

文章编号:1001-9081(2015)03-680-05

doi:10.11772/j.issn.1001-9081.2015.03.680

基于多维问题的交叉算子量子粒子群优化算法

奚茂龙^{1,2*}, 盛歆漪², 孙俊²

(1. 无锡职业技术学院 控制技术学院, 江苏 无锡 214121; 2. 江南大学 物联网工程学院, 江苏 无锡 214122)

(*通信作者电子邮箱 wx_xml@hotmail.com)

摘要:针对量子行为粒子群优化(QPSO)算法在求解多维问题时优秀维信息丢失的问题,引入交叉算子的策略,改善解的质量,提升算法性能。首先,分析了量子粒子群算法进化过程中的粒子整体更新评价策略,发现各维信息之间相互干扰,会丢失已经搜索到的优秀维信息;然后,指出如果采用逐维进化方法,会指数级增加算法的复杂度;最后,提出对进化过程中的问题解采用多点交叉的策略增加优秀维信息的保留概率,并将改进后的量子粒子群算法与线性下降参数控制策略、非线性下降参数控制策略方法通过12个CEC2005 benchmark测试函数进行了比较,并对结果进行了分析。仿真结果显示,所提算法比改进前在10个测试函数中取得了明显的改进效果,而比其他2种改进算法也在7个测试函数中取得了优势。因此该算法能够有效提升量子粒子群优化算法的性能。

关键词:粒子群优化算法;交叉算子;维信息;量子行为;交叉率

中图分类号: TP301.6 文献标志码:A

Quantum-behaved particle swarm optimization algorithm with crossover operator to multi-dimension problems

XI Maolong^{1,2*}, SHENG Xinyi², SUN Jun²

(1. School of Control Technology, Wuxi Institute of Technology, Wuxi Jiangsu 214121, China;

2. School of Internet of Things, Jiangnan University, Wuxi Jiangsu 214122, China)

Abstract: According to the problem that better dimensions information of particles will loss in Quantum-behaved Particle Swarm Optimization (QPSO) algorithm when solving multi-dimensions problems, a strategy with crossover operator was introduced and the quality of solutions and the performance of algorithm would be improved. Firstly, the whole update and evaluation strategy on solutions in algorithm was analyzed and the better dimensions information of particles would loss because of the mutual interference between dimensions. Secondly, when the evolution was executed dimension by dimension, the algorithm complexity would increase exponentially. Finally, multi-crossover method was employed to increase the retaining probability of excellent dimension information. The comparison and analysis results of the proposed method, with linearly decreased coefficient control method and non-linearly decreased coefficient control method on 12 CEC2005 benchmark functions were given. The simulation results show the modified algorithm can greatly improve the QPSO performance compared with the basic QPSO in 10 functions and also get better performance in 7 functions compared with the other two QPSO variants. Therefore, the proposed method can improve the performance of QPSO effectively.

Key words: Particle Swarm Optimization (PSO) algorithm; crossover operator; dimension information; quantum-behaved; crossover rate

0 引言

粒子群优化(Particle Swarm Optimization, PSO)算法是 Kennedy 等^[1]提出的群体优化算法, 算法源于对鸟群觅食行为的模拟。基本粒子群算法的进化方程为:

$$\begin{aligned} v_{id}(t+1) &= v_{id}(t) + c_1 \cdot r_1 \cdot (p_{id}(t) - x_{id}(t)) + \\ &\quad c_2 \cdot r_2 \cdot (P_{gd}(t) - x_{id}(t)) \end{aligned} \quad (1)$$

$$x_{id}(t+1) = v_{id}(t+1) + x_{id}(t) \quad (2)$$

在算法进化过程中, 粒子共享群体的“最优”位置信息 P_g , 通过搜索多维问题解空间, 不断地更新自身的速度信息 v_i 和位置信息 x_i , 不断地比较和追随候选的问题空间解, 最终发现问题的最优解或者局部最优解。自 PSO 算法提出以来,

