, 第 i 个粒子位置为 $X_i(t) = [x_{i,1}(t), x_{i,2}(t), \dots, x_{i,N}(t)]$, $i=1, 2, \dots, m$, 粒子没有速度向量. 个体最好位置表示:

^① 收稿时间: 2014-02-25; 收到修改稿时间: 2014-03-27

为 $P_i(t)=[p_{i,1}(t),p_{i,2}(t),\dots,p_{i,N}(t)]$, 群体的最好位置为 $G(t)=[g_1(t),g_2(t),\dots,g_N(t)]$, 且 $G(t)=P_g(t)$, 其中 g 为群体最好位置粒子的下标, $g \in \{1,2,\dots,M\}$.

对于最小化问题, 目标函数值越小, 对应的适应度值越好. 粒子 i 的个体最优位置由下式确定:

$$P_i(t) = \begin{cases} X_i(t) & f[X_i(t)] < f[P_i(t-1)] \\ P_i(t-1) & f[X_i(t)] \geq f[P_i(t-1)] \end{cases} \quad (1)$$

群体的最优位置由式(10)、(11)确定:

$$G(t) = P_g(t), \quad g = \arg \min_{1 \leq i \leq M} \{f[P_i(t)]\} \quad (2)$$

则 QPSO 算法的粒子位置更新方程为:

$$C(t)=[c_1(t),c_2(t),\dots,c_n(t)]=\frac{1}{M} \sum_{i=1}^M P_i(t) \quad (3)$$

$$P_{i,j}(t) = \varphi_j(t)P_{i,j}(t) + [1 - \varphi_j(t)]G_j(t) \quad (4)$$

$$X_{i,j}(t+1) = P_{i,j}(t) \pm \beta \cdot |C_j(t) - X_{i,j}(t)| \cdot \ln \frac{1}{u_{i,j}(t)} \quad (5)$$

式(3)~(5)中: t 为当前迭代次数; $\varphi_j(t)$ 和 $u_{i,j}(t)$ 为 $[0,1]$ 区间的随机数; $C(t)$ 为粒子群的平均最优位置; β 为扩张收缩因子, 其取值对算法的收敛速度有直接关系^[5]. 在大多数情况下, 令 β 在 $[1.0,0.5]$ 区间线性递减, 可取得较好的优化结果^[6]. 若 β_0, β_1 分别为 β 的初值和终值, t, t_{\max} 分别为当前迭代次数和最大迭代次数, 则 β 的值为:

$$\beta(t) = (\beta_0 - \beta_1)(t_{\max} - t) / t_{\max} + \beta_1 \quad (6)$$

1.2 QPSO 算法的改进

作为一种群体智能搜索算法, QPSO 算法也难以完全避免早熟现象的发生, 其产生条件为: ① 部分粒子收敛到当前全局最优; ② 相邻粒子之间的距离很小. 对于随机搜索过程, 条件①是不可避免的, 因为粒子之间的比较必然会产生当前全局最优. 于是, 解决算法早熟问题的主要途径就是使条件②不成立^[7].

本文对 QPSO 算法的粒子位置增加柯西分布随机数扰动项, 以使上述条件②不成立, 称此改进算法为 CQPSO 算法. 改进公式为:

$$cauchyrnd(0,1) = \tan(\pi \cdot (rand() - 0.5)) \quad (7)$$

$$x_{i,j} = x_{i,j} + x_{i,j} \cdot cauchyrnd(0,1) \quad (8)$$

式(7)~(8)中: $cauchyrnd(0,1)$ 为标准柯西分布随机数; $rand()$ 为 $[0,1]$ 区间的均匀分布随机数; $x_{i,j}$ 为粒子个体的当前位置.

