基于演化历史信息的自变异协同量子行为粒子群优化算法

赵 吉[1,2],傅 毅[1],梅 娟[1]
(1. 无锡环境科学与工程研究中心,江苏无锡 214063; 2. 江南大学物联网工程学院,江苏无锡 214122)

摘 要: 提出一种基于演化历史信息的自变异协同量子行为粒子群优化算法(ESH-CQPSO). 该算法采用二维空间分割树结构记录群体演化过程中的位置和适应值,借助群体之间的协同机制确保增强搜索能力,提高优化性能,防止过早收敛. 通过空间分割机制可以获得一个快速的近似适应度函数. 这个近似值可以提高 ESH-CQPSO 中的变异策略,使得相应的变异操作是一种无参数、多样性的自适应变异. 对比其他传统算法,通过对标准测试函数的实验结果表明,ESH-CQPSO 算法在处理多峰和单峰测试函数时具有更好的优化性能,收敛精度和收敛速度都得到了提高,证明该算法的有效性.

关键词: 量子行为粒子群优化; 演化历史信息; 自适应变异; 二维空间分割; 协同方式
中图分类号: TP301.6 **文献标识码:** A **文章编号:** 0372-2112 (2016)12-2900-08
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2016.12.013

An Improved Cooperative QPSO Algorithm with Adaptive Mutation Based on Entire Search History

ZHAO Ji[1,2], FU Yi[1], MEI Juan[1]
(1. *Research Centre of Environment Science & Engineering, Wuxi, Jiangsu* 214063, *China*;
2. *School of IoT Engineering, Jiangnan University, Wuxi, Jiangsu* 214122, *China*)

Abstract: An improved cooperative QPSO algorithm with adaptive mutation based on entire search history (ESH-CQPSO) is proposed. The proposed algorithm employs a binary space partitioning tree structure to memorize the positions and the fitness values of the evaluated solution. The cooperation mechanism between the solutions can ensure enhanced search capabilities, improve the optimize performance and prevent premature convergence. Benefiting from the space partitioning scheme, a fast fitness function approximation using the archive is obtained. The approximation is used to improve the mutation strategy in ESH-CQPSO. The resultant mutation is adaptive and parameter-less. Compared with other traditional algorithms, the experiment results on standard testing functions show that the proposed algorithm is superior regarding the optimization of multimodal and unimodal functions, with enhancement in both convergence speed and precision, which demonstrate the effectiveness of the algorithm.

Key words: quantum-behaved particle swarm optimization (QPSO); entire search history; adaptive mutate; binary space partitioning; cooperative method

1 引言

近年来,量子行为粒子群优化算法(Quantum-behaved Particle Swarm Optimization, QPSO)[1]因为其算法简单,优化性能强而受到越来越多关注. 类似于其他全局优化算法[2],QPSO 也会受制于过早收敛至局部最优的趋势,这也是群体智能优化算法的研究热点,很多文献都是围绕这个问题展开的[3-6]. 深入研究发现,QPSO 找到全局最优解的能力主要取决于两个方面:(1)跳出局部最优点的能力;(2)探索搜索空间的能力. 为此本

收稿日期:2015-05-21;修回日期:2015-11-12;责任编辑:梅志强
基金项目:国家自然科学基金(No. 61300149, No. 61105128, No. 61502203);江苏省青蓝工程资助(No. 2012-16);江苏省自然科学基金(No. BK20131106);江南大学自主科研重点计划(No. JUSRP51410B);中国博士后科学基金(No. 2014M560390)

文提出基于演化历史信息的自变异协同量子行为粒子群优化算法(Cooperative QPSO with adaptive mutation based on Entire Search History,ESH-CQPSO)以提高 QPSO 算法的优化性能.

此前也有相关文献提出了利用记忆的形式使用搜索历史信息自适应地引导搜索目标[7~9],但这些研究仅仅使用了部分搜索历史信息,而未使用的信息则被丢弃了. 演化搜索历史信息,包括估计解的位置和解的适应值对于提高群体智能算法的性能都是有价值的信息. 从直观上看,它可以用来保持多样性,引导搜索方向或指示有希望的搜索区域. 此外,当演化历史信息中出现相同的最优解时,它可以警告该搜索可能陷入局部最优. Yuen 和 Chow 提出的非重复访问机制[10]被应用于遗传算法(Genetic algorithm,GA),从而保证非重复访问遗传算法(Continuous Non-revisiting Genetic Algorithm,cNrGA)中的解不会被重复访问. 实验证明 cNrGA 比大多数 GA 算法更加优异. 但该算法中的非重复访问变异操作由于子空间限制而无法充分进行搜索,为此引入协同进化方法,利用二维空间分割机制,提出基于整个搜索空间带有重叠子区域的自变异协同量子行为粒子群优化算法,进而促进粒子的充分搜索能力以提高算法性能.