因其进化方程简单、搜索能力较好、收敛速度快得到广泛的应用。但从 PSO 算法本身并不是一种全局优化算法^[2], 许多学者对它进行了大量的研究工作, 提出了一些改进方法, 取得了一定的改进效果^[3-4]。Sun 等在分析粒子群优化算法机理的基础上, 深入研究了社会智能群体进化过程, 将量子理论引入了 PSO 算法, 提出了具有全局搜索能力的量子行为粒子群优化(Quantum-behaved Particle Swarm Optimization, QPSO)算法^[5-6]。QPSO 算法具有计算简单、编程易于实现、控制参数少等特点, 引起了国内外相关领域众多学者的关注和研究^[7-9], 也在一些实际问题中得到了应用^[10-12]。

在 QPSO 算法中, 群体中粒子信息的交互是通过粒子的每一维(多维问题)信息的共享而进化的, 而在粒子评价时,

收稿日期:2014-10-20;修回日期:2014-11-13。 基金项目:国家自然科学基金资助项目(61170119);江苏省“青蓝工程”资助项目。

作者简介:奚茂龙(1977-),男,江苏盐城人,副教授,博士,主要研究方向:智能算法; 盛歆漪(1979-),女,江苏无锡人,讲师,博士,主要研究方向:进化计算; 孙俊(1972-),男,江苏无锡人,副教授,博士,主要研究方向:算法设计与分析。

是以粒子的整体信息(多维信息的结合)评价的。也就是说,迭代进化后的粒子总体优于进化前的粒子并不表示,进化后粒子的每一维信息都优于进化前粒子的对应维信息,这样在进化过程中就会丢失整体较差粒子中维信息较好的情况。如果将QPSO算法中引入交叉算子,通过交叉操作,将增加粒子维信息保留的概率,从而提高算法的性能。基于这种思想,同时受遗传算法的启发,本文提出了一种具有交叉算子的量子粒子群优化算法(QPSO with Crossover Operator, COQPSO),改善算法的整体性能。

1 量子粒子群优化算法

基本PSO算法不是全局优化算法,Clerc等^[13]通过对PSO算法中粒子运动轨迹的分析,指出如果每个粒子都能够收敛到它的局部吸引点 $P_i = (P_{i1}, P_{i2}, \dots, P_{iD})$,那么PSO算法可能收敛。其中:

$$p'_{id}(t) = \frac{c_1 r_{1d}(t) P_{id}(t) + c_2 r_{2d}(t) P_{gd}(t)}{c_1 r_{1d}(t) + c_2 r_{2d}(t)}; 1 \leq d \leq D \quad (3)$$

并设

$$\varphi_d(t) = \frac{c_1 r_{1d}}{c_1 r_{1d}(t) + c_2 r_{2d}(t)} \quad (4)$$

式(3)、(4)中, t 是算法当前迭代次数, $r_{1d}(t)$ 和 $r_{2d}(t)$ 是 $[0,1]$ 的随机数, P_{id} 为粒子的当前最优位置, P_{gd} 为群体的全局最优位置。

Sun等^[5]将PSO算法的进化系统假设为量子系统,在量子空间中,粒子的速度和位置是不能同时确定的,每个粒子的状态都由波函数 ψ 来确定, $|\psi|^2$ 是粒子位置的概率密度函数。假定在第 t 次迭代,粒子 i 在 D 维的空间运动,该粒子在第 j 维的势阱为 $p'_{ij}(t)$,则在第 $t+1$ 次迭代可以得到粒子 i 的波函数为:

$$\psi[x_{ij}(t+1)] = \frac{1}{\sqrt{L_{ij}(t)}} \exp[-|x_{ij}(t+1) - p'_{ij}(t)|/L_{ij}(t)] \quad (5)$$

则概率密度函数 Q 为:

$$Q[x_{ij}(t+1)] = |\psi[x_{ij}(t+1)]|^2 = \frac{1}{L_{ij}(t)} \exp[-2|x_{ij}(t+1) - p'_{ij}(t)|/L_{ij}(t)] \quad (6)$$

概率分布函数 F 为:

$$F[x_{ij}(t+1)] = \exp[-2|x_{ij}(t+1) - p'_{ij}(t)|/L_{ij}(t)] \quad (7)$$

应用Monte Carlo方法,可以计算出在第 $t+1$ 次迭代,第 i 粒子第 j 维的位置:

$$x_{ij}(t+1) = p'_{ij}(t) \pm \frac{L_{ij}(t)}{2} \ln[1/u_{ij}(t)]; \\ u_{ij}(t) \sim U(0,1) \quad (8)$$