柯西分布具有两翼较长的概率特性, 它比高斯分布的范围更宽, 很容易产生远离原点的随机数^[8]. 因此, 给粒子位置增加柯西分布随机数扰动项, 可使其远离相邻粒子, 尽快跳出局部最优区域, 从而提高粒子的多样性, 增强全局搜索能力, 防止算法陷入早熟. 与此类似的方法还有高斯变异、混沌扰动等, 以后者应用较多. 混沌扰动以差异微小的随机混沌变量序列为基础, 将其变换到优化变量的取值范围, 若算法发生早熟收敛则在一定范围内进行混沌搜索, 使处于停滞状态的粒子离开原来的位置, 从而摆脱局部极小, 增强全局搜优能力^[9].

但是, 混沌扰动对初始条件极度敏感, 需调整混沌变量的取值范围, 且混沌搜索过程也要进行多次迭代, 因此算法的复杂度和运算量明显增加; 相比之下, 柯西扰动则算法简单, 对整体运行速度的影响可忽略不计. 值得注意的是, 这种改进策略会导致粒子位置超出优化变量取值范围的几率增加, 所以应在随后的步骤中进行粒子越界处理.

设增加柯西分布随机数扰动项之后的粒子位置为 $x_{i,j}$, 优化变量的取值范围为 $[rl,rr]$, 则越界处理可按下式进行:

$$\begin{cases} \text{if } x_{i,j} > rr \text{ then } x_{i,j} = rr \\ \text{else if } x_{i,j} < rl \text{ then } x_{i,j} = rl \end{cases} \quad (9)$$

1.3 改进 QPSO 与非线性规划的结合

CQPSO 算法的全局搜索性能得到了增强, 但同时也可能导致局部搜索能力减弱而降低其收敛速度. 为此, 我们在 CQPSO 中嵌入非线性规划函数构成一种混合算法, 称之为 NCQPSO 算法.

1.3.1 非线性规划

非线性规划(nonlinear programming)研究 n 元实函数在一组等式或不等式约束条件下的极值问题, 且目标函数和约束条件至少有一个是未知量的非线性函数. 其一般形式为:

$$\begin{aligned} & \min f(x) \\ & s.t. \begin{cases} g_i(x) = 0 & i = 1, 2, \dots, p \\ h_j(x) \leq 0 & j = 1, 2, \dots, q \end{cases} \end{aligned}$$

其中: $x = (x_1, x_2, \dots, x_n)^T \in E^n$, f, g_i, h_j 是定义在 E_n 上的实值函数. 称 x 为决策变量、 f 为目标函数、 g_i, h_j 分别为等式约束和不等式约束.

1.3.2 CQPSO 与非线性规划的结合

求解非线性规划问题的经典算法皆以梯度下降法为基础,其特点是局部搜索能力较强.因此,将 CQPSO 与非线性规划结合,既可增强算法的搜索能力,提高其计算精度,又能加快收敛速度.

Matlab 提供了求解非线性规划问题的函数 $fmincon()$, 可从一个预估值开始搜索约束条件下多元非线性函数的最小值.其约束条件如下:

$$\min f(x) \rightarrow \begin{cases} A \cdot x \leq b \\ Aeq \cdot x = beq \\ c(x) \leq 0 \\ ceq(x) = 0 \\ lb \leq x \leq ub \end{cases}$$

其中: x 为变量, A 、 b 为线性不等约束, Aeq 、 beq 为线性相等约束; c 、 ceq 分别为非线性不等约束和非线性相等约束; lb 、 ub 分别为 x 的下界与上界. x 、 b 、 beq 、 lb 和 ub 是向量, A 和 Aeq 为矩阵, $f(x)$ 、 $c(x)$ 和 $ceq(x)$ 为非线性函数.

$fmincon()$ 函数的调用格式为:

$$x = fmincon(fun, x0, A, b, Aeq, beq, lb, ub, nonlcon, options) \quad (10)$$

式(10)中: fun 为目标函数, $x0$ 为 x 的初设值, lb 、 ub 为 x 的下界和上界, 其余参数可取默认值^[10].