2　基于演化历史信息的协同量子行为粒子群优化算法

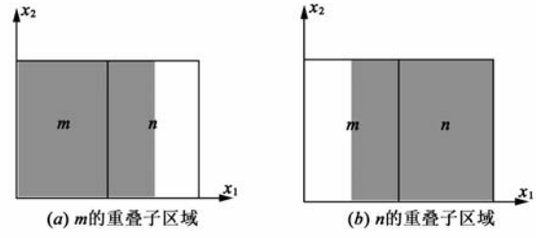
2.1　演化历史信息方案

定义 1　解 x 的子区域

假设 x 是搜索空间 S 中一个解,即 $x \in S$,S 被二维空间分割树(Binary Space Portioning,BSP)T 划分为一组子区域 $H = \cup_i h_i$. 如果 $x \in h$ 并且 h 为 T 的一个叶节点,定义子区域 $h \in H$ 为 x 的子区域.

在演化历史信息方案中,所有访问过的解的位置和适应值$[x_i, f(x_i)]$都由 BSP 树来保存. 算法迭代过程中,整个搜索空间会被划分为一组区域 H. 树结构中的每个节点代表了搜索空间的子区域. 假设 BSP 树中一个父节点 P 有两个子节点 m 和 n. 子节点沿着第 k 维将 P 的子区域线性分割为两个叠加的子区域,k 的选取满足 m 和 n 在 k 维上的距离最大,即 $k = \text{argmax}|m(k) - n(k)|$. 以这种方式,将 QPSO 算法之前产生的每个估计解记录在树的节点中. BSP 树在整个解过程中就记录了搜索空间中的所有解,利用之前解的演化历史信息对当前解进行分析. 随着迭代进行,BSP 树会随之增长. 因为树结构是依赖于每次迭代的解序列,所以 BSP 树是一个随机树,每次迭代的拓扑结构都不同.

BSP 中每个节点被划分的相邻子区域彼此之间有一定的重叠,即重叠子区域如图 1 所示. 搜索解 x 的重叠子区域 h 可以视为一个树节点的搜索. 从树的根节点开始搜索,而 h 被初始化为整个搜索空间. 每次向下移动搜索,h 就沿着指定方向不断收缩,直到搜索到达叶节点. 算法 1 给出了获取解 x 重叠子区域的伪代码. 程序中的子区域收缩方案保证了所得子区域与所有其相邻的子区域重叠. 假设图 1(a)、(b)中所示正方形表示父节点 p 的相同子区域,m 和 n 分别代表了两个估计解为 $m = [0.25, 0.5]$,$n = [0.75, 0.5]$. 因为 m 和 n 沿 x_1 轴有最大距离,所以沿 x_1 轴方向进行线性分割为两个子区域. m 和 n 的重叠子区域 h'_m 和 h'_n 分别为图 1(a)、(b)中的灰色区域. 可以发现,h'_m 和 h'_n 的合集为 h'_m 中的粒子移动到 h'_n 中提供了一个通道,反之亦然. 此外,算法 1 中的第 7 步和第 10 步的子区域收缩方案保证了所得到的子区域与所有其相邻的子区域重叠. 图 2 给出了利用 BSP 树的记录了二维空间中的四个估计解 x_1,x_2,x_3 和 x_4. 估计解分布在搜索空间为 $S = [0,1]^2$ 的二维空间内(图 2(a)),图 2(b)为对应的 BSP 树,该树包含了 7 个节点,其中根节点 R 代表整个搜索空间,节点 C 和 D 分别存储了估计解 x_1 和 x_2,节点 E 和 F 分别存储了估计解 x_3 和 x_4.



(a) m的重叠子区域　　(b) n的重叠子区域

图1　重叠子区域实例

算法 1　解 x 重叠子区域的伪代码

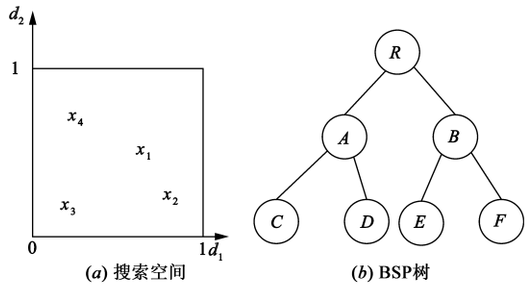
输入:(1)BSP 树 T,(2)估计解 $x \in \mathscr{R}^D$(D 为维数),(3)搜索空间 S

