

DOI:10.11918/j.issn.0367-6234.201806065

一种高维多模态优化的量子粒子群优化算法

张艺瀛, 金志刚

(天津大学 电气自动化与信息工程学院, 天津 300072)

摘要: 为求解实际工程中的高维多模态优化问题,提出了基于动态邻域的多策略进化的量子粒子群优化算法(QPSO). 针对QPSO算法存在的粒子“早熟”问题,首先定义了一种动态邻域选择机制以保持种群的“活跃性”;然后结合动态邻域机制,定义了三个不同策略的局部吸引子更新方程以保持种群进化的“多样性”. 为了防止算法的进化方向不发散,对收敛到全局最优解的局部吸引子更新策略赋予了较大权重;最后为了拓展最优解空间引入了狼群优化算法中的综合评价方法. 通过对不同类型的高维多模态基准测试函数的仿真实验结果表明:相比于其余四种优化算法,本文提出的优化算法在收敛精度和稳定性方面具有明显优势,并且随着测试维度的增加,这种优势更加凸显,展现出了较好的解决高维多模态优化问题的性能. 文中引入的综合评价方法在所有的测试函数中均具有较高的生效次数,综合评价生效意味着为下一次的进化找到一个更加有利的进化方向,这样能够减少算法找到最优解的次数,也能进一步提升算法的收敛精度.

关键词: 量子粒子群;高维;多模态;动态邻域;局部吸引子

中图分类号: TP18

文献标志码: A

文章编号: 0367-6234(2018)11-0050-10

Quantum particle swarm optimization algorithm for high-dimensional multi-modal optimization

ZHANG Yiyang, JIN Zhigang

(School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China)

Abstract: To solve the problem of high-dimensional multi-modal optimization in practical engineering, a multi-strategy evolutionary quantum particle swarm optimization (QPSO) algorithm based on dynamic neighborhood is proposed. For the "premature" problem of particles in QPSO algorithm, this paper first defines a dynamic neighborhood selection mechanism to maintain the "activity" of the population and then combines the dynamic neighborhood mechanism to define the local attractor update equations of three different strategies to maintain the "diversity" of population evolution. In order to prevent the evolutionary direction of the algorithm from diverging, the local attractor update strategy converging to the global optimal solution is given greater weight. In the end, the comprehensive evaluation method of the wolves optimization algorithm is introduced to expand the optimal solution space. The experimental based on different types of high-dimensional multi-modal benchmark functions show that compared to the other four optimization algorithms, the proposed optimization algorithm has obvious advantages in convergence accuracy and stability, and this advantage becomes more prominent with the increase of testing dimensions, which shows a better performance in solving high-dimensional multi-modal optimization problems. The comprehensive evaluation method introduced in this paper has a high effective number of times in all test functions. The comprehensive evaluation of the effective means to find a more favorable evolution direction for the next evolution and further enhance the accuracy convergence.

Keywords: quantum particle swarm; high dimension; multimodal; dynamic neighborhood; local attractor

群体智能优化算法处理高维多模态问题时面临的瓶颈是高维制约了每一维的可实现性和容易陷入局部最优解^[1]. 粒子群优化(PSO)算法^[2]是一种典型的群体智能优化算法,但不能全局收敛^[3]. QPSO

算法^[4]是将 PSO 算法拓展到量子空间. 相较于 PSO 算法,QPSO 算法具有控制参数选择少,收敛速度快,进化方程形式更为简单和鲁棒性好等优势,并且能够实现全局收敛^[4-5],在很多领域得到了广泛应用^[6-10]. 但是其依然存在着粒子早熟收敛^[11]的缺陷. 为此,很多学者提出了基于 QPSO 的改进算法,如用随机平均最优位置替代原始中的算法平均最优位置的 QPSO 算法^[12],个体间相互协作的 QPSO 算法^[13],基于平均最优位置加权的 QPSO 算法^[14]等.

收稿日期: 2018-06-10

基金项目: 国家自然科学基金(61571318);国家自然科学基金(71502125)

作者简介: 张艺瀛(1993—),男,博士生;

金志刚(1972—),男,教授,博士生导师

通信作者: 金志刚, zgjin@tju.edu.cn

针对 QPSO 算法在解决高维多模态问题存在早熟收敛的问题, 本文提出了一种改进的 QPSO 算法 (IQPSO). IQPSO 算法定义了一种动态邻域选择机制和三种不同策略的局部吸引子更新方程来保持种群的“活跃性”和“多样性”, 并且为拓展最优解空间引入了狼群优化中的综合评价方法.

1 量子行为粒子群优化算法

QPSO 算法可视为 PSO 算法在量子空间的扩展. QPSO 算法的进化方程和 PSO 算法具有根本的差别. PSO 算法依据粒子的飞行速度进行位置的更新, 而 QPSO 算法以 δ 势阱为基础, 通过求解解定谟方程得到粒子在某点出现的概率密度函数, 然后再利用蒙特卡罗反变换获得粒子位置^[1], QPSO 和 PSO 的进化方程如下:

$$m_{\text{best}}^t = \frac{1}{N} \sum_{i=1}^N P_{\text{best}_i^t}, \quad (1)$$

$$P_i^t = \alpha \times P_{\text{best}_i^t} + (1 - \alpha) \times G_{\text{best}}^t, \quad (2)$$

$$X_i^{t+1} = P_i^t \pm \beta |m_{\text{best}}^t - X_i^t| \ln\left(\frac{1}{u}\right), \quad (3)$$

$$\begin{cases} V_i^{t+1} = w * V_i^t + \{c_1 * r_1 * (P_{\text{best}_i^t} - X_i^t)\} + \\ \quad \{c_2 * r_2 * (G_{\text{best}}^t - X_i^t)\}; \\ X_i^{t+1} = V_i^{t+1} + X_i^t. \end{cases} \quad (4)$$

式中: N 为种群个数, t 为迭代次数; m_{best}^t 为种群的最优平均位置, $P_{\text{best}_i^t}$ 为第 i 个粒子的历史最优位置, X_i^t 为第 i 个粒子的当前位置, G_{best}^t 为种群的最优位置, V_i^t 为第 i 个粒子的当前速度; a, u, r_1 和 r_2 均满足 $(0, 1)$ 正态分布, 即 $(\alpha, u, r_1, r_2 \sim U(0, 1))$, β 是收缩膨胀因子, c_1 和 c_2 是学习因子, w 是惯性权重 (方程(1) ~ (3) 是 QPSO 的进化方程, 方程(4) 为 PSO 进化方程).

