

非线性动态自适应旋转角的量子菌群算法

刘 璐, 单 梁[†], 戴跃伟, 戚志东

(南京理工大学 自动化系, 南京 210094)

摘 要: 量子菌群算法是将量子理论引入到细菌觅食算法中的一种相对较新的组合优化算法, 虽然该算法在收敛速度上取得了一些重大的进步, 但是依然存在寻优时间较长的问题. 鉴于此, 设计一种非线性动态自适应旋转角, 并将其作用于细菌觅食算法的趋化操作, 提出一种非线性自适应旋转角的量子菌群算法. 基准函数的性能测试验证了该算法的正确性. 将所提出算法用于分数阶伺服系统的 PID 参数整定, 整定结果表明, 所提出的算法能有效地对伺服系统的 PID 控制器参数进行整定.

关键词: 量子理论; 菌群算法; 旋转角; 非线性; 分数阶; 伺服系统

中图分类号: TP301

文献标志码: A

Nonlinear notation angle for dynamic adaptation in quantum bacterial foraging optimization algorithm

LIU Lu, SHAN Liang[†], DAI Yue-wei, QI Zhi-dong

(School of Automation, Nanjing University of Science and Technology, Nanjing 210094, China)

Abstract: The quantum bacterial foraging algorithm is a relatively new combinatorial algorithm which combines the quantum evolutionary algorithm and the bacterial foraging algorithm. Although the algorithm has made some significant progress in the convergence speed, the algorithm still has the problem of longer searching time. In view of this, a quantum bacterial foraging algorithm with a nonlinear adaptive rotation angle is proposed. The performance test of the benchmark function proves the correctness of the algorithm. The proposed algorithm is applied to tune the PID parameters of the fractional-order servo system. The results show that the proposed algorithm can effectively tune the parameters of the PID controller in the servo system.

Keywords: quantum theory; bacterial foraging algorithm; rotation angle; nonlinear; fractional order; servo system

0 引 言

量子菌群算法(QBFO)是将量子理论引入菌群算法(BFO)中形成的一种新的智能优化算法^[1-2]. QBFO算法利用量子计算中量子的叠加态和纠缠态等特性实现量子的并行计算,用多量子比特编码染色体实现各基因位的叠加,增加了种群的多样性. 与传统的进化算法相比, QBFO算法具有更大的随机性和更强的全局搜索能力. 与量子遗传算法(QGA)^[3]一样,量子菌群算法最初被提出用于解决组合优化问题. 文献[4]利用量子菌群算法成功地解决了旅行商问题(TSP). 为了解决离散优化问题,文献[5]提出了一种类似的算法. 在此基础上,文献[6]进一步将量子菌群算法与粒子群遗传算法进行对比,并将其成功运用到频谱感知当中. 文献[7]对量子菌群算法的收敛

性进行了证明,并将其用于认知无线网络中的参数调整. 作为一种较新的智能优化算法,与其他智能算法类似, QBFO算法也存在寻优能力较差和鲁棒性差等问题,这些缺点限制了QBFO在实际中的应用.

对此,人们提出了多种改进策略^[8-10]. 文献[8]将量子粒子群算法与BFO算法相结合,提出了基于量子粒子群算法的菌群算法(QPSO-BFA). 文献[9]在此基础上加入了相对基学习,进一步改善了算法的收敛精度. 然而,在改善算法收敛精度的同时,还存在鲁棒性较差的问题. 文献[10]提出了一种线性自适应相位旋转的量子菌群算法(LAQBFO),但该算法只是简单地给出了旋转角的线性变化,并没有对此进行深入研究.

为了提高经典QBFO算法的鲁棒性和快速寻优

收稿日期: 2016-11-27; 修回日期: 2017-01-18.

基金项目: 国家自然科学基金项目(61374153).

作者简介: 刘璐(1990—), 男, 博士生, 从事伺服控制系统与智能控制算法的研究; 戴跃伟(1962—), 男, 教授, 从事多媒体安全、系统工程理论与网络安全等研究.

[†]通讯作者. E-mail: shanliang@njust.edu.cn

能力,本文提出一种非线性自适应旋转角的量子菌群算法(NAQBFO).在不同非线性调制指数情形下,通过多个基准函数对所提出算法的性能进行测试,然后将所提出算法与LAQBFO^[10]、QBFO^[1,2]、QGA^[3]三种相关算法进行对比.实验结果表明,与其他3种相关算法相比,所提出算法在保证较高的收敛精度的同时,能够有效地提高算法的寻优速度,提高优化效率.由于选取的3种对比算法都采用二进制编码,为了突出所提出算法的寻优能力,将所提出算法与实数编码经典粒子群算法(PSO)^[11]进行对比.对比结果表明,虽然PSO算法具有较快的寻优速度,但易陷入局部最优值,不能保证较好的寻优精度,而NAQBFO算法在保证相对较快的寻优速度的同时,能够保证较高的寻优精度.

1 QBFO算法

量子菌群算法是经典菌群算法改进的一个重要分支,将量子理论引入到菌群算法中,使其具有量子特性,大幅度增大搜索空间,提高算法的寻优能力.