将非线性规划函数嵌入到 CQPSO 算法中即构成 NCQPSO 算法: 令 CQPSO 算法每迭代 10 次即执行 1 次非线性规划, 每次调用时将粒子的当前位置传递给 $fmincon()$ 函数的参数 $x0$, 以此为初设值进行局部搜索, 从而提高算法的搜索精度并加快其收敛速度. NCQPSO 算法的步骤为:

Step 1: 在问题空间中初始化粒子群;

Step 2: 计算当前个体适应度值, 并根据式(1)更新粒子的个体最优位置;

Step 3: 以式(2)计算当前全局最优位置, 若其优于上次迭代的全局最优位置, 则更新群体的全局最优位置为该值;

Step 4: 由式(3)~(5)更新粒子的位置;

Step 5: 用式(8)对粒子位置施加柯西分布随机数扰动, 并以式(9)进行粒子越界处理;

Step 6: 若当前迭代次数为 10 的倍数, 则调用非线性规划函数;

Step 7: 重复 Step 2~6, 直到满足结束条件.

1.3.3 NCQPSO 算法测试

为验证算法的有效性, 选用三个基准函数分别以 QPSO、CQPSO 和 NCQPSO 算法进行测试. 设置运行参数为: 粒子群规模 30, 最大迭代次数 200; 以函数值小于 10^{-6} 为收敛条件, 分别独立运行 50 次. 基准函数及其测试结果分别如表 1、表 2 所示.

表 1 本文使用的基准函数

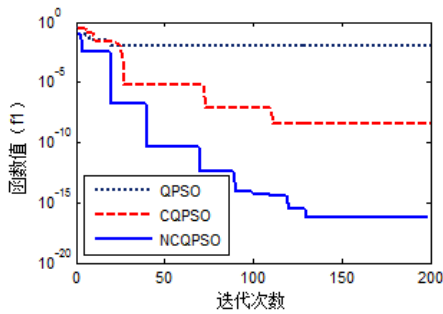
代号	函数名	函数表达式	变量区间
f_1	Schaffer F6	$0.5 + \frac{(\sin \sqrt{x_1^2 + x_2^2})^2 - 0.5}{(1.0 + 0.001(x_1^2 + x_2^2))^2}$	[-100,100]
f_2	Rastrigin	$\sum_{i=1}^{30} (x_i^2 - 10 \cos(2\pi x_i) + 10)$	[-5.12,5.12]
f_3	Griewank	$\sum_{i=1}^{30} \frac{x_i^2}{4000} - \prod_{i=1}^n \cos(\frac{x_i}{\sqrt{i}}) + 1$	[-600,600]

表 2 基准函数测试结果

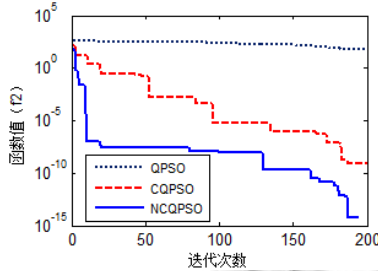
函数	算法	收敛次数	平均值	最优值	最差值
f_1	QPSO	2/50	0.0039	5.0166e-0 8	0.0097
	CQPSO	50/50(89)	1.4015e-08	1.1102e-16	2.3015e-07
	NCQPSO	50/50(18)	1.1102e-17	0	5.5511e-17
f_2	QPSO	0/50	82.9814	24.9217	174.5659
	CQPSO	50/50(141)	1.7892e-09	1.7764e-1 5	2.2533e-0 8
	NCQPSO	50/50(17)	8.2172e-1 2	0	8.1323e-1 1
f_3	QPSO	0/50	0.5768	0.0624	1.0260
	CQPSO	50/50(142)	1.4628e-0 8	8.7708e-1 5	1.2761e-0 7
	NCQPSO	50/50(10)	0	0	0

注: 括号中数据为达到收敛条件的平均迭代次数.

