

基于随机评价机制的交互式双子群 QPSO 算法

吴涛¹, 严余松¹, 陈曦²

(1. 西南交通大学信息科学与技术学院, 成都 610031; 2. 西南民族大学计算机科学与技术学院, 成都 610041)

摘要: 通过两组势阱中心不同且相互协同的主、辅子群, 在具有量子行为的粒子群优化(QPSO)算法基础上构造一种基于随机评价机制的交互式双子群 QPSO 算法(DIR-QPSO). 该算法通过子群间的协作避免了种群多样性的快速消失, 增强了算法的全局搜索能力. 同时, 随机因子的加入进一步提高了粒子摆脱局部极值的能力. 对6个测试函数的实验结果表明, DIR-QPSO 算法相对于传统的粒子群优化算法(PSO)在处理单峰和多峰函数时具有更好的优化性能, 收敛速度和收敛精度都得到了较大的提高.

关键词: 粒子群优化; 量子行为粒子群优化; 子群; 交互

中图分类号: TP301.6

文献标志码: A

Improved dual-group interaction QPSO algorithm based on random evaluation

WU Tao¹, YAN Yu-song¹, CHEN Xi²

(1. School of Information Science & Technology, Southwest Jiaotong University, Chengdu 610031, China; 2. School of Computer Science & Technology, Southwest University for Nationalities, Chengdu 610041, China. Correspondent: WU Tao, E-mail: dorawu840720@gmail.com)

Abstract: The dual-group interaction quantum-behaved particle swarm optimization(QPSO) algorithm based on random evaluation(DIR-QPSO) is proposed by constructing the master-slave sub-groups with different potential well centers, which avoids the rapid disappearance of swarm diversity and enhances the global searching ability through collaboration between sub-groups. Meanwhile, the involvement of random factor further improves the particles' ability to escape from local extremums. Experiment results on 6 testing functions show that the DIR-QPSO algorithm outperforms the traditional particle swarm optimization(PSO) algorithm regarding the optimization of unimodal and multimodal functions, with enhancement in both convergence speed and precision.

Keywords: particle swarm optimization; quantum-behaved particle swarm optimization; sub-group; interaction

0 引言

粒子群优化算法(PSO)是一种进化计算方法^[1], 该算法通过模拟鸟群的捕食行为来求解优化问题, 具有概念简明、操作简便、参数少、收敛能力强等特点, 在诸多工程领域得到了成功的应用. 针对 Bergh^[2]所证明的 PSO 算法不能保证以概率 1 收敛于全局最优解的问题, Sun 等^[3]利用群体智慧的聚集性提出了具有量子行为的粒子群优化(QPSO)算法. 理论证明 QPSO 算法是一种全局收敛算法. 目前, QPSO 在多个应用领域以及算法改进方面得到了广泛关注^[4-5].

然而, 与其他全局优化算法一样, QPSO 算法在进化过程中仍然存在早熟收敛的问题. 为了进一步提高 QPSO 的优化性能, 本文提出了一种基于随机评价机制的交互式双子群 QPSO 算法(DIR-QPSO). DIR-QPSO 算法通过势阱中心不同、相互协同进化的两个子群, 充分利用搜索空间内的有效信息, 避免了种群多样性的快速丧失, 防止粒子早熟而陷入局部最优解. 对测试函数的实验结果表明, DIR-QPSO 算法在处理单峰和多峰函数时, 有效地提高了已有 PSO 算法的收敛性能, 不仅收敛速度快而且具有较高的收敛精度.

收稿日期: 2013-09-17; 修回日期: 2014-01-03.

基金项目: 国家自然科学基金项目(61104175); 四川省软科学研究计划项目(2012ZR0022); 四川省科技支撑计划项目(2012GZX0090).

作者简介: 吴涛(1984—), 女, 博士生, 从事进化计算的研究; 严余松(1963—), 男, 教授, 博士生导师, 从事交通运输规划及系统优化等研究.