1.1 量子位和量子门

与经典信息论中的比特类似,量子位是量子世界中的基本单位,也称为量子比特.量子比特一般采用Hilbert空间中的矢量表示,例如一个单量子比特可以用二维Hilbert空间中的量子态 $|0\rangle$ 和 $|1\rangle$ 表示为

$$\varphi = \alpha|0\rangle + \beta|1\rangle, \quad (1)$$

其中 α 和 β 为量子态的概率幅,两者都是复数且满足归一化条件 $\alpha^2 + \beta^2 = 1$.式(1)也说明了比特与量子比特的区别:后者不仅可以处于 $|0\rangle$ 态或 $|1\rangle$ 态,也可以处于两者的任意线性组合,即叠加态.

测量是连接量子世界和经典世界的纽带,量子比特不像比特那样具有确定的0态或1态,而是经过测量产生波包塌缩,塌缩到 $|0\rangle$ 和 $|1\rangle$ 的概率分别是 α^2 和 β^2 .类似于单量子比特, p 个量子比特可以表示为如下几个基本状态的叠加态:

$$\varphi = \alpha|0\rangle + \beta|1\rangle, \quad (2)$$

其中 a_x 满足归一化条件 $\sum_{x \in \{0,1\}^p} |a_x|^2 = 1$.

与经典信息论利用各种逻辑门处理比特一样,量子信息通过量子门处理量子比特.量子门要时刻保持酉性,即 $U^+U = I$.单比特量子门中最常用的是量子旋转门,该门可以使单量子比特向 $|0\rangle$ 态或 $|1\rangle$ 态旋转 θ 弧度,如下所示:

$$U(\theta) = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}. \quad (3)$$

1.2 QBFO算法设计

QBFO算法的核心是利用量子形式实现BFO算法中的趋化操作.将QBFO算法中的细菌用多量子比特的形式表示,然后经过测量形成二进制码,二进制码对应的十进制码即为细菌在搜索空间中的位置信息.利用量子旋转门来控制量子比特的概率幅,从而改变细菌在每个位置的概率,实现细菌位置的更新,具体流程如下.

Step 1: 参数初始化.种群个数 N ,趋化次数 N_c ,复制次数 N_{re} ,驱散次数 N_{ed} ,驱散概率 P_{ed} ,初始种群 $Q(t)$ 和旋转角 θ .

Step 2: 对 $Q(t)$ 进行测量生成二进制解集 $P(t)$,对 $P(t)$ 进行适应度评估,获取最佳适应度值作为下一步的目标值.

Step 3: 趋化操作.利用量子旋转门对种群 $Q(t)$ 进行更新,对 $Q(t)$ 进行测量生成 $P(t)$,对 $P(t)$ 进行适应度评估,取其最佳适应度值作为下一步的目标值.

Step 4: 复制操作.趋化操作完成后,对适应度值进行降序排序,将较差的一半剔除,将较好的一半复制,保持种群数目不变.

Step 5: 驱散操作.复制操作完成后,随机生成一个随机概率 P_{rand} ,如果 $P_{rand} < P_{ed}$,则进行驱散操作,否则保持染色体的位置和信息不变.

Step 6: 当达到收敛精度或者最大迭代次数时,输出最佳个体及其位置,算法结束,否则继续.

2 改进NAQBFO算法

在量子菌群算法中,利用量子旋转门来更新染色体的位置,选择不同的旋转相位,对算法的收敛速度和寻优能力将产生很大影响.在细菌觅食的实际过程中,旋转角度并不是一个常数,固定的旋转相位不利于算法的收敛,较大的旋转相位有利于探索新的区域,加快收敛速度,较小的旋转相位能促使细菌在局部区域内进行细搜索,寻找最优解.基于此,本文提出一种新的非线性自适应旋转角来改善QBFO的性能,有

$$\theta_i = -\text{sgn}(A_i)\theta_i. \quad (4)$$

其中: $-\text{sgn}(A_i)$ 表示旋转角的方向,保留文献[10]的取法; $A_i = \begin{bmatrix} \alpha_0 & \alpha_i \\ \beta_0 & \beta_i \end{bmatrix}$, $\begin{bmatrix} \alpha_0 \\ \beta_0 \end{bmatrix}$ 为当前最优细菌的某个量子比特对应的概率幅, $\begin{bmatrix} \alpha_i \\ \beta_i \end{bmatrix}$ 为当前细菌的某个量子比特对应的概率幅; θ_i 为旋转角的大小.

本文提出的非线性旋转角为

$$\theta = \left| \frac{\theta_0 - \theta_i}{C} \right|^m (\theta_{\max} - \theta_{\min}) + \theta_{\min}. \quad (5)$$

其中: θ_{\max} 为旋转角的最大值; θ_{\min} 为旋转角的最小值; θ_0 和 θ_i 分别为当前最优细菌和当前细菌的某个量子比特在单位圆上的角度; C 为一个常数, 表示 $|\theta_0 - \theta_i|$ 角度差的最大值; m 为当前演化步骤的非线性调制指数.

图1为 m 取不同值时旋转角与细菌角度差之间的关系, 图中 θ_{\max} 取 0.5π , θ_{\min} 取 0.001π , C 取 π . 当 $m = 1$ 时, 旋转角变为线性时的情况, 即线性情况下旋转角的变化是非线性情况下的特例.