1. $h = \Pi_j[l_j, u_j] := S$
2. Curr_node := root node of T
3. While (Curr_node 有左右两个子节点 m 和 n)
4. 根据 $i = \text{argmax}|m(i) - n(i)|$ 获得分割维度 i,其中 $i = 1, \cdots, D$
5. If ($|m(i) - x(i)| \leqslant |n(i) - x(i)|$)
6. Curr_node := child node m
7. $u_i := n(i)$
8. Else
9. Curr_node := child node n
10. $l_j := m(j)$
11. End
12. Loop

输出:解 x 的重叠子区域 h

因为 ESH-CQPSO 算法中的 BSP 树同时记录了所有解的适应值,这个记录文档也可以看作是一个适应值 $f(x)$ 的近似值 $\tilde{f}(x)$. 如果解 y_k 的子区域是 h_k,一个未见解 x 在子区域 h_k 内,那么解 x 可以近似为 y_k,即

$f(x)\approx\tilde{f}(x)=y_k$. 随着解数量的增加并且解是一个一个记录的,$\tilde{f}(x)$的近似误差是单调减小,所以这个近似过程可以看作是一个机器学习领域简单的增量学习方法.



(*a*) 搜索空间 (*b*) BSP树

图2 BSP树记录演化历史信息

2.2 基于演化历史信息的自适应变异机制

定义2 子区域邻居

如果$X\subseteq Y$,节点y的子区域Y是叶子节点x的子区域X邻居.

如果节点x的适应值在其子区域X的邻居中值是最小的,那么子区域X是最佳的. 每一个子区域可以看作是一定邻居数量的最优值. 当找到最优子区域的近似适应度值后,子区域X的变异方向被定义为指向其最近的最优子区域方向.

在自适应变异方法中,估计解x向着离近似适应值$\tilde{f}(x)$局部增长最大的方向移动. 可以通过设置变异方向V作为指向x的最近历史最优值y,即$V=y-x$. $\tilde{f}(x)$可以看作是一个阶跃函数,因此其最优值的形式是一个N维子区域而不是一个N维点. 此外,BSP树的拓扑结构代表了子区域之间的邻接关系,因此找到x的最近最优值等同于找到x子区域中在$\tilde{f}(x)$中的最近最优子区域. 为了平衡这种梯度可能下降的方向分配搜索效果,引入变异步长α,其值为区间(0,1)内的随机值. 因此x变异后是x和y的线性组合,即$x'=x+\alpha v=(1-\alpha)x+\alpha y$. 变异操作伪代码如算法2所示.

算法2 ESH-CQPSO中的变异算法

输入:(1)需要变异的粒子x,(2)BSP树T
1. $\{x_i\}:=$在BSP树T中记录的粒子
2. $H=\cup_i h_i:=$被分割的子区域组,其中h_i是x_i的子区域
   /*搜索x的最近最优子区域y*/
3. 搜索x的子区域$h\subseteq H$
4. 找到h的最近最优子区域
5. $y:=$子区域h中的粒子 /*结束搜索x的最近最优子区域y*/
6. $\alpha:=\mathrm{Random}(0,1)$
7. $x':=\alpha x+(1-\alpha)y$

输出:变异后的粒子x'

2.3 协同机制

搜索算法,包括随机性算法(如量子行为粒子群算法和遗传算法等)在解决规模问题的高维数据时都会遭遇"维数灾难"现象. 粒子的每一个维度都会影响整体适应值. 因此,即使一些粒子某些维度上的值已经非常接近全局最优解的相应维度值,但这些粒子仍然获得较低的适应值. 协同运作的重要性就在于,在粒子群中"好"的维度信息得以保存,并且可以防止可能有用的信息被不必要地丢弃;其操作简单易实现,在高维问题中可以有效改善算法性能[5,6]. 引入动态协同机制[6],可以分别对每一维进行计算,并保存最有用的信息,以加速收敛. 动态协同进化算法的程序步骤如算法3所示.

算法3 ESH-CQPSO中的协同进化算法

输入:粒子群$P=\{p_1,p_2,\cdots,p_n\}$
1. for 粒子群中每个粒子
2. 根据QPSO算法产生5个粒子
3. $p_i:=p_{i_b}$选择5个中最好的粒子
4. $p_c=p_{ij}$
5. for 每个粒子p_k,$k=1,\cdots,5$
6. $f=f(p_k)$
7. for 每一维j
8. if $f(p_c(j,p_{kj}))<f$
9. $p_{ij}=p_{kj}$
10. Endif
11. $p_c=p_{i_b}$
12. End
13. $p_c=p_i$
14. End
15. End

输出:进化后的粒子群

2.4 基于演化历史信息的自变异协调量子行为粒子群优化算法

基于演化历史信息的自变异协调量子行为粒子群优化算法是一种实数编码进化算法. 整个算法主要由四个部分组成,即群体初始化,进化,变异和协同选择. ESH-CQPSO算法的重点在于提高基于整个演化历史信息的自变异操作,并且通过动态协同机制选择出最优粒子.