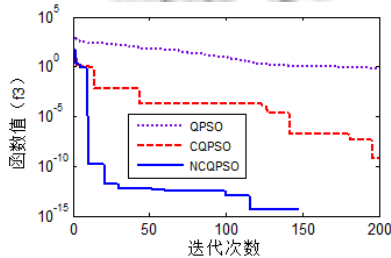
三个基准函数的理论最小值皆为 0, 其中 f_1 是二维多峰函数, f_2 、 f_3 是典型的非线性多模态函数, 它们都存在许多局部极小点, 很难找到其全局最小值. 表 2 的测试结果表明: 当粒子群规模较小、粒子维度和迭代次数较少时, QPSO 算法不能收敛到预设值, 且其寻优结果与理论值相差甚远; CQPSO 算法则显著改善了寻优效果, 三个基准函数都能稳定达到预设的收敛条件; 在此基础上, NCQPSO 算法进一步提高了结果精度, f_1 和 f_2 已经非常接近, f_3 则完全获得了理论最小值, 且其收敛速度也比 CQPSO 大为加快. 三种算法对各基准函数的寻优迭代过程如图 1 所示.



(a) 三种算法对 f_1 的寻优迭代过程



(b) 三种算法对 f_2 的寻优迭代过程



(c) 三种算法对 f_3 的寻优迭代过程

图 1 三种算法对各基准函数的寻优迭代过程

2 PID控制器优化问题

2.1 PID 控制器原理

PID 控制器根据给定值 $r(t)$ 与实际输出值 $y(t)$ 构成控制偏差 $e(t)$, 对其进行比例、积分和微分运算, 将三种运算的结果相加得到控制输出 $u(t)$ ^[11]. 其连续形式:

$$u(t) = K_p e(t) + K_i \int_0^t e(t) dt + K_d \frac{de(t)}{dt} \quad (11)$$

式(11)中: $e(t) = r(t) - y(t)$ 是系统误差, K_p 、 K_i 、 K_d 分别为对系统误差及其积分与微分量的加权.

PID 控制器的原理结构如图 2 所示:

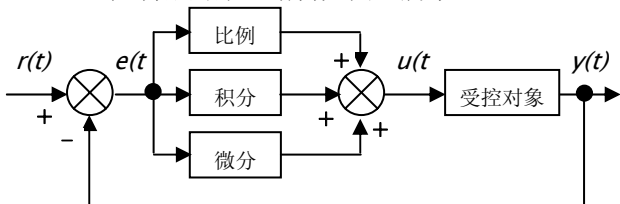


图 2 PID 控制器原理结构图

2.2 PID 控制器参数的优化

PID 控制器的优化问题即是确定一组合适的参数 K_p 、 K_i 、 K_d , 使其误差性能指标达到最优. 常用的误差性能指标有 ISE、IAE、ITAE、ISTE 等, 我们采用 ITAE(Integral of Time multiplied by Absolute Error, 积分时间绝对误差)指标, 其定义为:

$$T_{ITAE} = \int_0^T t|e(t)|dt \quad (12)$$

3 用NCQPSO算法优化PID控制器

3.1 建立 PID 控制模型

现以某不稳定系统为受控对象, 经试验、拟合得到如下 4 阶传递函数:

$$G(s) = \frac{s+2}{s^4+8s^3+4s^2-s+0.4} \quad (13)$$

已经确定相应 PID 控制器的 K_p 、 K_i 、 K_d 参数取值范围皆为 [0,100]. 以 ITAE 为误差性能指标, 可在 MATLAB 的 Simulink 环境中建立如图 3 所示的 PID 控制器模型:

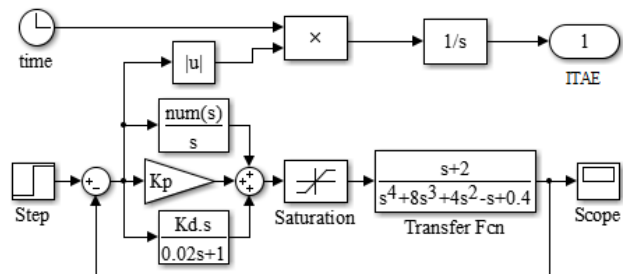


图 3 PID 控制器模型

3.2 PID 控制器参数的优化流程

用 MATLAB 编程实现 NCQPSO 算法, 在算法中以 PID 控制器模型为适应度函数, 运行算法程序即可实现控制参数的优化. 其流程如图 4 所示:

NCQPSO 算法与 Simulink 环境中的 PID 控制器模型之间通过粒子及其适应度值的传递实现交互: NCQPSO 算法产生粒子, 将其传递给 PID 控制器模型作为 K_p 、 K_i 、 K_d 参数值; PID 控制器模型以当前参数运行输出 ITAE 值, 将其返回 NCQPSO 算法作为该粒子的适应度值.