2 改进的量子行为粒子群优化算法

2.1 自适应动态邻域选择机制

Suganthan 于 1999 年在标准 PSO 算法中引入了空间邻域的概念^[15]; 同年, Kennedy 引入邻域拓扑的概念来调整邻域的动态选择^[16].

Kennedy 提出的邻域结构包括环形结构、随机环形结构、轮形结构以及随机轮形结构, 见图 1. 环形结构是一种最基本的邻域结构, 可看出, 无论环形结构还是随机环形结构, 由于每个粒子仅仅是与两个粒子进行信息交换, 都会导致信息交流非常慢; 而在轮形结构中, 每个粒子则可与结构中除此之外的所有粒子进行信息交换, 这样能够有效的提高了粒子之间的信息传播速度.

因而, 为了提高粒子之间的信息交换效率, 保持种群的“活跃性”, 本文在文献[15-16]关于邻域结构的思想基础上, 结合轮形邻域结构的特性, 提出了一种动态邻域更新机制.

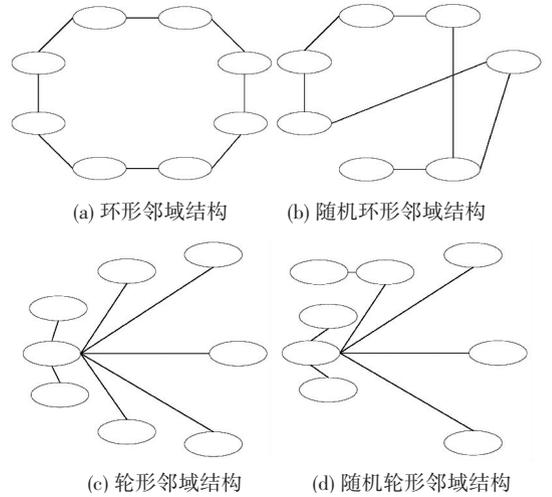


图 1 邻域结构类型

Fig.1 Neighborhood structure types

2.1.1 动态邻域生成算法

本文提出的动态邻域结构生成方式保证每个粒子至少与 1 个粒子进行信息交换, 其具体实现如下:

种群动态邻域生成算法

```

1: 输入: 种群数量  $NX$ , 邻域规模控制参数  $p$ 
2: 输出: 种群的邻域集合 set
3: 过程:
4:   for  $i \in N(i = 1, 2, 3, \dots, N)$  do
5:     flag = 1
6:     for  $j \in N(j = 1, 2, 3, \dots, N)$  do
7:       if  $i \sim j \&\& \text{rand} < p$ 
8:         set  $[i] [j] = 1$ , set  $[j] [i] = 1$ 
9:         flag = 0
10:      end if
11:    end for
12:    if flag == 1
13:       $j = \text{floor}(N \times \text{rand})$ 
14:      while  $i == j$ 
15:         $j = \text{floor}(N \times \text{rand})$ 
16:      end while
17:      set  $[i] [j] = 1$ , set  $[j] [i] = 1$ 
18:    end if
19:  end for

```

上述的动态邻域生成算法保证每个粒子的邻域规模至少为 1 个, 则每个粒子的邻域求解实现如下:

粒子邻域求解算法

```

1: 输入: 当前粒子编号  $i$ , 种群数量  $N$ , 邻域集合 set
2: 输出: 粒子编号  $i$  的邻域粒子编号集合 index
3: 过程:
4:   count = 0
5:   for  $j \in N(i = 1, 2, 3, \dots, N)$  do
6:     if set [ $i$ ] [ $j$ ] == 1
7:       count = count + 1
8:       index [count] =  $j$ 
9:     end if
10:  end for

```

2.1.2 自适应邻域规模控制参数

从种群动态邻域生成算法中可看出, 邻域规模控制参数 p 满足 $p \in [0, 1]$, 其大小和最终生成的动态邻域有密切关联. 记事件 A 为: $\text{rand} < p$; 记事件 B 为: 事件 A 发生的概率变大时, 每个粒子的邻域中的最优粒子是全局最优的粒子的概率变大. 则在多次独立重复实验中, 若 p 变大, 那么事件 A 发生的概率变大, 同时事件 B 会发生, 给出详细证明:

证明 设 $p, p' \in (0, 1)$ 且满足 $p < p'$, 在 N 次独立重复实验中, $\text{rand} < p$ 发生的次数记为 M , $\text{rand} < p'$ 发生的次数记为 M' , 则有 $M \leq M'$, 即若 p 变大, 事件 A 发生的概率变大. 本文提出的邻域生成算法, 可视为概率抽样, 即每个粒子被抽取到的概率是相等的. N 可看作种群数量, M 和 M' 分别看作粒子 $i (i = 1, 2, \dots, N)$ 在 p 和 p' 时对应的邻域中的粒子数量, 事件 B 就转化为: 从样本 N 中分别抽取 M 和 M' 次, 抽到最优粒子概率的大小问题. 根据概率抽样理论, 每个粒子被抽到的概率是相等的, 则抽取 M 和 M' 次, 其抽到最优粒子的概率分别为 $\frac{M}{N}$ 和 $\frac{M'}{N}$, 则有 $\frac{M}{N} \leq \frac{M'}{N}$.

根据上述证明可推导出, 当邻域规模控制参数较大时, 不但会使得每个粒子的邻域中具有最优粒子的概率增加, 也会使得每个粒子的邻域中具有其它“优秀粒子”(比如稍劣于最优粒子的粒子)的概率增加, 这样会使得每个粒子的邻域中的最优粒子相差不大或者相同的概率增加, 这样就与提出的多策略局部吸引子更新机制的初衷是相悖的. 因而, 实验中所选取的 p 值不宜较大, 且 p 应当与 N 成反比, 这样保证即使 N 过大, 也不会导致邻域规模过大.

基于上述分析, 给出了 p 的表达式为

$$p = 1.0 - \left(1.0 - \frac{1.0}{N}\right)^3. \quad (5)$$

p 可根据实际需要调整 p 的大小.