对于一个给定的D维最小化问题$F(\cdot)$,其搜索空间为R^D,算法从初始化当前粒子群粒子开始$P=\{p_1,p_2,\cdots,p_n\}$,同时BSP树初始化为只包含根节点的树T,每个粒子计算的适应度值后由树T进行记录. 粒子群$P=\{p_i\}$经过变异操作产生$Y=\{y_i\}$,即$p_i\xrightarrow{\text{变异}}y_i$. 随后根据动态协同机制,每个粒子利用QPSO算法进化公式产生4个粒子$\{p_{i1},\cdots,p_{i4}\}$,并与相应的变异体进行每一维协同进化,即第i个粒子的动态协同进化粒子群体为$\{y,p_{i1},\cdots,p_{i4}\}$,选择最好置换粒子形成新一代粒子. 产

生新的粒子群子代后需要重新计算粒子的适应度值并插入树 T 中,重复算法直到满足终止条件. 算法 4 为算法的程序步骤.

算法 4　ESH-CQPSO 算法

输入:(1)给定 D 维最小化问题 $F(\cdot)$;(2)搜索空间 $S\subset\mathscr{R}^D$;(3)粒子群规模 n

1. 初始化粒子群 $P=\{p_1,p_2,\cdots,p_n\}$
2. 计算每个粒子的适应度值:$F_i=F(p_i)$
3. $P_{\text{best}_i}:=p_j$
4. $G_{\text{best}}:=p_j\in P$ where $j=\text{argmin}\{F(p_i)\}$
   /* BSP 树初始化 */
5. 初始化 BSP 树 T 仅包含根节点
6. for $i:=1$ to n
7. 将$[p_i,F(p_i)]$记录到 T 中
8. Next i
   /* BSP 树初始化结束 */
9. While 未达到终止条件
10. for $i=1,2,\cdots,n$
11. p_i 变异 y_i
12. 利用 QPSO 更新公式获得$\{p_{i1},\cdots,p_{i4}\}$[1]
13. $\{y_i,p_{i1},\cdots,p_{i4}\}$根据动态协同机制产生新的下一代粒子 p_i
14. Next i
    /* BSP 树更新 */
15. for $i=1,2,\cdots,n$
16. 将$[p_i,F(p_i)]$记录到 T 中
17. Next i
    /* BSP 树更新结束 */
18. for $i=1,2,\cdots,n$
19. if $F(p_i)<F(P_{\text{best}_i})$
20. $P_{\text{best}_i}=p_i$
21. Endif
22. Next i
23. $G_{\text{best}}=p_j\in P$ where $j=\text{argmin}\{F(p_i)\}$
24. Loop

输出:最优解 G_{best}

3　实验设置

3.1　测试函数

本文采用一组标准测试函数对提出的算法进行测试,验证其性能,如表 1 所示. 其中 $f_1(x)-f_8(x)$ 为经典测试函数. 同时引入 CEC14 中提出的测试函数[11],F_3 为单峰函数,F_4,F_6,F_7 和 F_{16} 为简单多峰函数,F_{19} 为混合函数.

表 1　测试函数

			X	y_0
f_1	Spherical model	$\sum_{i=1}^{D}x_i^2$	$[-100,100]^D$	0
f_2	Schwefel's Problem 2.22	$\sum_{i=1}^{D}\lvert x_i\rvert+\prod_{i>0}^{D}\lvert x_i\rvert$	$[-10,10]^D$	0
f_3	Schwefel's Problem 1.2	$\sum_{i=1}^{D}\left\{\sum_{j=1}^{i}x_j\right\}^2$	$[-100,100]^D$	0
f_4	Schwefel's Problem 2.26	$-\sum_{i=1}^{D}x_i\sin\sqrt{\lvert x_i\rvert}$	$[-500,500]^D$	$-418.9298D$
f_5	Restrigin's Function	$\sum_{i>1}^{D}[x_i^2-100\cos(2\pi x_i)+10]$	$[-5.12,5.12]^D$	0
f_6	Whitely Function	$\sum_{j=1}^{D}\sum_{i=1}^{D}\left(\frac{y_{i,j}}{4000}-\cos(y_{i,j})+1\right)$ $y_{i,j}=100(x_j-x_i^2)^2+(1-x_i)^2$	$[-100,100]^D$	0
f_7	Rosenbrock's Function	$\sum_{i=1}^{D-1}[100(x_{i+1}-x_i^2)^2+(x_i-1)^2]$	$[-29,31]^D$	0
f_8	Griewank Function	$\frac{1}{4000}\sum_{i=1}^{D}x_i^2-\prod_{i=1}^{D}\cos\frac{x_i}{\sqrt{i}}+1$	$[-600,600]^D$	0
$F3$	Rotated Discus Function		$[-100,100]^D$	300
$F4$	Shifted and Rotated Rosenbrock's Function		$[-100,100]^D$	400
$F7$	Shifted and Rotated Griewank's Function		$[-100,100]^D$	700
$F16$	Shifted and Rotated Expanded Scaffer's $F6$ Function		$[-100,100]^D$	1600
$F19$	Hybrid Function 3 ($N=4$)		$[-100,100]^D$	1900

3.2 算法设置

将提出的 ESH-CQPSO 算法与标准 QPSO 算法(SQPSO),动态协同 QPSO 算法(CCQPSO)[6],连续非重复访问遗传算法(cNrGA)[10],综合学习 QPSO 算法(CLQPSO)[12]进行性能比较.