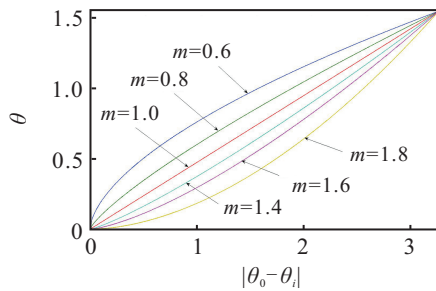


图1 在不同调制指数下旋转角与细菌角度差之间的关系

本文不仅提出了非线性自适应变化的旋转角, 而且对相应的参数(如 θ_{\max} 、 θ_{\min} 和 m)进行了合理的推导, 目的在于, 采用本文所提出算法解决给定问题时, 需要较少的计算量和少量的实验便可以确定所需参数, 给用户节省时间.

3 NAQBFO算法收敛性和参数分析

3.1 NAQBFO收敛性分析

定义1 假设 S 为种群个数, E 为搜索空间, 定义 $X^t = \{X_1^t, X_2^t, \dots, X_S^t\}$ 为第 t 次迭代的位置信息, f 为自适应函数值, f_{best} 为全局最优解, $f_{\text{best}} = \lim_{x_i^t \in X^t} \{f(x_i^t), i = 1, 2, \dots, S\}$.

定义2^[12] 对于一组随机序列 $\tau_n (n = 1, 2, \dots)$, 若存在变数 τ , 使 $p\{\lim_{n \rightarrow \infty} \tau_n = \tau\} = 1$, 或对于 $\forall \delta > 0$, 有 $p(\bigcap_{n=1}^{\infty} \bigcup_{k \geq n} |\tau_k - \tau| \geq \delta) = 0$, 则称随机序列 $\{\tau_n\}$ 以概率1收敛于随机变数 τ .

引理1^[13] NAQBFO的种群序列 $\{Q^t, t \geq 0\}$ 是有限状态空间的齐次马尔科夫链.

引理2(波莱尔-坎泰利(Borel-Cantelli))^[14] 设 B_1, B_2, \dots, B_n 是概率空间上相互独立的事件系列, $p(B_k)$ 是其概率, 则有以下两个结论:

- 1) $\sum_{n=1}^{\infty} p(B_k) < \infty$, 则 $p(\bigcap_{n=1}^{\infty} \bigcup_{k \geq n} B_k) = 0$;
- 2) $\sum_{n=1}^{\infty} p(B_k) = \infty$, 则 $p(\bigcap_{n=1}^{\infty} \bigcup_{k \geq n} B_k) = 1$.

引理3 在NAQBFO中存在常数 $\mu \in (0, 1)$, 使得

一步转移概率 $p_{22} < \mu$ ^[15], p_{ii} 表示由状态空间 A_i 转移到状态空间 A_j 的概率.

定理1 NAQBFO优化解序列 $\{X^t, t \geq 0\}$ 是有限状态空间的齐次马尔科夫链.

证明 NAQBFO中的状态转移过程是在有限空间中进行的, 且其量子编码、测量、趋化、复制和驱散等操作都是在独立随机过程中进行的, 整个过程都采用的是保留最优个体的策略, 第 $t + 1$ 代 Q^{t+1} 仅与第 t 代 Q^t 有关, 与之前的种群信息无关, 由引理1可知, $\{X^t, t \geq 0\}$ 是有限状态空间的齐次马尔科夫链. \square

定理2 NAQBFO算法以概率1收敛到全局最优.

证明 由定理1可知, NAQBFO的优化解序列 X^t 为一个有限状态空间的马尔科夫链, 其状态空间 $E = \{E_1, E_2, \dots, E_D\}$. 其中: $D = \prod_i^n \frac{a_i - b_i}{\delta}$, a 和 b 分别为搜索空间的上域和下域, $\delta > 0$ 为给定精度. 可将状态空间分为以下两种:

- 1) $A_i = \{E_i | |f_t - f_{\text{best}}| < \delta\}$;
- 2) $A_i = \{E_i | |f_t - f_{\text{best}}| \geq \delta\}$.

从 t 代进化到 $t + 1$ 代后, 不满足精度的概率为 p_{22} . 由引理3可知, $\sum_{t=1}^{\infty} p_{22} < \sum_{t=1}^{\infty} \mu$, 因为 $\mu \in (0, 1)$, 根据幂级数收敛定理可知 $\sum_{t=1}^{\infty} \mu = \frac{\mu}{1 - \mu}$, 即 $\sum_{t=1}^{\infty} p_{22} < \frac{\mu}{1 - \mu} < \infty$. 由引理2可知

$$p(\bigcap_{i=1}^{\infty} \bigcup_{t \geq i} |f_t - f_{\text{best}}| \geq \delta) = 0,$$