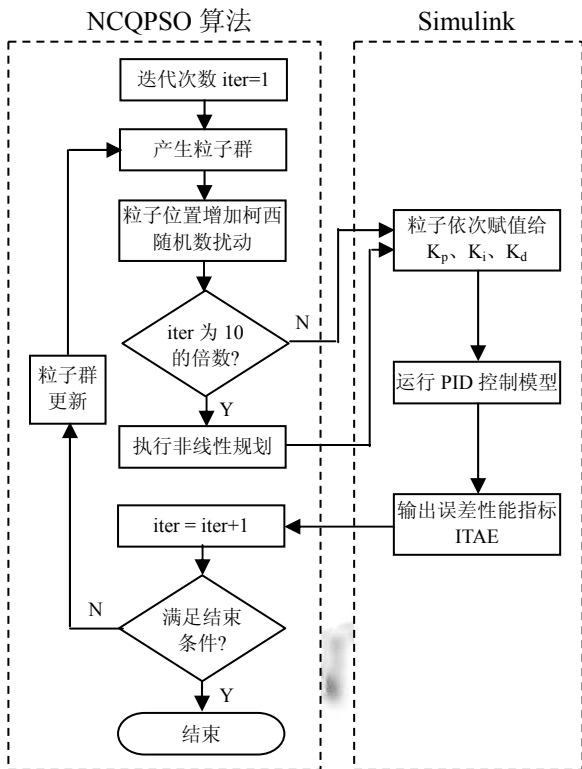


图 4 NCQPSO 算法优化 PID 控制器的流程

3.3 优化结果分析

程序运行时, NCQPSO 算法的运行参数设置为: 粒子群规模 50, 最大迭代次数 100, 各优化变量(分别对应于 PID 控制器参数 K_p 、 K_i 、 K_d)的取值范围[0,100]; 非线性规划函数 `fmincon()`的参数设置为: 目标函数 `fun` 为图 3 所示 PID 控制器模型, 初始值 `x0` 为粒子的当前位置, `lb`、`ub` 分别为[0 0 0]和[100 100 100]. 得到 PID 控制器参数优化结果为:

$$K_p = 34.5318, K_i = 0.1662, K_d = 39.8222$$

此时 $ITAE = 1.0581$, 迭代 81 次即可获得该值. $ITAE$ 与各参数的优化曲线分别如图 5、图 6 所示.

程序运行时, NCQPSO 算法的运行参数设置为: 粒子群规模 50, 最大迭代次数 100, 各优化变量(分别对应于 PID 控制器参数 K_p 、 K_i 、 K_d)的取值范围[0,100]; 非线性规划函数 `fmincon()`的参数设置为: 目标函数 `fun` 为图 3 所示 PID 控制器模型, 初始值 `x0` 为粒子的当前位置, `lb`、`ub` 分别为[0 0 0]和[100 100 100]. 得到 PID 控制器参数优化结果为:

$$K_p = 34.5318, K_i = 0.1662, K_d = 39.8222$$

此时 $ITAE = 1.0581$, 迭代 81 次即可获得该值. $ITAE$ 与各参数的优化曲线分别如图 5、图 6 所示.