对于 SQPSO、ESH-CQPSO、CCQPSO 及 CLQPSO,收缩扩张因子 β 从 1 线性递减到 0.5;对于 cNrGA 按交叉率 $r_x=0.5$ 进行均匀交叉;CCQPSO 和 CLQPSO 分别按文献[6]和文献[12]设置. 所有算法种群的规模设置为 100,对于 $f_1(x)-f_8(x)$ 最大迭代次数设置为 1000,CEC14 测试函数最大迭代次数为 5000. 测试函数的维数 $D=30$,每个测试函数独立运行 30 次,30 次实验结束后获得测试函数的最优值、均值和方差.

4 实验结果及分析

利用不同算法优化各测试函数 30 次所得到函数最优、均值和方差如表 2 所示. 从表 2 可以看出来,ESH-CQPSO 对于绝大部分测试函数,不论是寻优精度,还是稳定性上都能获得最佳效果. 对于 F_4 函数 CLQPSO 算法的均值和方差达到最优,但 ESH-CQPSO 算法也能获得最优值,说明对于 F_4 函数 ESH-CQPSO 搜索能力还是比较强,但稳定性不够. 观察表 2 中各算法的方差,除了 F_4、F_{16} 和 F_{19} 以外,ESH-CQPSO 算法方差都是最小的,说明对于绝大多数测试函数,ESH-CQPSO 算法的稳定性也是最好的. 图 3(a) ~图 3(m)分别展示了使用 5 种算法求解各测试函数的收敛曲线. 优化 f_1 ~f_8 函数时 ESH-CQPSO 都具有最佳寻优精度,而且收敛速度远远快于其他算法. 因为 ESH-CQPSO 算法通过演化历史信息,找到最优子区域值后进行自适应变异操作,可以更快的搜索到最优值,而且利用协同机制可以保留最优信息,因此可以快速收敛到最优值. 虽然表 2 显示对于 f_1、f_2 和 F_3,CCQPSO 也能获得最优值,但从收敛曲线可以发现 ESH-CQPSO 的收敛速度要远快于 CCQPSO. 在 F_7 和 F_{16} 优化过程中,ESH-CQPSO 的优化性能都要远好于另外 4 种算法. CLQPSO 对于的 F_4 优化较好,虽然 ESH-CQPSO 算法的收敛速度最快,但在搜索后期的能力稍显不足. 从 F_{19} 的收敛曲线可以看出,虽然各算法的收敛曲线比较相近,但 ESH-CQPSO 的求解能力还是最好的,收敛速度优于其他算法,表现出了最好的收敛性能,在收敛速度和精度上都获得了最佳结果. 综上所述,对比各算法,ESH-CQPSO 算法具有更好的收敛性能.

表 2 各测试函数的优化结果

		SQPSO	ESH-CQPSO	CCQPSO	cNrGA	CLQPSO
f_1	BEST	3.17761416E-15	0.0000E00	0.0000E00	2.01625131E-05	0.10280488
	MEAN	1.47998E-14	0.0000E00	0.0000E00	9.14003181E-02	0.24313441
	STD	(3.04472E-14)	(0.0000E00)	(0.0000E00)	(0.275677938)	(0.080794574)
f_2	BEST	0.0000E00	0.0000E00	0.0000E00	8.66076483E-04	0.031108973
	MEAN	1.11684E-10	0.0000E00	0.0000E00	0.014986892	0.047356203
	STD	(4.15217E-10)	(0.0000E00)	(0.0000E00)	(0.012854059)	(0.008233747)
f_3	BEST	45.46166824	0.0000E00	4.90309E-05	1870.54556310	11909.75678479
	MEAN	186.3872131	0.0000E00	7.755494E-03	3353.915859	17341.37742
	STD	(104.0295677)	(0.0000E00)	(9.302 E-03)	(1894.914266)	(2989.341129)
f_4	BEST	-10552.99604	-12569.48662	-10789.87748	-12569.48660	-12569.39291
	MEAN	-7724.424098	-12569.48662	-9621.236512	-12438.97273	-12569.30445
	STD	(1839.533905)	(1.85009E-12)	(1052.096436)	(129.3499778)	(0.060056446)
f_5	BEST	6.96658336	0.0000E00	5.969754343	1.69498487E-03	0.13957755
	MEAN	15.71033071	0.0000E00	11.51326251	0.845910299	0.261776014
	STD	(4.179950722)	(0.0000E00)	(3.131219788)	(1.20528348)	(0.094784807)
f_6	BEST	167.988932	0.0000E00	5.11547708E-1	83.72666934	930.3931761
	MEAN	520.1031329	3.29886908E-1	210.0628884	159.7049732	987.5081823
	STD	171.1253268	(6.08193271E-1)	(139.2575811)	(60.18946741)	35.50248595
f_7	BEST	8.96354670	0.0000E00	16.80000815	14.77122802	177.06310304
	MEAN	25.71260941	1.7356007	29.41097	35.22865152	248.6310265
	STD	(1.573154449)	(6.605046823)	(20.54513241)	(20.92783323)	(42.63673184)