由定义2可知, 算法NAQBFO以概率1收敛到最优解. \square

3.2 NAQBFO算法参数分析

为了分析所提出NAQBFO算法的参数, 本文采用经典的Sphere函数, Sphere函数是 n 维并且只有一个最小值的函数, 即

$$f_1(x) = \sum_{d=1}^n x_d^2. \quad (6)$$

函数 f_1 在 $x = (0, 0, \dots, 0)$ 处取到全局最小值 $f_1(x) = 0$. 在分析过程中, 采用2阶Sphere函数模型, 初始搜索空间限制为 $x_d \in [-100, 100]$. 实验中细菌个数 S 取40, 趋化、复制和驱散迁徙操作的代数分别为 $N_c = 25$ 、 $N_{re} = 5$ 和 $N_{ed} = 4$, 驱散迁徙概率 P_{ed} 取0.25, 染色体长度 $length$ 取44, 角度差 C 取 π , 旋转角的最大值 θ_{\max} 取固定值. 对 $m = 1$ 的线性情形进行分析, θ_i 的斜率为

$$k = \frac{\theta_{\max} - \theta_{\min}}{C} \quad (7)$$

每一次仿真实验中,对于相同的 θ_{\max} 和 C , k 的不同取值决定了 θ_{\min} 的不同取值,进行100次仿真实验,实验结果如表1所示.其中:S.R.代表进行100次实验寻到最优值的成功率,Iter代表寻到最优值时所用的迭代次数.满足下式则表示寻优成功:

$$|f^w - f^*| < \varepsilon. \quad (8)$$

其中: f^* 为目标函数的全局最优解; f^w 为通过算法获取的最优解; ε 为精度系数,本文取 10^{-4} .

表1 θ_{\max} 和 k 取不同值时, $m = 1$ 情形下Shpere函数达到全局最优值的性能对比

θ_{\max}	k	S.R./%	Average	时间/s	Iter
0.1 π	0.095	100	8.311 1e-06	3.3933	227
	0.099	98	1.315 7e-04	3.400 6	284
0.2 π	0.099	100	2.653 6e-06	3.387 0	201
	0.195	100	8.975 0e-06	3.391 5	226
0.3 π	0.199	100	3.603 2e-05	3.391 5	269
	0.195	99	1.230 4e-05	3.412 5	209
0.4 π	0.199	100	2.368 7e-06	3.403 5	198
	0.295	99	4.096 5e-05	3.365 6	228
0.5 π	0.299	94	2.788 8e-04	3.396 4	270
	0.199	100	4.009 1e-06	3.416 6	203
0.6 π	0.295	100	1.588 1e-05	3.404 8	209
	0.299	99	1.046 4e-04	3.405 6	203
0.7 π	0.395	100	3.395 1e-06	3.394 0	223
	0.399	95	2.304 0e-04	3.383 3	277
0.8 π	0.295	100	2.130 0e-07	3.416 3	237
	0.299	100	9.135 8e-06	3.418 6	215
0.9 π	0.395	100	1.028 6e-05	3.365 6	218
	0.399	100	1.366 1e-05	3.378 5	200
1.0 π	0.495	100	1.732 0e-06	3.371 0	226
	0.499	94	4.160 7e-04	3.370 6	300

由表1可见,每一次仿真实验的平均消耗时间相差不大,波动范围在0.053 s以内,因此平均消耗时间在性能比较中占比重很小,几乎可忽略不计.在所有给出的组合中,当 $\theta_{\max} = 0.3\pi$, $k = 0.199$ 时,所提出算法的性能最好,此时 θ_{\min} 的取值为 $\theta_{\min} = \theta_{\max} - kC = 0.101\pi$.

$m = 1$ 线性情况下的最优组合已经确定,为了取得非线性情况下的最优组合,保持线性情况下 θ_{\max} 和 θ_{\min} 的最优取值不变,取不同的调制指数进行100次实验以确定非线性情况下的最优参数组合,结果

如表2所示.

表2 不同调制指数下Shere函数达到全局最优值时的性能对比

调制指数 m	S.R./%	Iter
0.1	100	202
0.2	100	207
0.4	100	209
0.6	100	212
0.8	100	210
1.0	100	198
1.1	100	194
1.3	100	198
1.4	100	214
1.6	100	201
1.8	100	188
2.0	100	198
2.2	100	206

由表2可见,当 $m = 1.8$ 时,Sphere函数取得最小迭代次数188.因此,调制指数 m 的合适取值有利于改善算法的性能.

相对于线性模型,当 $m = 1.8$ 时,较大的旋转角有助于大范围地搜索解空间,尽可能地遍历每一个可能存在最优解的解空间,但是值不能无限制地增大,这会使算法产生较大的震荡,影响算法性能甚至导致算法失控.当 $m > 1.8$ 时,在算法后期具有较快的收敛速度,但是较大的震荡不利于算法性能的改善.显然, $m = 1.8$ 是一个典型值,对于不同的研究对象, m 可能取不同的值,但是这种推理是行之有效的.

4 性能评估

利用表3中的9个经典基准函数对NAQBFO算法进行测试,并与其他4种算法进行性能对比,结果如表4所示.QBFO算法中, N_c 、 N_{re} 、 N_{ed} 、 P_{ed} 和length的取值与NAQBFO和LAQBFO的参数取值相同,转角步长 θ_0 取 0.08π ,QGA算法中,最大迭代次数 N 取500,交叉和变异概率分别为 $P_c = 0.7$ 和 $P_m = 0.15$,SPSO算法中,种群大小 S 取40,最大迭代次数取500,学习因子 c_1 和 c_2 取值均为1.5,惯性因子 w 取0.8.