1 量子粒子群优化算法

QPSO算法根据 Clerc等^[6]对粒子运行轨道的分析结果, 通过在局部吸引点 $p_i = (p_{i,1}, p_{i,2}, \dots, p_{i,N})$ 处建立一个 δ 势阱来对群体中的粒子产生影响, 其 p_i 点位置为

$$p_{i,j}(t) = \frac{c_1 r_{1,i,j} P_{i,j}(t) + c_2 r_{2,i,j} G_{i,j}(t)}{c_1 r_{1,i,j} + c_2 r_{2,i,j}}, \quad 1 \leq j \leq N. \quad (1)$$

其中: $P_{i,j}$ 和 G_j 分别表示 D 维搜索空间中粒子 i 的个体极值和全局极值的第 j 维分量; r_1 和 r_2 为 $[0, 1]$ 之间独立分布的随机数, 称为随机因子; c_1 和 c_2 为加速系数. QPSO 中粒子的位置更新公式为

$$X_{i,j}(t+1) = p_{i,j}(t) \pm \alpha |C_j(t) - X_{i,j}(t)| \ln[1/u_{i,j}(t)]. \quad (2)$$

其中: $u_{i,j}(t) \sim U(0, 1)$, α 称为收缩-扩张系数, 可以采用固定取值和线性减小的控制方式. $L_{i,j}(t)$ 为势阱长度, 可通过下式进行评价:

$$L_{i,j}(t) = 2\alpha |C_j(t) - X_{i,j}(t)|, \quad (3)$$

其中 $C(t)$ 定义为所有粒子个体最好位置的平均, 有

$$C(t) = (C_1(t), C_2(t), \dots, C_n(t)) = \frac{1}{N} \sum_{i=1}^N P_i(t). \quad (4)$$

2 交互式QPSO算法

QPSO算法虽然可以保证以概率1收敛, 但仍面临容易陷入局部最优解的问题. 其最主要的原因是该算法在单点处建立势阱, 种群中的所有粒子都在单势阱的引力作用下逐渐向势阱中心靠拢. 这种信息交流方式虽然速度快但方向单一, 粒子会快速地聚集在一个较小的搜索区域内, 导致种群多样性降低. 为了提高QPSO的优化性能, 本文在已有研究的基础上提出了基于随机评价机制的交互式双子群QPSO算法. 新算法扩展了粒子的搜索范围, 通过子群间的交流和合作增强了粒子间的等待效应^[7], 从而避免了种群的过快聚集, 提高了算法的收敛性能.

2.1 学习及评价策略

交互式双子群QPSO算法的具体思想是: 在随机化一组粒子群之后, 将其分成两个相互独立的子群. 其中一组按照标准QPSO算法的规则在 p_i 点处建立 δ 势阱, 将其称为主子群, 用 S_1 表示; 另外一组子群中的粒子在以 p'_i 点为中心建立的 δ 势阱的引力作用下进行迭代搜索, 称为辅子群, 用 S_2 表示. 若整个种群为 S , 则有 $S_1 \cup S_2 = S$. p'_i 点的位置计算公式为

$$p_{i,j}(t)' = [1 - \varphi_{i,j}(t)]P_{i,j}(t) + \varphi_{i,j}(t)G_j(t), \quad (5)$$

其中 $\varphi_{i,j}(t)$ 为 $[0, 1]$ 范围内的随机数. 在QPSO算法的基本迭代方程中, $L_{i,j}(t)$ 表征了粒子知识搜寻的范围, 在每次迭代之前, $L_{i,j}(t)$ 必须被评价. 本文给出随机

最好位置 $R_j(t)$ 的概念, 在种群最好位置 gbest 与平均最好位置 C 之间作出有效的协调, 提出了更为合理的 $L_{i,j}(t)$ 控制方法. 将使用该评价策略的交互式双子群QPSO算法简称为 DIR-QPSO 算法.