2.2 多策略局部吸引子更新机制

在 QPSO 算法中, 每个粒子的历史最优位置都会随着进化代数的增加而变的更加优秀或者与上一次进化的结果相同. 粒子的历史最优位置之所以持续不断改善, 与局部吸引子的更新方程具有密切关联, 式(2)给出了 QPSO 中的局部吸引子更新方程. G_{best^t} 中汲取“优良基因”, 从而使自身变的“优秀”, 但是同时也会在一定程度上破坏粒子自身的“属性”, 从而导致整个种群的多样性出现“萎缩”, 进而会使得种群粒子过早的“成熟”. 基于此, 结合文中提出的自适应动态领域选择机制, 提出一种多策略的局部吸引子更新机制. 设粒子 i 的邻域中的最优粒子位置记为 LP_{best^t} , 则本文提出的多策略局部吸引子更新方程如下:

$$P_i^t = \alpha \times P_{\text{best}^t} + (1 - \alpha) \times G_{\text{best}^t}, \quad (6)$$

$$P_i^t = \alpha \times P_{\text{best}^t} + (1 - \alpha) \times LP_{\text{best}^t}, \quad (7)$$

$$P_i^t = \alpha \times P_{\text{best}^t} + (1 - \alpha) \times (G_{\text{best}^t} - LP_{\text{best}^t}). \quad (8)$$

式(6)是 QPSO 算法中的原始局部吸引子更新方程, 表示从种群最优位置中吸收“优秀基因”; 式(7)将全局最优位置替换为了邻域最优位置, 表示从邻域最优位置中吸收“优秀基因”; 式(8)将全局最优位置替换为了全局最优位置与邻域最优位置的差, 表示从两者的差中吸收“优秀基因”. 可看出本文提出的多策略局部吸引子更新方程丰富了粒子进化的方向, 能够有效的抑制粒子“早熟”. 提出的多策略局部吸引子更新机制的具体实现如下:

多策略局部吸引子更新算法

```

1: 输入: 种群数量  $N$ , 全局最优粒子  $G_{\text{best}^t}$ , 个体历史最优粒子  $P_{\text{best}^t} (i = 1, 2, 3, \dots, N)$ , 粒子邻域最优粒子  $LP_{\text{best}^t} (i = 1, 2, 3, \dots, N)$ , 权重  $l$ 
2: 输出: 粒子的局部吸引子  $P_i^t (i = 1, 2, 3, \dots, N)$ 
3: 过程:
4: for  $i \in N(i = 1, 2, 3, \dots, N)$  do
5:    $a = \text{rand}$ 
6:    $w = \text{rand}$ 
7:   if  $w < l$ 
8:      $P_i^t = \alpha \times P_{\text{best}^t} + (1 - \alpha) \times LP_{\text{best}^t}$ 
9:   else if  $l < w < 1 - l$ 
10:     $P_i^t = \alpha \times P_{\text{best}^t} + (1 - \alpha) \times G_{\text{best}^t}$ 
11:   else
12:     $P_i^t = \alpha \times P_{\text{best}^t} + (1 - \alpha) \times (G_{\text{best}^t} - LP_{\text{best}^t})$ 
13:   end if
14: end for

```

虽然算法中提出了 3 种局部吸引子更新策略,

但是从种群最优位置中吸收“优秀基因”依然是最有利于算法收敛到最优解的,因而应当赋予较大的比重,入口参数权重 l 需满足 $l \in (0, 1/3)$. 如果 l 过小则无法充分体现所提出的多策略局部吸引子更新机制的有效性,因而在实验中取 $l = 0.2$.

2.3 狼群优化的综合评价方法

在灰狼优化算法中,为了进行模拟灰狼的狩猎行为,确定最佳的猎物潜在位置,会把 3 个最优灰狼的位置进行平均^[17]. 本文借鉴灰狼优化算法的综合评价方法,在每次进化之后,按照适应度值的优劣对种群中的所有粒子进行排序,越优的粒子排的越靠前,排序之后的粒子个体历史最优位置的顺序记为 $\hat{P}_{best_i^t}$ ($i = 1, 2, \dots, N$), 然后挑选出最优的 3 个粒子位置进行粒子位置平均,得到新生成位置 \overline{m}_{best^t} 为

$$\overline{m}_{best^t} = \frac{1}{3} \sum_{i=1}^3 \hat{P}_{best_i^t}. \quad (9)$$

将新生成的位置 \overline{m}_{best^t} 与最优的全局位置 G_{best^t} 进行比较,若优于最优的全局位置,则用新生成的位置替代全局最优位置.

2.4 改进算法的实现流程

本文提出的改进的量子粒子群优化算法实现流程如下:

步骤 1 初始化种群大小 N 、粒子位置 X_i^0 、最大迭代次数 $MAXIT$ 、粒子最优位置 $P_{best_0} = X_i^0$; 依据式 (5) 求邻域规模控制参数 $prob$, 执行种群动态邻域生成算法,初始化邻域结构.

步骤 2 对粒子个体历史最优适应度值升序排列(假设求解最小值),然后依据式 (10) 更新个体最好位置的平均位置 m_{best^t} 为

$$m_{best^t} = \frac{1}{N} \sum_{i=1}^N \hat{P}_{best_i^t}. \quad (10)$$

步骤 3 由式 (11) 获取整个种中的历史最优位置 G_{best^t} , 根据式 (9) 求解 \overline{m}_{best^t} , 若 \overline{m}_{best^t} 优于 G_{best^t} , 则 $G_{best^t} = \overline{m}_{best^t}$.

$$G_{best^t} = \hat{P}_{best_1^t}. \quad (11)$$

步骤 4 对种群中的每一个粒子 i ($1 \leq i \leq N$) 执行步骤 5~7.

步骤 5 执行粒子邻域求解算法,依据适应度值的优劣找出每个粒子邻域中的 $lP_{best_i^t}$; 执行多策略局部吸引子更新算法得到局部吸引子 P_i^t .

步骤 6 由式 (12) 更新粒子的进化方程为

$$X_i^{t+1} = P_i^t \pm \beta |m_{best^t} - X_i^t| \ln\left(\frac{1}{u}\right). \quad (12)$$

其中, β 和 u 值的选取,请参见文献[14].

步骤 7 由式 (13) 获取个体粒子历史最好位置为

$$P_{best_i^{t+1}} = \begin{cases} X_i^t, & f[X_i^t] < f[P_{best_i^t}], \\ P_{best_i^t}, & f[X_i^t] \geq f[P_{best_i^t}]. \end{cases} \quad (13)$$

步骤 8 执行步骤 3, 得全局最优位置 $G_{best^{t+1}}$.

步骤 9 重复步骤 4~8, 直到迭代次数达到最大,最终得到的 $G_{best^{t+1}}$ 即为所提出算法得到的最优位置, $f(G_{best^{t+1}})$ 即为所求得的最优解.

3 实验结果及分析

3.1 测试函数

为测试本文提出算法的寻优性能,实验选取 12 个经典的多峰基准测试函数(测试函数来自于西蒙弗雷泽大学仿真实验函数集, <http://www.sfu.ca/~ssurjano/index.html>), 表 1 中给出了每个测试函数的具体数学表达式,搜索区间的上下限及其所对应的最优解.