由表4可见,从平均迭代次数分析,与LAQBFO算法相比,对于多峰函数 f_1 和 f_2 、单峰函数 f_6 和 f_9 而言,NAQBFO算法具有更快的收敛速度,对于其他函数两者具有相似的性能;从取得的平均最优值分析,NAQBFO算法具有比LAQBFO算法相似或更高

表3 基准函数基本信息

函数	公式	寻优范围	最优值
Griewank	$f_1(x) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_i \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-100, 100]^D$	0
Ackley	$f_2(x) = 20 + e - 20e^{-\frac{1}{5}\sqrt{\frac{1}{D}\sum_{i=1}^D x_i^2}} - e^{-\frac{1}{D}\sum_{i=1}^D \cos(2\pi x_i)}$	$[-100, 100]^D$	0
Rastrigrin	$f_3(x) = \sum_{i=1}^D [x_i^2 - 10 \cos(2\pi x_i^2) + 10]$	$[-100, 100]^D$	0
Schewefel	$f_4(x) = 418.9829D + \sum_{i=1}^D (-x_i \sin(\sqrt{ x_i }))$	$[-500, 500]^D$	0
Rosenbrock	$f_5(x) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-10, 10]^D$	0
Shubert	$f_6(x, y) = \left(\sum_{i=1}^5 i \cos((i+1)x + i)\right) \left(\sum_{i=1}^5 i \cos((i+1)y + i)\right)$	$[-100, 100]^D$	-186.7309
Schaffer	$f_7(x) = 0.5 - \frac{\sin^2 \sqrt{x_1^2 + x_2^2} - 0.5}{[1 + 0.001(x_1^2 + x_2^2)]^2}$	$[-100, 100]^D$	1
Sum Squares	$f_8(x) = \sum_{i=1}^D ix_i^2$	$[-100, 100]^D$	0
Dixon-Price	$f_9(x) = (x_1 - 1)^2 + \sum_{i=2}^D i(2x_i^2 - x_{i-1})^2$	$[-10, 10]^D$	0

表4 NABFO、LABFO、QBFO与QGA和DWQPSO算法的性能比较(D = 2)

函数	NAQBFO		LAQBFO		QBFO		QGA		SPSO	
	Iter	平均值	Iter	平均值	Iter	平均值	Iter	平均值	Iter	平均值
Griewank	178	2.2019e-07	193	9.0413e-06	293	2.2738e-04	313	5.9350e-03	125	1.6849e-02
Ackley	188	1.1773e-03	220	1.4299e-03	301	1.8844e-01	315	2.3753e-01	114	3.8460e-01
Rastrigrin	192	8.5985e-05	195	2.7934e-04	300	1.1920e+00	328	1.2123e+00	128	2.9849e-01
Schewefel	249	1.6849e+00	252	1.7693e+00	124	1.2609e+02	171	5.2406e+01	87	4.8232e+02
Rosenbrock	264	1.7460e-01	266	2.0631e-01	131	3.0704e+00	233	5.3869e+00	105	6.9559e+00
Shubert	275	-1.8567e+02	287	-1.8565e+02	121	-1.3081e+02	164	-1.6979e+02	94	-8.3287e+01
Schaffer	3	9.9043e-01	3	9.8614e-01	4	9.6373e-01	2	9.3093e-01	2	9.8032e-01
Sum Squares	212	2.7683e-07	213	2.9536e-06	308	5.3433e-03	366	6.5631e-03	176	3.6667e-01
Dixon-Price	247	2.8852e-02	306	3.5423e-02	177	2.4846e+00	141	1.7247e+00	137	2.7291e+00

表5 NABFO、LABFO、QBFO与QGA和DWQPSO算法的性能比较(D = 30)

函数	NAQBFO		LAQBFO		QBFO		QGA		SPSO	
	Iter	平均值	Iter	平均值	Iter	平均值	Iter	平均值	Iter	平均值
Griewank	234	1.0252e-01	239	1.1690e-01	362	1.1269e+00	353	4.7286e+00	154	1.5572e+00
Ackley	261	1.5857e-01	276	1.6004e+00	402	1.1078e+01	335	5.4582e+01	176	2.5978e+01
Rastrigrin	381	2.0961e+01	393	3.5393e+01	399	1.0229e+02	408	3.8394e+02	137	3.2015e+03
Schewefel	271	1.7355e+03	277	9.2676e+03	249	1.3313e+04	237	1.3148e+04	91	1.8424e+04
Rosenbrock	255	2.6086e+01	260	2.6369e+01	265	1.2098e+02	255	1.1461e+02	102	1.8105e+02
Sum Squares	252	2.1493e-01	259	2.5189e-01	321	6.4255e+00	317	8.617e+00	179	1.0306e+01
Dixon-Price	251	2.0162e+02	268	2.1785e+02	247	5.4392e+03	229	4.7508e+03	144	8.3748e+03

的精度. 与QBFO和QGA算法相比, 对于某些基准函数, 虽然两者在平均迭代次数上优于所提出算法, 但是对应的最优值远大于NAQBFO对应的最优值. 与SPSO算法相比, 虽然NAQBFO的收敛速度相对较慢, 但是具有较好的收敛精度. 导致这一现象的原因是: SPSO算法易陷入局部最优值, 导致其过快地收敛. 无论是对于 $f_1 \sim f_4$ 的多峰函数, 还是 $f_5 \sim f_9$ 的单峰函数, NAQBFO算法都能以相对较小的迭代次数快速地收敛, 性能优于其他4种相关算法.