续表

		SQPSO	ESH-CQPSO	CCQPSO	cNrGA	CLQPSO
f_8	BEST	1.35447209E-14	0.0000E00	0.0000E00	3.8530308E-03	0.15966625
	MEAN	0.008289674	0.0000E00	0.00821035	0.11359066	0.423170878
	STD	(0.009056309)	(0.0000E00)	(0.010412577)	(0.145049358)	(0.078499638)
F_3	BEST	300.6370925	300	300	2219.206249	302.5599846
	MEAN	324.9545027	300	300	13976.26597	485.3156056
	STD	(36.3434255)	(1.24895E-13)	(1.88862E-11)	(8918.856761)	(180.6492199)
F_4	BEST	404.7112233	400.5305189	401.880502	406.2815043	424.9381277
	MEAN	483.5517841	450.8843823	476.407026	500.0765755	435.5459942
	STD	(28.43023807)	(38.9646409)	(33.21739716)	(34.87640064)	(4.949274492)
F_7	BEST	700	700	700	700.0004949	700.0000072
	MEAN	700.0060727	700	700.0050081	700.065836	700.0000349
	STD	(0.011329917)	(2.98556E-14)	(0.006793092)	(0.058539411)	(3.1831E-05)
F_{16}	BEST	1609.682951	1601.993427	1607.166417	1608.112104	1609.4677
	MEAN	1611.006224	1603.25258	1608.695524	1609.614678	1609.992514
	STD	(0.549202026)	(0.570662724)	(0.749200249)	(0.798261829)	(0.306569496)
F_{19}	BEST	1903.302441	1904.282716	1607.166417	1905.207875	1905.948957
	MEAN	1906.02606	1905.393292	1608.695524	1913.962304	1907.833315
	STD	(1.294344375)	(1.129562243)	(0.749200249)	(18.44748002)	(0.603899214)

为了判断 ESH-CQPSO 算法与其他 4 种算法的性能是否存在显著性差异,除了对算法的精度和收敛性进行比较外,本文还进行了 Wilcoxon 秩和检验[13]. 表 3 给出了 ESH-CQPSO 算法与其他 4 种算法的 Wilcoxon 秩和检验结果. 从表 3 可以看出对于 f_1、f_2 和 F_{19},ESH-CQPSO 和 CCQPSO 的显著性差异不大,而对于 F_7,SQPSO 也能获得和 ESH-CQPSO 相近的优化性能. 因此对于绝大多数测试函数,ESH-CQPSO 算法都能获得 $h=1$,说明对比其他算法,ESH-CQPSO 的优化性能有显著提高. 由此可以看出不管是在单峰函数还是多峰函数中 ESH-CQPSO 算法都表现出良好的收敛性和搜索精度,优化性能显著提高.

表 3　Wilcoxon 秩和检验结果

	ESH-CQPSO vs CCQPSO		ESH-CQPSO vs SQPSO		ESH-CQPSO vs CLQPSO		ESH-CQPSO vs cNrGA	
	p	h	p	h	p	h	p	h
f_1	NaN	0	1.2118E-12	1	1.2118E-12	1	1.2108E-12	1
f_2	NaN	0	5.2823E-06	1	1.2118E-12	1	1.2118E-12	1
f_3	1.2118E-12	1	1.2118E-12	1	1.2118E-12	1	1.2118E-12	1
f_4	1.2118E-12	1	1.2118E-12	1	1.2118E-12	1	1.2118E-12	1
f_5	1.1990E-12	1	1.2118E-12	1	1.2118E-12	1	1.2118E-12	1
f_6	2.6139E-11	1	1.2676E-11	1	1.2676E-11	1	1.2676E-11	1
f_7	2.3080E-09	1	1.5964E-09	1	1.7895E-11	1	3.3002E-11	1
f_8	1.2705E-05	1	1.2118E-12	1	1.2118E-12	1	1.2118E-12	1
F_3	1.3100E-08	1	2.9972E-11	1	2.9972E-11	1	2.9972E-11	1
F_4	9.2113E-05	1	8.2919E-06	1	2.7100E-02	1	7.2208E-06	1
F_7	2.3638E-12	1	1.5610E-01	0	2.3620E-12	1	2.3638E-12	1
F_{16}	3.0180E-11	1	3.0199E-11	1	3.0199E-11	1	3.0199E-11	1
F_{19}	2.6310E-01	0	1.1650E-01	0	6.7425E-11	1	4.9074E-09	1