3.2 参数设置

本文选取了 5 个进化算法进行性能比较,即差分进化(DE)算法、PSO 算法、QPSO 算法、WQPSO 算法和本文提出的 IQPSO 算法,各算法的具体参数设置如下:

DE 算法:种群数量 40,交叉率 0.8,变异控制参数的上限为 0.8,变异控制参数的下限为 0.2,最大迭代次数 1 000,重复实验次数 100.

PSO 算法:种群数量 40,惯性权重 1,惯性权重阻尼比 0.99,个体学习系数 2.05,全局学习系数 2.05,最大迭代次数 1 000,重复实验次数 100.

QPSO 算法:种群数量 40,最大迭代次数 1 000,重复实验次数 100.

WQPSO 算法:种群数量 40,权重设置 1.5 到 0.5 线性下降,最大迭代次数 1 000,重复实验次数 100.

IQPSO 算法:种群数量 40,最大迭代次数 1 000,重复实验次数 100.

以上算法的仿真硬件设备为 Intel core i5 1.8 GHz 8GB RAM 笔记本;仿真软件环境 MATLAB R2017a. 为更好的体现算法性能在横向和纵向的对比性,本文实验选取两个测试维度分别为 10 和 40.

3.3 评价指标

为了比较不同算法的性能着重考虑以下几个指标:

寻优精度:用 100 次独立重复实验得到的最优解的平均值作为衡量的依据.

收敛速度:用同一个测试函数,不同算法在相同条件下搜索到相同最优解所用的评价次数来衡量,评价次数越少,表明收敛性能越好;反之,越差.

表 1 实验采用的基准测试函数的最优解、搜索范围和初始化范围

Tab.1 The optimal solution, search scope and initialization scope of benchmark functions in the experiment

测试函数	函数表达式	搜索空间	最优解
Sphere	$f_1 = \sum_{i=1}^D x_i^2$	$[-100, 100]$	0
Rastrigin	$f_2 = \sum_{i=1}^D x_i^2 - 10\cos(2\pi x_i) + 10$	$[-5.12, 5.12]$	0
Rosenbrock	$f_3 = \sum_{i=1}^{D-1} [100 \times (x_i^2 - x_{i+1})^2 + (x_i - 1)^2]$	$[-5, 10]$	0
Ackley	$f_4 = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^D \cos 2\pi x_i\right) + 20 + e$	$[-30, 30]$	0
Griewank	$f_5 = \frac{1}{4000}\sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos \frac{x_i}{\sqrt{i}} + 1$	$[-600, 600]$	0
Zakharov	$f_6 = \sum_{i=1}^D x_i^2 + \left(\sum_{i=1}^D 0.5 \times i \times x_i\right)^2 + \left(\sum_{i=1}^D 0.5 \times i \times x_i\right)^4$	$[-5, 10]$	0
Schwefel Problem 2.2	$f_7 = \sum_{i=1}^D x(i) + \prod_{i=1}^D x(i) $	$[-100, 100]$	0
Sum Different Powers	$f_8 = \sum_{i=1}^D (x_i)^{i+1}$	$[-1, 1]$	0
Sum Squares	$f_9 = \sum_{i=1}^D i \times [x(i)]^2$	$[-10, 10]$	0
Rotated Hyper-Ellipsoid	$f_{10} = \sum_{i=1}^D \left(\sum_{j=1}^i x_j^2\right)$	$[-65.536, 65.536]$	0
Rast	$f_{11} = \sum_{i=1}^D (x_j^2 - 10 \times \cos(2 \times \pi \times x_j))$	$[-5.12, 5.12]$	0
Schwefel Problem 1.2	$f_{12} = \sum_{i=1}^D \left(\sum_{j=1}^i x_j\right)^2$	$[-100, 100]$	0

稳定性:用标准差来衡量,标准差越小,表明算法的稳定性越好;标准差越大,表明算法的稳定性越差.

有效性:用来衡量 IQPSO 算法引入狼群优化综合评价方法有效性的指标,本文用该方法生效的次數来衡量.

3.4 实验结果分析

表 2、3 分别给出了测试维度在 10 和 40 时的实验仿真结果,包括不同算法在不同测试函数的最优解,平均解,最差解和均方差.

3.4.1 寻优精度

表 4 中给出了 5 种算法在 12 个基准测试函数中收敛优势的统计表现. 可看出本文的提出的 IQPSO 算法在维度为 10 时,最优解和平均最优解中

的统计结果分别为 11 和 11;在维度为 40 时,最优解和平均最优解中的统计结果分别为 11 和 12. 可看出,相较于其余 4 种算法, IQPSO 算法在寻优精度方面具有明显的优势,并且随着维度的增加,这种优势进一步凸显. 展现了 IQPSO 算法求解高维优化问题的优越性能.

3.4.2 稳定性

表 4 中给出了 5 种算法在 12 个基准测试函数中均方差方面的统计表现,可看出 IQPSO 算法在维度为 10 和维度为 40 时的最优均方差的个数分别是 8 个和 12 个,也就是说相较于其余 4 种算法,本文提出的算法展现出极强的稳定性,并且这种表现随着维度增加更加凸显,展现出提出的 IQPSO 算法在求解高维优化问题时的较高稳定性.

表 2 仿真实验结果(维度 10)

Tab.2 Simulation experiment results (dimension 10)