表4只是在低维情况下对算法进行的测试, 为了进一步检验算法的性能, 取 $D = 30$, 其余算法的参数设置与 $D = 2$ 时相同, 由于 f_6 和 f_7 函数都是二维函数, 只利用剩下的7个基准函数对所提出算法进行高维情形下的测试, 测试结果如表5所示.

由表5可见: 在高维情况下, 与LAQBFO、QBFO和QGA相比, 对于多峰函数 $f_1 \sim f_3$ 和单峰函数 f_8 、 f_9 , NAQBFO算法都能以相对较小的迭代次数收敛; 对于函数 f_4 和 f_5 , 虽然NAQBFO算法的平均迭代次数相对其他3种算法未取到最小值, 但是具有最小的收敛精度. 与SPSO算法相比, 与低维下的情况类似, 所提出算法能够取得较好的收敛精度.

5 NAQBFO算法在伺服系统中的应用

为了验证NAQBFO算法的有效性, 本文引入带有负反馈的位置伺服系统, 对位置环分数阶PID控制器的参数进行整定, 如图2所示. $\theta_m^*(s)$ 为位置环输入信号, $\theta_m(s)$ 为位置环输出信号, $G_1(s)$ 为位置环的分数阶PID控制器, $G_2(s)$ 为速度环的整数阶PID控制

器, K_{AP} 为功率放大器的增益, y 为速度环输入, x 为速度环输出, θ' 为扰动, G_{com} 为补偿项, T_Y 为伺服电机响应时间, α 为速度反馈系数, T_{on} 为测速发电机的输出滤波器时间常数. 速度环的传递函数为

$$G(s) = K_{p2} + \frac{K_{i2}}{s} + \frac{K_{d2}s}{1 + T_m s}. \quad (9)$$

其中: K_{p2} 、 K_{i2} 、 K_{d2} 分别为整数阶PID控制器的比例增益、积分增益和微分增益, s 为拉普拉斯因子, T_m 为微分滤波器的时间常数. 整个速度环的传递函数为

$$H_s(s) = \frac{x(s)}{y(s)} = \frac{G_2 K_{AP} (T_{on} s + 1)}{T_Y T_{on} s^2 + (T_Y + T_{on}) s + G \alpha K_{AP} + 1} = \frac{a_3 s^3 + a_2 s^2 + a_1 s + a_0}{b_4 s^4 + b_3 s^3 + b_2 s^2 + b_1 s + b_0}. \quad (10)$$

其中

$$\begin{aligned} a_0 &= K_{AP} K_{i2}, \\ a_1 &= K_{AP} K_{p2} + K_{AP} K_{i2} T_m + K_{AP} T_{on} K_{i2}, \\ a_2 &= T_{on} K_{AP} K_{p2} + T_{on} K_{AP} K_{i2} T_m + K_{AP} K_{p2} T_m + K_{AP} K_{d2}, \\ a_3 &= T_{on} K_{AP} K_{p2} T_m + T_{on} K_{AP} K_{d2}, \\ b_0 &= \alpha K_{AP} K_{i2}, \\ b_1 &= \alpha K_{AP} K_{p2} + \alpha K_{AP} K_{i2} K_m + 1, \\ b_2 &= T_{on} + T_m + T_Y + \alpha K_{AP} K_{p2} T_m + \alpha K_{AP} K_{d2}, \\ b_3 &= T_{on} T_m + T_{on} T_Y + T_m T_Y, \\ b_4 &= T_m T_Y T_{on}. \end{aligned}$$

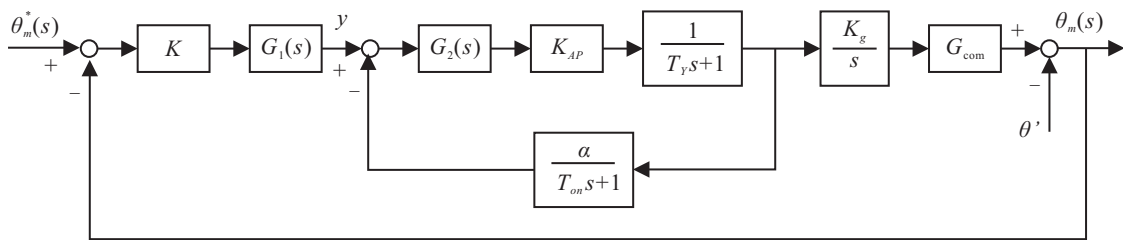


图2 位置伺服系统结构

5.1 NAQPSO-PID控制器设计

与整数阶PID控制器相比, 分数阶控制器对自身和被控对象的参数变化均不敏感, 具有较强的鲁棒性和干扰能力, 因此引入分数阶PID控制器来控制伺服系统的位置环. 分数阶PID控制器^[16]的传递函数为

$$G(s) = K_p + \frac{K_i}{s^\lambda} + K_d s^\mu. \quad (11)$$

与整数阶PID控制器相比, 分数阶控制器多了积分阶次 λ 和微分阶次 μ 两个参数. 实际上, 分数阶控

制器是整数阶控制器的广义表达式, 选择不同的 λ 和 μ , 可以构成不同结构的控制器, 使得分数阶控制器具有更好的灵活性.