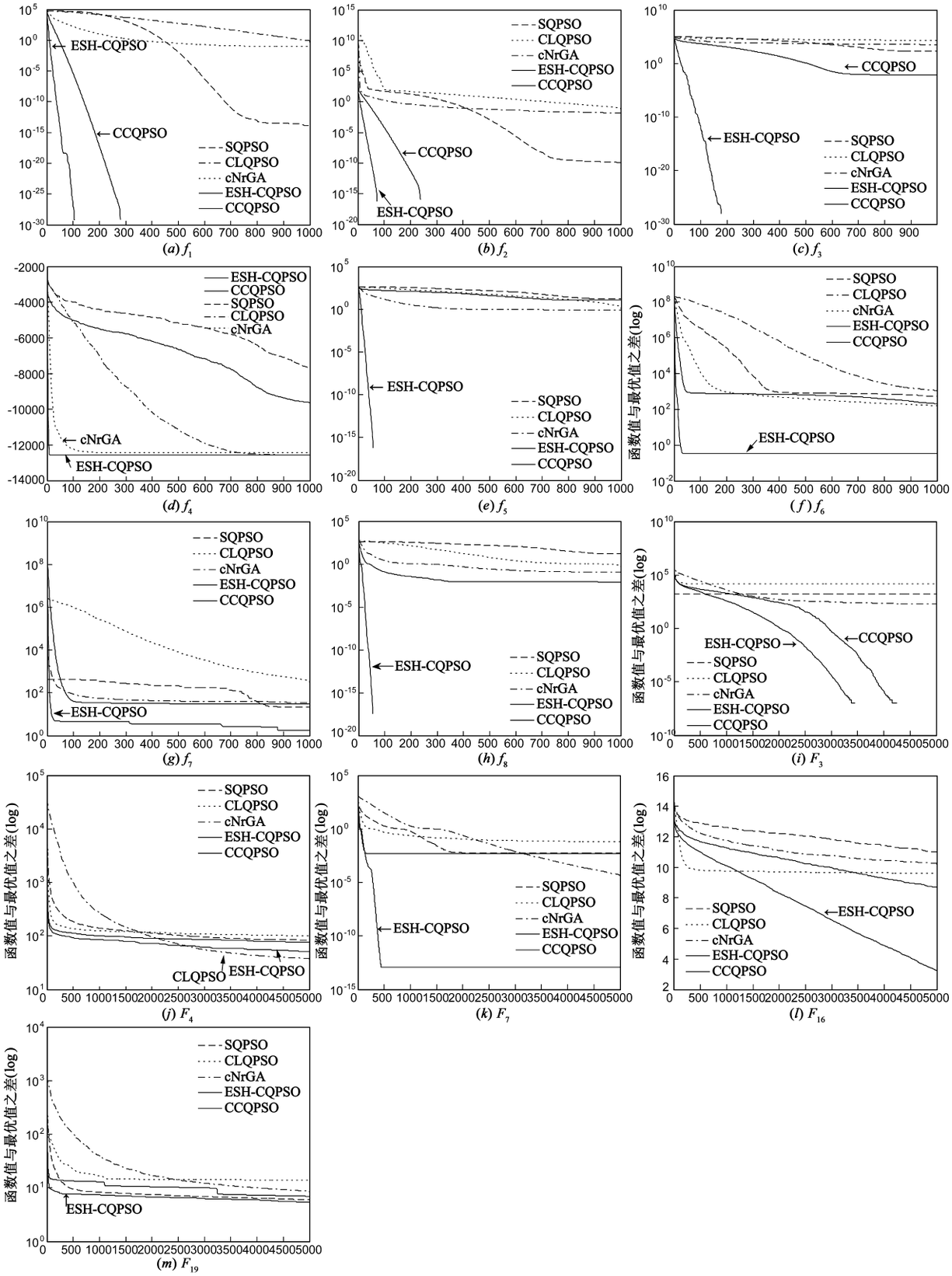

ESH-CQPSO
CCQPSO
SQPSO
CLQPSO
cNrGA
ESH-CQPSO
CCQPSO
(a) f_1
(b) f_2
(c) f_3
(d) f_4
(e) f_5
(f) f_6
(g) f_7
(h) f_8
(i) F_3
(j) F_4
(k) F_7
(l) F_{16}
(m) F_{19}
函数值与最优值之差(log)