DIR-QPSO算法中, 随机最好位置 $R_j(t)$ 通过下式确定:

$$R_j(t) = r_j(t)\text{gbest} + (1 - r_j(t))C(t), \quad (6)$$

其中 $r_j(t) \sim U(0, 1)$. $L_{i,j}(t)$ 则可通过下式来评价:

$$L_{i,j}(t) = 2\alpha |R_j(t) - X_{i,j}(t)|. \quad (7)$$

因此, DIR-QPSO算法中粒子的进化公式为

$$\begin{cases} X_{i,j}(t+1) = \\ p_{i,j}(t) \pm \alpha |R_j(t) - X_{i,j}(t)| \ln[1/u_{i,j}(t)], & i \in N_{S_1}; \\ X_{i,j}(t+1) = \\ p_{i,j}(t)' \pm \alpha |R_j(t) - X_{i,j}(t)| \ln[1/u_{i,j}(t)], & i \in N_{S_2}. \end{cases} \quad (8)$$

其中: $u_{i,j}(t) \sim U(0, 1)$, α 为收缩-扩张系数.

在 DIR-QPSO 算法中, 主、辅子群通过群体最优个体进行信息交互, 即在每次迭代结束时, 对两个子群 S_1 和 S_2 的最优位置所对应适应值的大小进行比较. S_1 中的群体最优个体值用 gbest_1 表示, S_2 中的群体最优个体值用 gbest_2 表示, 如果 gbest_2 的适应值优于 gbest_1 所对应的适应值, 则将 gbest_2 赋值给 gbest_1 , 如果 gbest_2 不优于 gbest_1 , 则对 S_1 不做任何操作. 以上操作的实质是对整个粒子群最优位置的更新, 帮助 S_1 逃离局部最优, 即使种群中出现超级个体, 也不能诱导群体陷入局部极值点. 由计算公式可知, 两个子群的吸引点相对于整个种群的 pbest 和 gbest 的中心对称. 在收敛过程中, 由于势阱引力的作用, 所有粒子的 pbest 逐渐向 gbest 靠拢, 最终主群 S_1 的吸引点 p_i 和辅群 S_2 的吸引点 $p_{i'}$ 将会合二为一, 所以 DIR-QPSO 依然可以保证算法的收敛性能.

通过主群 S_1 和辅群 S_2 的相互补充和协同进化, DIR-QPSO 在不增加粒子规模、算法参数和计算复杂度, 且保证算法收敛的情况下, 通过降低群体聚集的可能性来维持较高的种群多样性, 提高算法的全局搜索能力.

2.2 算法执行流程

根据以上对算法的设计和定义, DIR-QPSO 的执行过程如下:

Step 1: 设置参数. 包括加速系数 c_1, c_2 , 收缩-扩张因子 α , 种群规模 N , 子群 S_1 的规模 N_{S_1} , 子群 S_2 的规模 N_{S_2} , 最大允许迭代次数 iterMax 或者是适应度的误差精度.

Step 2: 种群初始化. 对于 S_1 , 在求解空间中随机

产生粒子的初始位置 $X_{i,j}(0)$, 并令 $P_{i,j}(0) = X_{i,j}(0)$, 其中 $i \in \{1, 2, \dots, N_{S_1}\}$, $j \in \{1, 2, \dots, D\}$; 对 S_2 , 进行与 S_1 相同的操作, 随机产生粒子的初始位置 $X_{i,j}(0)$, 并令 $P_{i,j}(0) = X_{i,j}(0)$, 其中 $i \in \{N_{S_1} + 1, \dots, N\}$, $j \in \{1, 2, \dots, D\}$.

Step 3: 计算 S_1 和 S_2 中每个粒子的适应值, 假设待求解的优化问题为最小化问题, 并将适应值最小的粒子所在位置赋值给群体最优值, 即 $\min(f(X_i)), i \in \{1, 2, \dots, N\} \rightarrow \text{gbest}$.