测试函数	算法	最优解	平均解	最差解	均方差
Sphere	DE	1.09×10^{-49}	3.47×10^{-46}	9.61×10^{-45}	1.08×10^{-45}
	PSO	1.07×10^{-89}	1.74×10^{-73}	1.26×10^{-71}	1.30×10^{-72}
	QPSO	3.24×10^{-90}	8.72×10^{-76}	4.83×10^{-74}	5.78×10^{-75}
	WQPSO	4.59×10^{-95}	7.00×10^{-80}	6.66×10^{-78}	6.67×10^{-79}
	IQPSO	9.34×10^{-104}	3.68×10^{-97}	3.29×10^{-95}	3.29×10^{-96}
Rastrigin	DE	0	1.04×10^1	2.42×10^1	5.18×10^0
	PSO	2.00×10^0	1.04×10^1	2.49×10^1	4.54×10^0
	QPSO	0	2.51×10^0	8.38×10^0	1.59×10^0
	WQPSO	0	2.16×10^0	6.08×10^0	1.37×10^0
	IQPSO	0	8.27×10^{-1}	1.70×10^1	2.75×10^0
Rosenbrock	DE	3.46×10^{-1}	2.65×10^0	4.24×10^0	9.25×10^{-1}
	PSO	1.22×10^{-2}	7.32×10^0	1.66×10^2	1.90×10^1
	QPSO	6.04×10^{-4}	5.02×10^0	8.26×10^1	8.28×10^0
	WQPSO	1.84×10^{-2}	6.33×10^0	1.04×10^{-2}	1.63×10^{-1}
	IQPSO	4.44×10^0	4.74×10^0	5.16×10^0	1.39×10^{-1}
Ackley	DE	4.44×10^{-15}	4.44×10^{-15}	4.44×10^{-15}	0
	PSO	4.44×10^{-15}	5.08×10^{-15}	7.99×10^{-15}	1.37×10^{-15}
	QPSO	8.88×10^{-16}	4.41×10^{-15}	4.44×10^{-15}	3.55×10^{-16}
	WQPSO	8.88×10^{-16}	4.41×10^{-15}	4.44×10^{-15}	3.55×10^{-16}
	IQPSO	8.88×10^{-16}	3.73×10^{-15}	4.44×10^{-15}	1.43×10^{-15}
Griewank	DE	0	8.38×10^{-2}	3.13×10^{-1}	8.97×10^{-2}
	PSO	1.48×10^{-2}	9.13×10^{-2}	2.32×10^{-1}	4.04×10^{-2}
	QPSO	6.85×10^{-4}	4.92×10^{-2}	3.32×10^{-1}	4.00×10^{-2}
	WQPSO	7.40×10^{-2}	4.80×10^{-2}	3.36×10^{-1}	3.89×10^{-2}
	IQPSO	0	3.65×10^{-2}	3.28×10^{-1}	6.06×10^{-2}
Zakharov	DE	1.40×10^{-20}	3.76×10^{-18}	8.02×10^{-17}	1.13×10^{-17}
	PSO	2.42×10^{-16}	1.30×10^{-11}	2.38×10^{-10}	3.54×10^{-11}
	QPSO	1.56×10^{-31}	2.24×10^{-21}	2.09×10^{-19}	2.09×10^{-20}
	WQPSO	4.65×10^{-30}	4.57×10^{-24}	3.40×10^{-22}	3.46×10^{-23}
	IQPSO	5.42×10^{-53}	8.46×10^{-46}	8.33×10^{-44}	8.33×10^{-45}
Schwefel problem 2.2	DE	1.33×10^{-24}	2.67×10^{-23}	2.51×10^{-22}	3.61×10^{-23}
	PSO	1.05×10^{-55}	4.19×10^{-40}	4.18×10^{-38}	4.18×10^{-39}
	QPSO	8.47×10^{-49}	1.40×10^{-37}	4.96×10^{-36}	7.47×10^{-37}
	WQPSO	7.54×10^{-53}	1.29×10^{-44}	7.62×10^{-43}	7.87×10^{-44}
	IQPSO	1.00×10^{-53}	1.46×10^{-49}	4.43×10^{-48}	6.08×10^{-49}
Sum different powers	DE	1.02×10^{-103}	1.74×10^{-95}	1.01×10^{-93}	1.09×10^{-94}
	PSO	4.96×10^{-131}	1.54×10^{-95}	1.03×10^{-93}	1.14×10^{-94}
	QPSO	2.84×10^{-130}	7.96×10^{-105}	7.59×10^{-103}	7.59×10^{-104}
	WQPSO	3.51×10^{-142}	1.61×10^{-121}	1.30×10^{-119}	1.31×10^{-120}
	IQPSO	6.16×10^{-179}	1.67×10^{-162}	1.67×10^{-160}	1.68×10^{-161}
Sum Squares	DE	4.13×10^{-50}	1.04×10^{-47}	2.21×10^{-46}	2.72×10^{-47}
	PSO	4.93×10^{-92}	2.44×10^{-73}	1.59×10^{-71}	1.77×10^{-72}
	QPSO	5.62×10^{-91}	2.30×10^{-76}	2.11×10^{-74}	2.11×10^{-75}
	WQPSO	3.96×10^{-96}	9.26×10^{-83}	8.63×10^{-81}	8.63×10^{-82}
	IQPSO	1.70×10^{-105}	6.51×10^{-99}	3.57×10^{-97}	4.29×10^{-98}
Rotated Hyper-Ellipsoid	DE	3.49×10^{-49}	7.17×10^{-46}	1.80×10^{-44}	2.10×10^{-45}
	PSO	2.12×10^{-90}	4.55×10^{-71}	4.38×10^{-69}	4.38×10^{-70}
	QPSO	4.25×10^{-90}	3.82×10^{-74}	3.09×10^{-74}	3.11×10^{-73}
	WQPSO	4.75×10^{-94}	4.96×10^{-81}	4.66×10^{-79}	4.67×10^{-80}
	IQPSO	1.54×10^{-105}	7.67×10^{-98}	6.07×10^{-96}	6.09×10^{-97}
Rast	DE	7.79×10^{-06}	1.07×10^1	2.39×10^1	5.17×10^0
	PSO	1.99×10^0	9.00×10^0	2.29×10^1	4.28×10^0
	QPSO	0	2.57×10^0	7.96×10^0	1.70×10^0
	WQPSO	0	1.86×10^0	9.20×10^0	1.53×10^0
	IQPSO	0	7.55×10^{-1}	1.44×10^1	2.31×10^0
Schwefel problem 1.2	DE	2.61×10^{-16}	8.01×10^{-14}	1.29×10^{-12}	1.99×10^{-13}
	PSO	5.84×10^{-10}	5.03×10^{-6}	8.70×10^{-5}	1.32×10^{-5}
	QPSO	6.05×10^{-15}	8.09×10^{-10}	7.67×10^{-9}	1.63×10^{-9}
	WQPSO	9.48×10^{-16}	8.31×10^{-11}	2.36×10^{-11}	3.09×10^{-10}
	IQPSO	2.05×10^{-44}	2.16×10^{-37}	1.74×10^{-35}	1.75×10^{-36}

注:表中标注粗体数据为特征值,最优解特征值为最小值;平均解特征值为最小值;最差解特征值为最大值;均方差特征值为最小值。

表 3 仿真实验结果(维度 40)

Tab.3 Simulation experiment results (dimension 40)