图3 算法收敛曲线

5 结论

早熟和多样性是 QPSO 算法需要解决的问题,为此本文提出了基于演化历史信息的自变异协同量子行为粒子群优化算法(ESH-CQPSO). 该算法利用二维空间分割树(BSP)记录演化过程中的估计解和适应值,BSP 树将连续搜索空间进行划分为不同子区域,并且产生近似适应值函数,从而获得无参的自适应变异. 同时引入动态协同机制,通过更好地利用上下文信息,在协同过程中各个维度上连续动态地进行更新,最大限度地利用任何新的信息,提高变异粒子的性能,防止算法进入局部收敛,阻止早熟发生. 通过实验结果表明,提出的 ESH-CQPSO 算法具有更好的寻优能力,收敛速度和收敛精度都有所提高,是一个可靠的有效的全局优化算法. 本文提出的改进算法能在理论上证明具有一定的优越性,但在实际问题中还需要进一步进行测试,因此将 ESH-CQPSO 算法应用于实际工程问题是今后的研究重点,下一步研究内容就是采用一定的工程问题作为应用,如图像分割、数据聚类、生物代谢途径参数辨识等,进一步验证 ESH-CQPSO 算法的性能.

参考文献

[1] 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. 31 – 50.

[2] Kennedy J,Eberhart R C. Particle swarm optimization[A]. Proc of IEEE Int. Conf. on Neural Networks[C]. Perth:IEEE Press,1995. 1942 – 1948.

[3] 刘朝华,李小花,章兢. 精英免疫克隆选择的协同进化粒子群算法[J]. 电子学报,2013,14(11):2167 – 2173.
Liu Zhao-hua, Li Xiao-hua, Zhang Jing. Co-evolutionary particle swarm optimization algorithm based on elite immune clonal selection[J]. Acta Electronica Sinica,2013,14(11):2167 – 2173. (in Chinese)

[4] Sun J,Fang W,Wu X J,Palade V,Xu W B. Quantum-behaved particle swarm optimization:analysis of individual particle behavior and parameter selection[J]. Evolutionary Computation,2012,20(3):349 – 393.

[5] Li Y, Xiang R, Jiao L, et al. An improved cooperative quantum-behaved particle swarm optimization[J]. Soft Computing,2012,16(6):1061 – 1069.

[6] Li Y,Jiao L,Shang R,et al. Dynamic-context cooperative quantum-behaved particle swarm optimization based on multilevel thresholding applied to medical image segmentation[J]. Information Sciences,2015,294:408 – 422.

[7] Glover F,Laguna M. Tabu Search[M]. New York:Springer,2013. 3261 – 3362.

[8] Reynolds R G. An introduction to cultural algorithms[A]. Proceedings of the Third Annual Conference on Evolutionary Programming[C]. River Edge,NJ,USA :World Scientific Publishing Co,Inc,1994. 131 – 139.

[9] Farmer J D,Packard N H,Perelson A S. The immune system,adaptation,and machine learning[J]. Physica D:Nonlinear Phenomena,1986,22(1):187 – 204.

[10] Yuen S Y,Chow C K. Continuous non-revisiting genetic algorithm[A]. Proceedings of IEEE Conference Evolutionary Computation[C]. Trondheim:IEEE,2009. 1896 – 1903.

[11] Liang J J,Qu B Y,Suganthan P N. Problem definitions and evaluation criteria for the CEC 2014 special session and competition on single objective real-parameter numerical optimization[R]. Singapore:Computational Intelligence Laboratory,Zhengzhou University,2013.

[12] 陈伟,周頔,孙俊,须文波. 一种采用完全学习策略的量子行为粒子群优化算法[J]. 控制与决策,2012,27(5):719 – 723.
Chen Wei,Zhou Di,Sun Jun,XU Wenbo. Improved quantum-behaved particle swarm optimization algorithm based on comprehensive learning strategy[J]. Control and Decision,2012,27(5):719 – 723. (in Chinese)

[13] Li Zhonghua, He Chunhui, Li Jianming, Tan Hongzhou. Adaptive hierarchical artificial immune system and its application in RFID reader collision avoidance[J]. Applied Soft Computing,2014(21):119 – 138.

作者简介

赵　吉　女,1980 年 9 月生,江苏无锡人,博士,博士后,副教授. 主要从事进化智能算法理论与应用等领域研究.
E-mail:queenji97@ sohu. com

傅　毅　女,1981 年 11 月生,安徽六安人,博士,博士后,副教授. 主要从事计算机数值模拟、计算生物学等领域研究.