Step 4: 根据式(4)和(6)计算种群的随机最好位置 $R_j(t)$, 并用该值评价 S_1 和 S_2 的 $L_{i,j}(t)$.

Step 5: 根据式(8)更新粒子 $i(1 \leq i \leq N)$ 在求解空间中的位置.

Step 6: 按目标函数重新计算粒子 i 的当前位置 $X_i(t)$ 所对应的适应值.

Step 7: 更新主、辅两个子群的所有粒子的个体最好位置, 将 $X_i(t)$ 的适应值与 $P_i(t-1)$ 的适应值进行比较, 如果 $f(X_i(t)) < f(P_i(t-1))$, 则执行 $P_i(t) = X_i(t)$ 操作; 否则, 执行 $P_i(t) = P_i(t-1)$ 操作.

Step 8: 将种群中所有粒子当前的适应值与 $P_g(t-1)$ 的适应值进行比较. 对于 $i \in S_1$, 如果 $f(P_i(t)) < f(P_g(t-1))$, 则保存该粒子的当前适应度值为 S_1 的全局历史最好位置, 记为 $P_{gS_1}(t)$, 否则 $P_g(t-1) \rightarrow P_{gS_1}(t)$; 对于 $i \in S_2$, 如果 $P_i(t)$ 的适应度值优于 $P_g(t-1)$ 的适应值, 则保存该粒子的当前适应度值为 S_2 的全局历史最好位置 $P_{gS_2}(t)$, 否则 $P_g(t-1) \rightarrow P_{gS_2}(t)$.

Step 9: 比较两组子群的全局最优位置 $P_{gS_1}(t)$ 和 $P_{gS_2}(t)$, 如果 $f(P_{gS_1}(t)) > f(P_{gS_2}(t))$, 则将 $P_{gS_2}(t)$ 赋值给整个种群的全局最优位置, 即 $P_{gS_2}(t) \rightarrow P_g(t)$, 否则执行 $P_{gS_2}(t) \rightarrow P_g(t)$ 操作.

Step 10: 判断算法终止条件. 如果达到给定的最大迭代次数或者适应值的误差已经达到设定的误差精度, 则搜索停止, 输出搜索结果; 如果不满足终止条件, 则令 $t = t + 1$, 重复 Step 3 ~ Step 10.

3 实验设计与结果分析

3.1 实验设计

本文使用 Sphere, Rosenbrock, Rastrigin, Ackley Griewank 和 Schwefel 六个标准测试函数对时变加速度系数 PSO (PSO-TVAC) 算法^[8]、全信息 PSO (FIPSO) 算法^[9]、全面学习 PSO (CLPSO) 算法^[10]、QPSO 算法、统一 PSO (UPSO) 算法^[11]和 DIR-QPSO 算法的性能进行对比. 实验中 DIR-QPSO 算法的收缩扩张因子从 1 到 0.5 线性递减, 种群的规模设置为 50, 主子群规模 N_{S_1} 和辅子群规模 N_{S_2} 采用平均分配 $N_{S_1} = N_{S_2}$ 策

略; 各算法最大迭代次数 iterMax 设置为 1000; 每个函数独立运行 30 次, 考察 30 次实验迭代结束时测试函数的均值、方差、最优值和最劣值. 各测试函数的表达式, 自变量的搜索范围, 维数, 初始化范围以及理论极值如下:

1) Sphere 函数

$$f_1(X) = \sum_{i=1}^D x_i^2,$$

$$x_i \in [-100, 100], D = 20, \min f_1(0, 0, \dots, 0) = 0.$$

2) Rosenbrock 函数

$$f_2(X) = [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2],$$

$$x_i \in [-10, 10], D = 20, \min f_2(1, 1, \dots, 1) = 0.$$

3) Rastrigin 函数

$$f_3(X) = \sum_{i=1}^D (x_i^2 - 10 \cos(2\pi x_i) + 10),$$

$$x_i \in [-5.12, 5.12], D = 20, \min f_3(0, 0, \dots, 0) = 0.$$

4) Griewank 函数

$$f_4(X) = \sum_{i=1}^D \frac{x_i^2}{4000} - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1,$$

$$x_i \in [-600, 600], D = 20, \min f_4(0, 0, \dots, 0) = 0.$$

5) Ackley 函数

$$f_5(X) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos(2\pi x_i)\right) + 20 + e,$$

$$x_i \in [-32.786, 32.786], D = 20,$$

$$\min f_5(0, 0, \dots, 0) = 0.$$

6) Schwefel 函数

$$f_7(X) = 418.9829D - \sum_{i=1}^D x_i \sin(x_i^{\frac{1}{2}}),$$

$$x_i \in [-500, 500], D = 20,$$

$$\min f_7(420.98, 420.98, \dots, 420.98) = 0.$$

3.2 实验结果

PSO-TVAC, FIPSO, CLPSO, UPSO, QPSO, DIR-QPSO 优化各函数 30 次所得到适应值的均值、方差、最优值和最劣值如表 1 所示. 图 1 展示了使用种群规模为 50 的 6 种算法求解维度为 20 的单峰函数和多峰函数的收敛曲线.

由表 1 可知, DIR-QPSO 在优化大部分函数时得到了最理想的结果. 各函数求解 30 次获得的最优值中, 使用 DIR-QPSO 算法求解 f_1, f_3, f_5, f_6 的结果明显优于其他几种算法, 而 UPSO 在求解 f_2, f_4 时获得

表 1 各函数优化结果($D = 20, N = 50$)

算法	统计特性	f_1	f_2	f_3	f_4	f_5	f_6
PSO-TVAC	Mean	2.62e-102	3.55e+01	2.59e+01	7.13e-01	3.42	4.86e+03
	Std.	2.433e-203	1.46e+03	5.73e+01	3.53e-01	2.92	2.62e+05
	Best	7.17e-115	8.35	1.51e+01	7.19e-02	1.16	1.21e+03
	Worst	2.11e-101	1.22e+02	3.58e+01	1.88	6.28	1.972e+03
FIPSO	Mean	7.56e-48	1.11e+01	3.08e+01	9.56e-04	3.55e-15	1.31e+03
	Std.	5.28e-95	0.575	194.92	6.50e-05	0	1.13e+05
	Best	2.32e-48	1.02e+01	1.59e+01	2.74e-05	3.55e-15	1.06e+03
	Worst	2.63e-47	1.26e+01	5.93e+01	8.12e-04	3.55e-15	1.90e+03
CLPSO	Mean	2.31e-64	8.07	1.16e+01	1.22e-08	4.97e-15	7.51e+02
	Std.	7.51e-128	13.07	3.47	7.61e-03	3.37e-30	1.35 e+03
	Best	1.40e-65	1.02e+00	9.03	1.91e-09	3.55e-15	7.14e+02
	Worst	7.55e-64	1.39e+01	1.53e+01	8.98e-08	7.11e-15	1.66e+03
UPSO	Mean	5.65e-254	1.73e+00	3.91e+01	3.70e-03	6.22e-15	2.80e+03
	Std.	0	2.62	71.5 e+01	2.46e-05	0	2.0 e+05
	Best	7.02e-258	4.57e-02	2.89e+01	0	6.22e-15	7.14e+03
	Worst	5.51e-253	5.21	5.28e+01	1.23e-02	6.22e-15	3.40e+03
QPSO	Mean	1.387e-287	1.47e+01	9.67	5.66e-03	1.30e-14	2.91e+03
	Std.	1.69e-291	0.81	5.72	6.87e-05	3.49e-29	1.10e+03
	Best	2.39e-295	1.37e+01	6.96	0	6.22e-15	5.63e+02
	Worst	2.54e-269	1.69e+01	1.39e+01	2.22e-02	2.04e-14	3.16e+03
DIR-QPSO	Mean	1.046e-313	1.5672	6.00	4.18e-13	3.55e-15	1.65e+02
	Std.	2.75e-299	2.02	5.52	9.42e-05	0	8.38e+03
	Best	6.83e-318	1.98e-01	3.98	0	3.55e-15	1.34e+02
	Worst	2.66e-288	5.30	1.09e+01	2.95e-12	3.55e-15	1.30e+03