本文采用的性能评价函数为

$$f = w_1 \sigma + w_2 t_r + w_3 t_s + w_4 e_{ss}. \quad (12)$$

其中: σ 为系统超调; t_r 为上升时间; t_s 为调节时间; e_{ss} 为稳态误差; w_i 为权重因子, 且满足 $\sum_{i=1}^4 w_i = 1$.

考虑到等速和正弦输出信号中不包含超调、调

节时间等时域指标,测试等速和正弦信号时采用经典的时间乘以误差绝对值积分(ITAE)的性能指标,如下所示:

$$J_{ITAE} = \int_0^{t_f} t|e(t)|dt. \quad (13)$$

5.2 实验和结果

为了测试NAQBFO-PID控制器的性能,引入一个双环伺服系统对分数阶控制器的PID控制器参数

进行整定,参数设置如表6所示. P 、 I 、 D 、 λ 和 μ 的取值范围设置为 $[0,2]$. 仿真时间为5s,最大迭代次数为500,采用3个典型的位置伺服测试信号: $\theta_m^*(t) = 70$ 、 $\theta_m^*(t) = 30t$ 、 $\theta_m^*(t) = 3\sin(2\pi t)$. 测试步骤:首先,对阶跃信号进行参数整定,获得5个最优参数 P 、 I 、 D 、 λ 和 μ ;然后,将5个参数用于等速信号,利用算法进行单参数整定,获得前馈系数 K_{fe} ;最后,将获得的6个参数用于正弦信号的参数整定.

表6 伺服系统参数设置

调制指数 K_{p2}	K_{i2}	K_{d2}	K	α	K_{p2}	T_{on}	K_{AP}	T_Y	K_g	θ'
0.8	10	0	480	19.1	0.08	23/440	0.08	1	$\sin(0.1t)$	$\frac{(1/25)^2 s^2 + 2 * (1/25) * 0.06 + 1}{(1/40)^2 s^2 + 2 * (1/40) * 0.09 + 1}$

经过30次测试后整定结果的平均值如表7所示,对应的阶跃响应曲线、等速误差曲线和正弦误差曲线如图3~图5所示,阶跃响应对应的性能指标如表8所示.

表7 参数整定结果

Algorithm	NAQBFO	LAQBFO	QBFO	QGA	SPSO
K_p	0.2559	0.2152	0.1812	0.1405	0.2355
K_i	0.0014	0.0021	0.0013	0.0015	0.0016
K_d	0.1011	0.0984	0.0734	0.0515	0.0702
K_{fe}	19.0220	18.9931	18.9882	19.1047	19.0214
λ	0.0002	0.0001	0.0001	0.0004	0.0003
μ	0.8262	0.6741	0.7789	0.6810	0.7011

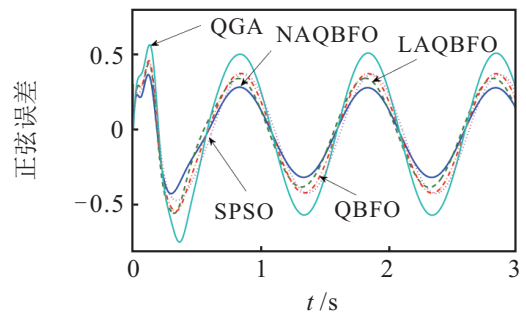


图5 幅值为3°周期为1s的正弦信号误差曲线

表8 阶跃响应的性能指标

性能指标	算法				
	NAQBFO	LAQBFO	QBFO	QGA	SPSO
上升时间/s	0.2559	0.2152	0.1812	0.1405	0.2355
稳态时间/s	0.7980	0.8400	1.0020	1.1230	1.0015
超调/%	0	1.56	0	0	0
稳态误差	4.32e-02	1.16e-01	8.98e-02	1.46e-01	8.56e-01

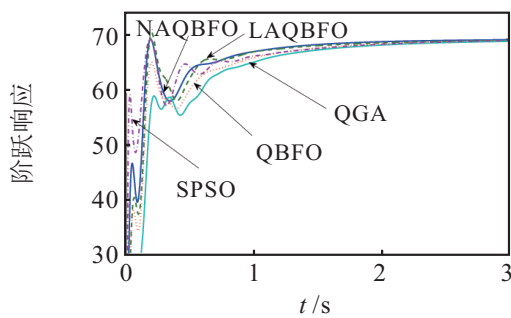


图3 70°阶跃响应曲线

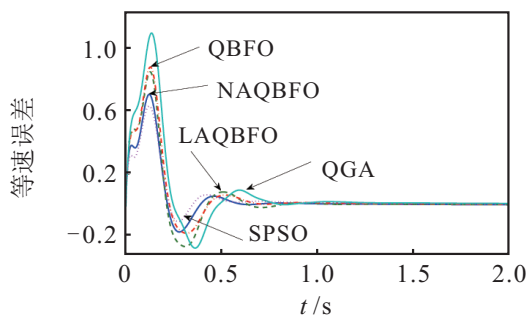


图4 30°/s等速信号误差曲线

由图3~图5和表8可见,对于两环位置伺服系统的3个典型测试信号,与其他3种量子算法相比,NAQBFO算法在取得较快收敛速度的同时,还具有较小的收敛精度,对位置伺服系统的分数阶PID控制器能够进行有效的参数整定.同时,也能够反映出与SPSO算法的特点:前期收敛速度较快,后期易陷入局部最优值,影响算法最终的收敛精度和稳态时间.