测试函数	算法	最优解	平均解	最差解	均方差
Sphere	DE	1.72×10^{-7}	1.51×10^{-6}	5.86×10^{-6}	1.16×10^{-6}
	PSO	1.01×10^{-4}	1.17×10^{-1}	9.86×10^{-1}	1.63×10^{-1}
	QPSO	8.84×10^{-11}	6.09×10^{-8}	3.03×10^{-6}	3.14×10^{-7}
	WQPSO	1.58×10^{-15}	1.58×10^{-13}	2.16×10^{-12}	6.66×10^{-13}
	IQPSO	1.78×10^{-43}	8.02×10^{-41}	1.40×10^{-39}	2.14×10^{-40}
Rastrigin	DE	2.30×10^2	2.69×10^2	3.11×10^2	1.66×10^1
	PSO	2.59×10^1	6.39×10^1	1.13×10^2	1.61×10^1
	QPSO	2.28×10^1	4.33×10^1	8.88×10^1	1.16×10^1
	WQPSO	1.88×10^1	4.03×10^1	7.32×10^1	1.06×10^1
	IQPSO	0	1.31×10^{-1}	8.71×10^0	9.12×10^{-1}
Rosenbrock	DE	3.15×10^1	5.64×10^1	2.02×10^2	3.58×10^1
	PSO	7.77×10^2	2.34×10^2	1.27×10^3	1.55×10^2
	QPSO	2.93×10^1	8.63×10^1	4.97×10^2	8.01×10^1
	WQPSO	2.86×10^1	5.39×10^1	1.66×10^2	3.44×10^1
	IQPSO	3.53×10^1	3.56×10^1	3.60×10^1	1.43×10^{-1}
Ackley	DE	8.19×10^{-5}	2.93×10^{-4}	6.24×10^{-4}	9.82×10^{-5}
	PSO	2.00×10^{-3}	6.71×10^{-1}	2.79×10^0	7.90×10^{-1}
	QPSO	1.90×10^{-6}	3.18×10^{-5}	2.02×10^{-4}	4.16×10^{-6}
	WQPSO	6.17×10^{-9}	5.80×10^{-8}	1.94×10^{-7}	3.58×10^{-8}
	IQPSO	4.44×10^{-15}	4.51×10^{-15}	7.99×10^{-15}	5.00×10^{-16}
Griewank	DE	4.15×10^{-7}	7.77×10^{-4}	1.72×10^{-2}	2.77×10^{-3}
	PSO	2.63×10^{-3}	1.79×10^{-1}	9.44×10^{-1}	2.15×10^{-1}
	QPSO	2.33×10^{-10}	5.44×10^{-3}	3.93×10^{-2}	8.42×10^{-2}
	WQPSO	2.58×10^{-14}	6.22×10^{-3}	4.02×10^{-2}	8.85×10^{-3}
	IQPSO	0	1.68×10^{-4}	1.07×10^{-2}	1.18×10^{-3}
Zakharov	DE	1.66×10^2	2.75×10^2	3.93×10^2	4.30×10^1
	PSO	5.09×10^0	8.06×10^0	1.28×10^1	1.44×10^0
	QPSO	1.94×10^1	4.19×10^1	8.41×10^1	1.34×10^1
	WQPSO	7.97×10^0	3.45×10^1	7.53×10^1	1.33×10^1
	IQPSO	1.96×10^{-2}	2.04×10^{-1}	7.17×10^{-1}	1.45×10^{-1}
Schwefel problem 2.2	DE	2.11×10^{-2}	3.08×10^{-1}	2.94×10^0	4.04×10^{-1}
	PSO	9.99×10^{-5}	1.72×10^2	1.06×10^3	3.03×10^2
	QPSO	2.88×10^{-3}	5.20×10^0	1.93×10^2	2.38×10^1
	WQPSO	6.38×10^{-7}	2.41×10^6	2.41×10^8	2.41×10^7
	IQPSO	4.26×10^{-23}	6.66×10^{-18}	3.70×10^{-16}	4.39×10^{-17}
Sum different powers	DE	1.56×10^{-28}	7.77×10^{-16}	6.05×10^{-14}	6.26×10^{-15}
	PSO	4.04×10^{-34}	5.40×10^{-24}	2.13×10^{-22}	2.67×10^{-23}
	QPSO	4.48×10^{-38}	2.63×10^{-28}	2.45×10^{-26}	2.45×10^{-27}
	WQPSO	9.43×10^{-50}	1.81×10^{-40}	5.65×10^{-39}	9.21×10^{-40}
	IQPSO	1.30×10^{-113}	6.02×10^{-99}	5.14×10^{-97}	5.19×10^{-98}
Sum Squares	DE	2.72×10^{-8}	2.54×10^{-7}	1.06×10^{-6}	1.57×10^{-7}
	PSO	2.60×10^{-5}	2.45×10^{-2}	2.88×10^{-1}	4.75×10^{-2}
	QPSO	1.27×10^{-11}	5.50×10^{-9}	1.29×10^{-7}	1.80×10^{-8}
	WQPSO	6.50×10^{-16}	2.38×10^{-14}	2.61×10^{-13}	3.90×10^{-14}
	IQPSO	5.54×10^{-44}	1.06×10^{-41}	3.14×10^{-40}	3.52×10^{-41}
Rotated Hyper-Ellipsoid	DE	1.76×10^{-6}	1.03×10^{-5}	4.15×10^{-5}	6.90×10^{-6}
	PSO	3.80×10^{-4}	9.52×10^{-1}	1.54×10^1	2.00×10^0
	QPSO	5.36×10^{-10}	1.07×10^{-7}	2.07×10^{-6}	2.72×10^{-7}
	WQPSO	2.13×10^{-14}	1.03×10^{-12}	1.11×10^{-11}	1.56×10^{-12}
	IQPSO	5.58×10^{-43}	2.30×10^{-40}	4.34×10^{-39}	5.45×10^{-40}
Rast	DE	2.17×10^2	2.70×10^2	3.00×10^2	1.78×10^1
	PSO	3.02×10^1	6.42×10^1	1.34×10^2	1.81×10^1
	QPSO	2.60×10^1	4.45×10^1	1.17×10^2	1.23×10^1
	WQPSO	2.05×10^1	3.69×10^1	6.50×10^1	9.00×10^0
	IQPSO	0	1.07×10^{-1}	4.98×10^0	6.30×10^{-1}
Schwefel problem 1.2	DE	1.37×10^4	3.12×10^4	5.13×10^4	7.35×10^3
	PSO	9.38×10^2	2.03×10^2	3.84×10^3	5.82×10^2
	QPSO	2.02×10^3	5.53×10^3	1.08×10^4	1.77×10^3
	WQPSO	3.58×10^2	1.41×10^3	2.87×10^3	4.51×10^2
	IQPSO	4.58×10^{-4}	3.31×10^{-1}	1.39×10^1	1.47×10^0