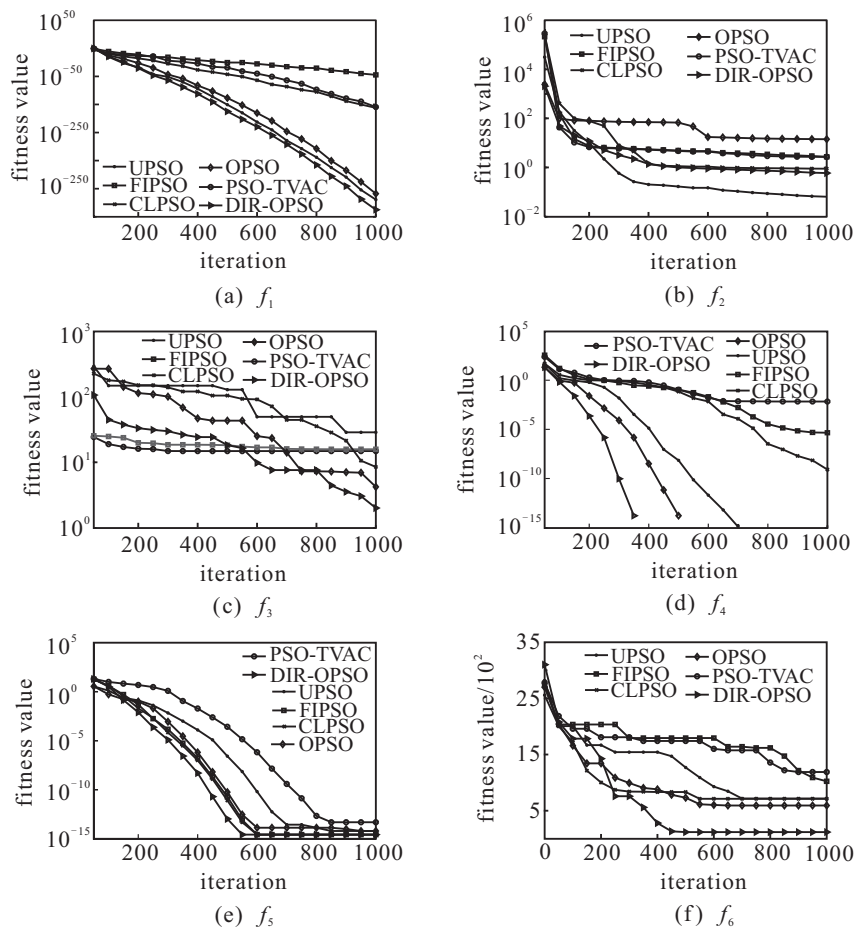


图 1 $D = 20, N = 50$ 时 4 种算法的收敛曲线

的结果最好;从各函数适应值的均值和方差来看,DIR-QPSO算法所获得的均值精度最高,并且求解结果也最稳定.图1(a)~图1(f)分别展示了使用6种算法求解单峰和多峰函数的收敛曲线.其中,DIR-QPSO算法在优化 f_1 时具有最高的寻优精度,UPSO次之,而FIPSO的寻优效果最差;使用PSO-TVAC,FIPSO算法优化 f_2 时迭代到250次左右进入收敛状态,DIR-QPSO算法在450次左右时也陷入收敛,UPSO得到了更好的收敛结果;对于 f_3 ,本文提出的DIR-QPSO算法由于具有更强的随机性表现出了最好的求解性能;在求解 f_4, f_5 时,DIR-QPSO在收敛速度上更有优势;相对其他算法,使用DIR-QPSO算法优化 f_6 在收敛速度和收敛精度上都获得了最理想的结果.