6 结论

本文提出了一种新的非线性自适应旋转角的量子菌群算法,采用非线性变化,可以在迭代初期进行缓慢的搜索,更大范围搜索解空间,在后期进行快速地收敛,加快寻优速度.对非线性情形下的参数(如 θ_{max} 、 θ_{min} 和 m)进行了合理的推导,为用户采用繁琐的试错法进行参数选取节省时间.通过几个基准测

试函数对所提出算法进行性能测试,并与LAQBFO、QBFO、QGA和SPSO四种算法进行比较.测试结果表明,与其他4种算法相比,NAQBFO算法能以较小的收敛精度进行快速的收敛.分数阶伺服系统的PID控制器参数整定结果表明,所提出算法能有效地对PID控制器参数进行整定.

为了进一步验证所提出算法在实际系统中参数整定的效果,今后的工作中,将在以工控机为控制器的实际系统中对算法进行验证.

参考文献(References)

- [1] Huang S, Zhao G L. A comparison between quantum inspired bacterial foraging algorithm and GA-like algorithm for global optimization[J]. *Int J of Computational Intelligence & Applications*, 2012, 11(3): 1969-2013.
- [2] Zhang G Y. Bacterial foraging optimization algorithm with quantum behavior[J]. *J of Electronics & Information Technology*, 2013, 35(3): 614-621.
- [3] Narayanan A, Moore M. Quantum-inspired genetic algorithms[C]. *Proc of IEEE Int Conf on Evolutionary Computation*. Nagoya: IEEE, 1996: 61-66.
- [4] Li F, Zhang Y T, Wu J L, et al. Quantum bacterial foraging optimization algorithm[C]. *Proc of 2014 IEEE Congress on Evolutionary Computation*. Beijing: IEEE, 2014: 1265-1272.
- [5] Cao J L, Gao H Y. A Quantum-inspired bacterial swarming optimization algorithm for discrete optimization problems[C]. *Proc of Int Conf on Advances in Swarm Intelligence*. Springer, 2012: 29-36.
- [6] Diamantopoulos A, Lereun C, Rasul F, et al. A quantum bacterial foraging optimization algorithm and its application in spectrum sensing[J]. *Int J of Modelling Identification & Control*, 2013, 18(3): 234-242.
- [7] Gao H Y, Li C W. Quantum-inspired bacterial foraging algorithm for parameter adjustment in green cognitive radio[J]. *J of Systems Engineering & Electronics*, 2015, 26(5): 897-907.
- [8] Li L, Mai X F. Bacterial foraging algorithm based on quantum-behaved particle swarm optimization for global optimization[J]. *Advanced Materials Research*, 2013, 655/657: 948-954.
- [9] Mai X F, Li L. Bacterial foraging algorithm based on quantum-behaved particle swarm optimization and opposition-based learning[J]. *J of Computational Information Systems*, 2013, 9(3): 1157-1165.
- [10] 吴九龙, 李飞, 郑宝玉. 自适应相位旋转的量子菌群算法[J]. *信号处理*, 2015, 31(8): 901-911.
(Wu J L, Li F, Zheng B Y. Quantum bacterial foraging algorithm with adaptive phase rotation[J]. *J of Signal Processing*, 2015, 31(8): 901-911.)
- [11] Kennedy J, Eberhart R. Particle swarm optimization[C]. *Proc of Int Conf on Neural Networks*. Perth: IEEE, 1995: 1942-1948.
- [12] 靳利霞, 唐焕文, 李斌, 等. 一类连续函数模拟退火算法及其收敛性分析[J]. *计算数学*, 2005, 27(1): 19-30.
(Jin L X, Tang H W, Li B, et al. A simulated annealing algorithm for continuous functions and its convergence properties[J]. *Mathematica Numerica Sinica*, 2005, 27(1): 19-30.)
- [13] 李盼池. 基于量子位Bloch坐标的量子遗传算法及其应用[J]. *控制理论与应用*, 2008, 25(6): 985-989.
(Li P C. Quantum genetic algorithm based on bloch coordinates of qubits and its application[J]. *Control Theory & Applications*, 2008, 25(6): 985-989.)
- [14] 张修文, 梁怡. 遗传算法的数学基础[M]. 西安: 西安交通大学出版社, 2000: 110-112.
(Zhang X W, Liang Y. The mathematical basis of genetic algorithm[M]. Xi'an: Xi'an Jiaotong University Press, 2000: 110-112.)
- [15] 覃朝勇, 郑建国, 朱佳俊. 一种实数编码量子进化算法及其收敛性[J]. *控制与决策*, 2009, 24(6): 854-858.
(Qin C Y, Zheng J G, Zhu J J. Real-coded quantum-inspired evolutionary algorithm and its convergence[J]. *Control and Decision*, 2009, 24(6): 854-858.)
- [16] Podlubny I. Fractional-order systems and controllers[J]. *IEEE Trans on Automatic Control*, 1999, 44(1): 208-214.

(责任编辑: 郑晓蕾)