注:表中标注粗体数据为特征值,最优解特征值为最小值;平均解特征值为最小值;最差解特征值为最大值;均方差特征值为最小值。

表 4 特征指标数量统计

Tab.4 Statistics of characteristic indicators

维度	算法	最优解	平均解	最差解	均方差
10	DE	0	1	5	1
	PSO	0	0	6	0
	QPSO	1	0	0	0
	WQPSO	0	0	1	3
	IQPSO	11	11	0	8
40	DE	0	0	7	0
	PSO	0	0	5	0
	QPSO	0	0	0	0
	WQPSO	1	0	0	0
	IQPSO	11	12	0	12

3.4.3 有效性

为检验引入的狼群优化综合评价方法的有效性,用 IQPSO 算法对 12 个测试函数优化时的评价方法生效次数的平均值作为衡量指标,图 2 给出了 IQPSO 算法对 12 个测试函数优化时分别做 100 次独立重复实验(迭代次数设置为 1 000 次)的评价方法生效次数的平均值:

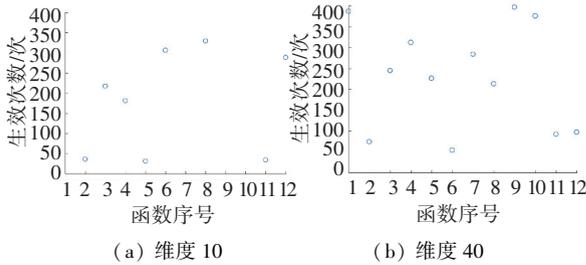


图 2 综合评价方法的生效次数

Fig.2 Effective times of the comprehensive evaluation method

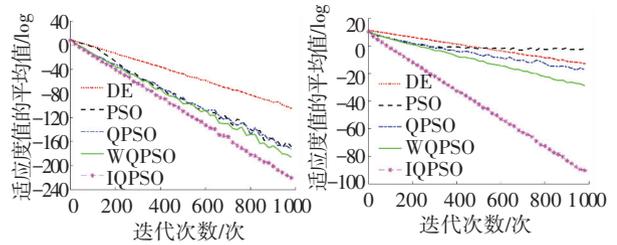
由图 2 可看出,引入的狼群算法中的综合评价方法在 12 个测试函数中均具有较高的生效次数.图 2(a)中最低生效次数为 31,最高生效次数为 497;图 2(b)中最低生效次数为 54,最高生效次数为 397;综合评价生效意味着为下一次的进化找到一个更加有利的进化方向,能够减少算法找到最优解的次数,并且丰富了最优解搜索空间使得 IQPSO 算法能够进一步提升收敛精度.

3.4.4 收敛速度

为考察不同算法的收敛速度,图 3 给出了 12 个测试函数在不同算法下的平均适应度值在不同迭代次数下的表现情况(由于取的是平均适应度值的对数值,因而当平均适应度值为 0 时,其对应的对数值为负的无穷大,故不在图中体现).

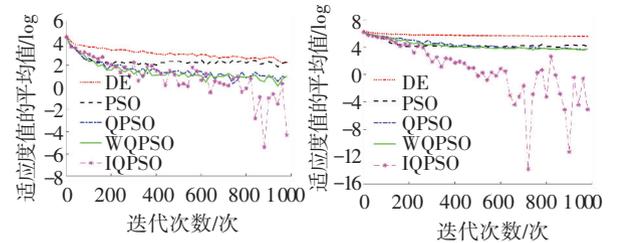
从图 3 可看出,无论测试维度是 10 还是 40,相较于其他 4 种算法,提出的 IQPSO 算法均展现出了明显的收敛优势(除了图 3(c),图 3(e),图 3(g),图 3(e)和图 3(u)),并且这种收敛优势随着迭代次数的增加更加凸显,这是因为在算法后期,粒子之间

的位置差距逐渐缩小,三种进化策略逐渐转变为在粒子自身周围寻找最优解.相较于原始 QPSO 算法的单一进化策略,三种不同的搜索策略联合,拓展了搜索范围,强化了全局搜索能力.另外,从图 3 中也可看出,本文提出的 IQPSO 算法,在维度为 40 时,其在 12 个测试函数中均具有收敛优势,而维度为 10 时,仅在 10 个测试函数中有收敛优势,这种现象展现出了 IQPSO 算法解决高维优化问题的优越性.