综上所述,DIR-QPSO算法与其他PSO算法相比具有更好的收敛性能.这是因为DIR-QPSO算法通过主群 S_1 和辅群 S_2 不断地交流群体中的最优位置信息,两个子群在求解空间中同时搜索且互相学习的进化模式增加了找到最优解的概率,提高了算法的收敛速度.而随机因子 $R_j(t)$ 的加入使得粒子更加充分地利用群体信息,在学习和创造的过程中不断地发展自身的知识,在控制种群多样性的同时使种群最好位置 g_{best} 和平均最好位置 C 不断地发展,达到最优水平,提高算法的优化性能.

4 结 论

本文提出了一种基于随机评价机制的量子行为粒子群优化算法.在DIR-QPSO算法中,通过主辅子群的相互协同合作,使得粒子间的相互学习更充分,防止了种群多样性减少过快,从而增加了算法的全局搜索能力,提高了寻优性能.在算法中引入的随机最好位置增强了算法的随机性,也对算法的性能改进起到了较大的作用.对单峰函数和多峰函数的实验求解结果表明,DIR-QPSO算法是一个有发展前景的全局优化算法,与传统的PSO算法相比,该算法在处理单峰和多峰函数时具有更高的收敛精度和收敛速度,在对函数求解精度要求较高的相关领域中有着良好的应用前景.

参考文献(References)

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

- Press, 1995: 1942-1948.
- [2] Bergh F V D. An analysis of particle swarm optimizers[D]. Pretoria: Faculty of Natural and Agricultural Science, University of Pretoria, 2001.
- [3] Sun J, Feng B, Xu W B. Particle swarm optimization with particles having quantum behavior[C]. Proc of 2004 Congress on Evolutionary Computation. Piscataway: IEEE Press, 2004, 1: 325-331.
- [4] Liu J, Wu Q, Zhu D. Thruster fault-tolerant for UAVs based on quantum-behaved particle swarm optimization[C]. Opportunities and Challenges for Next-generation Applied Intelligence. Berlin: Springer, 2009: 159-165.
- [5] Fang W, Sun J, Xu W. Analysis of mutation operators on quantum-behaved particle swarm optimization algorithm[J]. New Mathematics and Natural Computation, 2009, 5(2): 487-496.
- [6] Clerc M, Kennedy J. The particle swarm-explosion, stability, and convergence in a multidimensional complex space[J]. IEEE Trans on Evolutionary Computation, 2002, 6(1): 58-73.
- [7] 孙俊, 方伟, 吴小俊, 等. 量子行为粒子群优化: 原理及其应用[M]. 北京: 清华大学出版社, 2011: 49-50. (Sun J, Fang W, Wu X J, et al. Quantum-behaved particle swarm optimization: Theory and application[M]. Beijing: Tsinghua University Press, 2011: 49-50.)
- [8] Ratnaweera A, Halgamuge S, Watson H C, et al. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients[J]. IEEE Trans on Evolutionary Computation, 2004, 8(3): 240-255.
- [9] Mendes R, Kennedy J, Neves J. The fully informed particle swarm: Simpler, maybe better[J]. IEEE Trans on Evolutionary Computation, 2004, 8(3): 204-210.
- [10] Liang J J, Qin A K, Suganthan P N. Comprehensive learning particle swarm optimizer for global optimization of multimodal functions[J]. IEEE Trans on Evolutionary Computation, 2006, 10(3): 281-295.
- [11] Parsopoulos K, Vrahatis M. UPSO: A unified particle swarm optimization scheme[J]. Lecture Series on Computer and Computational Sciences, 2004, 1: 868-873.

(责任编辑: 孙艺红)