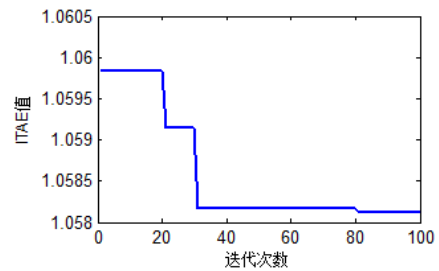
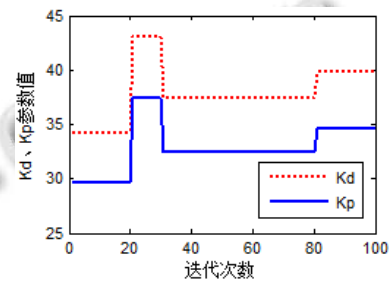
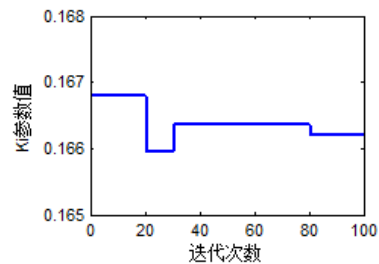


图 5 ITAE 指标优化曲线



(a) K_p 、 K_d 参数优化曲线



(b) K_i 参数优化曲线

图 6 各参数的优化曲线

将优化的 K_p 、 K_i 、 K_d 参数值回代 PID 控制器模型, 运行可得阶跃响应指标为: 上升时间 2.28s、稳定时间 7.18s、最大超调量 5.87%. 对于 4 阶控制对象来说, 这应该是一个较为满意的控制结果. 单位阶跃响应曲线如图 7 所示:

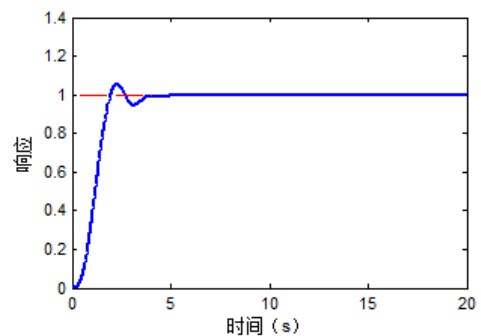


图 7 优化 PID 参数对应的单位阶跃响应曲线

4 结语

本文在改进 QPSO 算法中嵌入非线性规划函数构成 NCQPSO 算法,使全局搜索能力与局部搜索性能同时得到显著增强。测试结果表明,本算法比 QPSO 算法在计算精度与收敛速度上都大为提高;在此基础上设计了用本算法实现 PID 控制器优化的流程,并对一个 4 阶控制对象进行了编程仿真,结果表明优化之后的 PID 控制器响应快、超调量小、稳定性好,达到了较好的控制效果。

参考文献

- 1 周天沛,孙伟.基于粒子群模糊规则优化的水煤浆气化炉温度控制系统.计算机测量与控制,2013,21(4):913-915.
- 2 边丽华.PID 控制器参数自整定方法的研究与实现[学位论文].大连:大连理工大学,2009.
- 3 唐玉兰,徐明亮,梅娟,陈建慧.粒子群算法在 PID 控制器参数整定中的研究与应用.计算机工程与应用,2012,48(34):221-224.
- 4 黄胜忠,黄天开.一种改进粒子群算法在水轮机控制中的应用.计算机仿真,2012,29(4):345-348.
- 5 孙俊.量子行为粒子群优化算法研究.无锡:江南大学,2009.
- 6 方伟,孙俊,谢振平,须文波.量子粒子群优化算法的收敛性分析及控制参数研究.物理学报,2010,59(6):3686-3694.
- 7 徐文龙,孙俊,须文波.基于干扰因子的 QPSO 算法改进.微电子学与计算机,2008,25(4):128-130.
- 8 张兰.改进量子粒子群算法求解电力经济调度.计算机与数字工程,2013,41(8):1268-1269.
- 9 邹毅,朱晓萍,王秀平.一种基于混沌优化的混合粒子群算法.计算机技术与发展,2009,19(11):18-22.
- 10 史峰,王辉,郁磊,胡斐.MATLAB 智能算法 30 个案例分析.北京:北京航空航天大学出版社,2011.
- 11 黄友锐,曲立国.PID 控制器参数整定与实现.北京:科学出版社,2010:5-7.