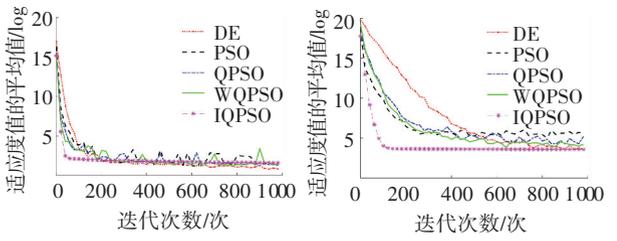
(a) f_1 , 维度为 10

(b) f_1 , 维度为 40



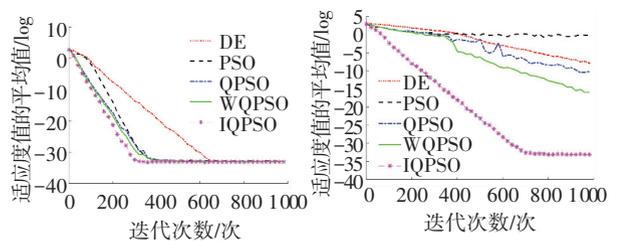
(c) f_2 , 维度为 10

(d) f_2 , 维度为 40



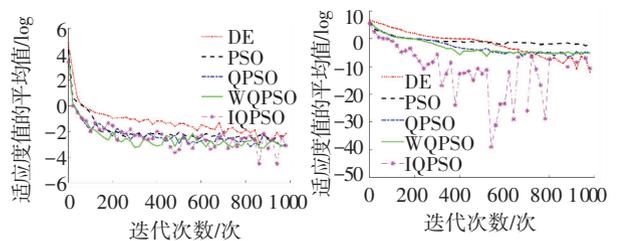
(e) f_3 , 维度为 10

(f) f_3 , 维度为 40



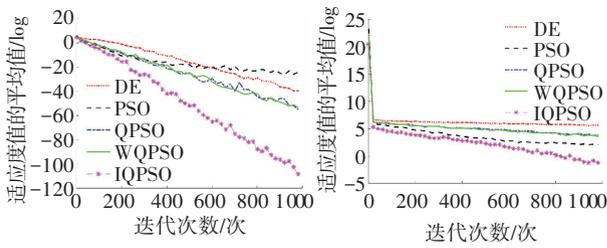
(g) f_4 , 维度为 10

(h) f_4 , 维度为 40



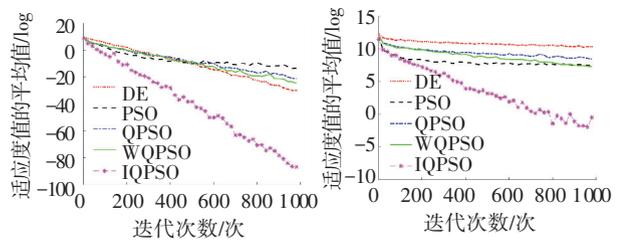
(i) f_5 , 维度为 40

(j) f_5 , 维度为 40



(k) f_6 , 维度为 10

(l) f_6 , 维度为 40

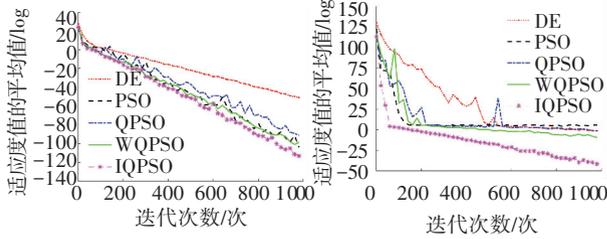


(w) f_{12} , 维度为 10

(x) f_{12} , 维度为 40

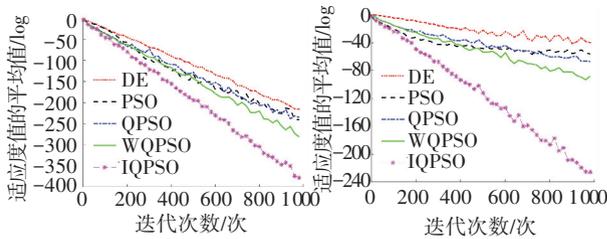
图 3 适应度值随迭代次数变化

Fig.3 The fitness value varies with the number of iterations



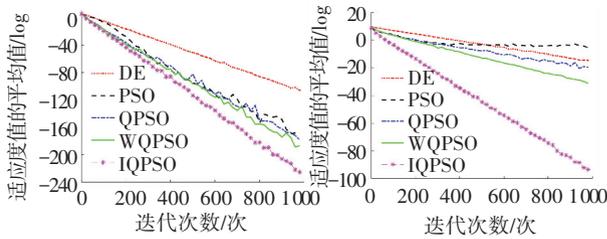
(m) f_7 , 维度为 10

(n) f_7 , 维度为 40



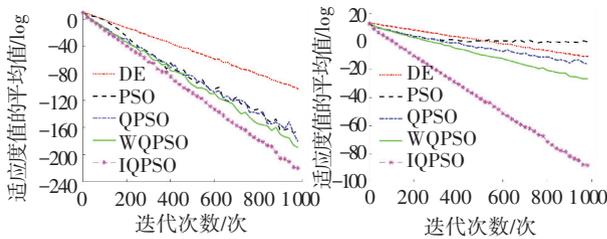
(o) f_8 , 维度为 10

(p) f_8 , 维度为 40



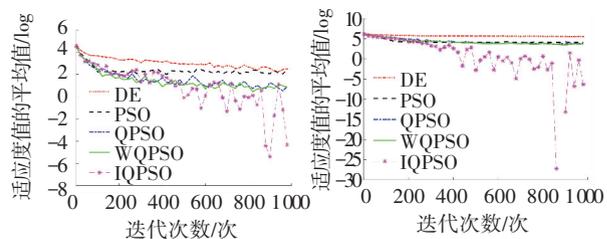
(q) f_9 , 维度为 10

(r) f_9 , 维度为 40



(s) f_{10} , 维度为 10

(t) f_{10} , 维度为 40



(u) f_{11} , 维度为 10

(v) f_{11} , 维度为 40

4 结 论

为了求解高维多模态优化问题,本文在分析 QPSO 算法原理的基础上,提出了一种改进 QPSO 算法.改进的算法通过定义的自适应动态邻域机制保持种群的“活跃性”以及采取的多策略局部吸引子方程保持种群的“多样性”,从而实现对粒子“早熟”的抑制,并且为了拓展最优解空间,引入了狼群优化算法的综合评价方法.实验仿真结果验证了算法改进策略的有效性.

参考文献

[1] 田瑾. 高维多峰函数的量子行为粒子群优化算法改进研究[J]. 控制与决策, 2016, 31(11):1967. DOI: 10.13195/j.kzyjc.2015.1132
Tian Jin. An improvement of quantum-behaved particle swarm optimization algorithm for high-dimensional multi-modal function [J]. Control and Decision, 2016, 31 (11): 1967. DOI: 10.13195/j.kzyjc.2015.1132

[2] Eberhart R, Kennedy J. A new optimizer using particle swarm theory [C]//Micro Machine and Human Science, 1995. MHS'95., Proceedings of the Sixth International Symposium on. IEEE, 1995: 39

[3] Van d B F, Engelbrecht A P. A new locally convergent particle swarm optimizer [C]//Systems, Man and Cybernetics, 2002 IEEE International Conference on. IEEE, 2002, 3(3):94. DOI: 10.1109/ICSMC.2002.1176018

[4] Sun J, Xu W, Feng B. A global search strategy of quantum-behaved particle swarm optimization [C]//Cybernetics and Intelligent Systems, [s. l.]. 2004 IEEE Conference on. IEEE, 2004, 1: 111. DOI: 10.1109/ICCIS.2004.1460396

[5] Sun J, Feng B, Xu W. Particle swarm optimization with particles having quantum behavior [C]//Evolutionary Computation, 2004. CEC2004. Congress on. IEEE, 2004, 1: 325. DOI: 10.1109/CEC.2004.1330875

[6] Xue T, Li R, Tokgo M, et al. Trajectory planning for autonomous mobile robot using a hybrid improved QPSO algorithm [J]. Soft Computing, 2017, 21(9): 2421. DOI: 10.1007/s00500-015-1956

[7] Li R G, Wu H N. Adaptive synchronization control based on QPSO algorithm with interval estimation for fractional-order chaotic systems and its application in secret communication [J]. Nonlinear Dynamics, 2018, 92(3): 935. DOI: 10.1007/s11071-018-4101