其中 $L_{ij}(t)$ 的值由下式确定:

$$L_{ij}(t) = 2\beta \cdot |M_j(t) - x_{ij}(t)| \quad (9)$$

其中 M 称为平均最优位置,也记为 m_{best} ,是所有粒子自身最优位置的中心点,由下式计算得到:

$$M(t) = (m_1(t), m_2(t), \dots, m_D(t)) = \left(\frac{1}{N} \sum_{i=1}^N P_{i1}(t), \frac{1}{N} \sum_{i=1}^N P_{i2}(t), \dots, \frac{1}{N} \sum_{i=1}^N P_{iD}(t) \right) \quad (10)$$

其中: N 是群体规模的大小, P_i 是第 i 个粒子自身最优位置。因

此,可以得到粒子的位置更新方程:

$$x_{ij}(t+1) = p'_{ij}(t) \pm \beta \cdot |M_{ij}(t) - x_{ij}(t)| \cdot \ln[1/u_{ij}(t)]; \\ u_{ij}(t) \sim U(0,1) \quad (11)$$

其中: β 称为压缩扩张因子,用来调节粒子的收敛速度。

粒子当前最优位置 P_i 和全局最优位置 P_g 的更新方式与基本PSO算法的更新方式完全相同,即:

$$P_i(t+1) = \begin{cases} P_i(t), & f(X_i(t+1)) \geq f(P_i(t)) \\ x_i(t+1), & f(X_i(t+1)) < f(P_i(t)) \end{cases} \quad (12)$$

$$P_g(t+1) = \arg \min_{P_i} f(P_i(t+1)); 1 \leq i \leq N \quad (13)$$

这里粒子位置更新式(11)的PSO算法为具有量子行为的粒子群优化(Quantum-behaved Particle Swarm Optimization, QPSO)算法^[5-6]。

2 具有交叉算子的QPSO算法

2.1 算法思想

在QPSO算法中,粒子的进化过程是通过迭代方程(4)、(10)、(11)实现的。在多维问题中(假设为 m 维),粒子表示为 $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,m})$ 。分析式(11)可知,在进化迭代时,粒子的每一维信息根据式(11)计算,在此过程中并不对粒子进行评价。完成粒子的每一维进化操作后,对综合了多维信息后的粒子进行优劣评价。因此,算法过程中并不对粒子的维信息单独评价,只对维信息的结合体(粒子)评价,具体过程如下:

```
设定群体中有n个粒子,问题解为m维
For each particle X_i of t+1 generation
  For each dimension d of X_i
    Iteration of dth dimension using
      formula(11)
    endfor
    If f(X_i^{t+1}) < f(X_i^t) then
      X_i^t = X_i^{t+1}
    Endif
  endfor
```

这样,QPSO算法在进化过程中,就可能会丢失粒子信息中单维已经出现的较好信息。

例如: $f(x) = (x_1 - 2)^2 + (x_2 - 3)^2 + (x_3 - 4)^2$ 的最小化问题。在第 t 次迭代时,产生的可行性解为 $(4,3,5)$, $f(x,t)$ 的值为5;在第 $t+1$ 次迭代时,解的每一维进行分别迭代,产生了更新后的可行性解 $(3,4,5)$, $f(x,t+1)$ 的值为3,由于 $f(x,t+1) < f(x,t)$,则 $x_i(t+1) = (3,4,5)$ 被保留,并进入下一轮循环迭代。将 $x_i(t+1)$ 和 $x_i(t)$ 比较后,可以看出,在第 t 代,算法已经搜索到可行性解的第2维的最优值3,但由于粒子进化过程中,每一维的同步操作,在 $t+1$ 代进化时,可行性解的第2维更新维4,丢失了在 t 代时更优的值3,从整个算法进化来看,减慢了算法寻优的速度,降低整个算法的性能。

2.2 具有交叉算子的QPSO算法过程

根据论述的QPSO算法进化过程中存在的解空间维信息丢失的问题,可以有直接的解决办法,就是粒子在进化过程中,信息的更替不依赖于粒子多维信息的同时更替,并比较保留,而是在每一轮信息更替时,以粒子的维信息为单位,粒子的每一维进行更替进化,并在每一维更新后即进行信息的比较,从而判断当前维信息更新与否。粒子的维信息更新如下:

设定群体中有 n 个粒子,问题解为 m 维

For each particle X_i of $t + 1$ generation

For each dimension d of X_i

If $f(X_{i,1}^t, X_{i,2}^t, \dots, X_{i,d}^t, X_{i,d+1}^{t+1}, \dots, X_m^t) < f(X_{i,1}^t, X_{i,2}^t, \dots, X_{i,d}^t, X_{i,d+1}^t, \dots, X_m^t)$

then

$X_{i,d}^t = X_{i,d}^{t+1}$

Endif

endfor

endfor

在此方法中,由于粒子信息在迭代更新过程中对粒子的每一维信息进行比较,算法复杂度由 $O(n)$ 变为 $O(n^m)$,随着问题维数的增加,复杂度成指数级增加,增加了大量的计算开销,实际可行性较小。

为了在不增加 QPSO 算法复杂度的情况下,能够保留粒子的维信息,本文在 QPSO 算法中引入了交叉算子,即通过群体中粒子间的维交叉操作,产生新的粒子并进入下一轮迭代。这样通过交叉操作将不同粒子中较好的维信息有机会组合在一起,形成新的粒子,从而保留群体进化过程中已经搜索到的较优的维信息,提高整体算法性能。交叉操作包括单点交叉和多点交叉。下面以 2 点交叉为例讲解交叉过程,其中 X_i^t , X_j^t 为第 t 代进化过程中群体的两个随机粒子:

$$X_i^t = (x_{i,1}^t, x_{i,2}^t, x_{i,3}^t, \dots, x_{i,m-3}^t, x_{i,m-2}^t, x_{i,m-1}^t, x_{i,m}^t)$$

$$X_j^t = (x_{j,1}^t, x_{j,2}^t, x_{j,3}^t, \dots, x_{j,m-3}^t, x_{j,m-2}^t, x_{j,m-1}^t, x_{j,m}^t)$$

在两个粒子的维上随机选择 2 个交叉点,在本例中选取了粒子维中的第 2 维和第 $m - 2$ 维为交叉点,则经过交叉后,得到新的粒子:

$$X_i^{t+1} = (x_{i,1}^t, x_{i,2}^t, x_{j,3}^t, \dots, x_{i,m-3}^t, x_{j,m-2}^t, x_{i,m-1}^t, x_{i,m}^t)$$

$$X_j^{t+1} = (x_{j,1}^t, x_{j,2}^t, x_{i,3}^t, \dots, x_{j,m-3}^t, x_{i,m-2}^t, x_{j,m-1}^t, x_{j,m}^t)$$

交叉后产生的新的粒子进入新一轮的粒子进化中,这里

将提出的具有交叉算子的 QPSO 算法称为 COQPSO 算法 (QPSO with Crossover Operator)。在 COQPSO 算法中,群体中的粒子将按照一定的比例进行交叉。

COQPSO 算法设计如下。

- 1) 初始化参数设置。包括群体个数;扩张收缩系数 β 取值范围;算法迭代次数;群体中参与交叉操作的粒子比例 γ ;随机产生的初始解 $x(i)$,并设置 $pbest(i) = x(i)$,并计算全局最优值 $gbest$ 。
- 2) 根据式(3)计算局部吸引子 $p(i)$ 。
- 3) 根据式(10)计算平均最优值 $mbest$ 。
- 4) 根据设定的群体 S_t 中参与交叉操作的粒子比例 γ ,随机选择粒子,进行交叉操作,生成新的子群体 S_{t1} 。
- 5) 根据 QPSO 算法迭代方程(10)、(11)更新群体中未参与交叉操作的粒子的新位置,生成新的子群体 S_{t2} 。
- 6) 子群体 S_{t1} 和子群体 S_{t2} 组成新的群体 S_{t+1} 。
- 7) 回到 4) 直至循环条件结束。

3 COQPSO 算法仿真分析

3.1 仿真设置

在 COQPSO 算法中,群体中参与交叉操作的粒子比例 γ 没有确定,为了研究 γ 的取值对算法性能的影响,本文对 γ 分别取值 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1.0 的情况进行测试分析,测试函数使用 CEC2005 benchmark 测试函数^[14],其中 F1 到 F6 为单峰函数,F7 到 F12 为多峰函数。算法中,其他的参数设置为:收缩扩张因子 β 从 1.0 线性下降到 0.5,问题维数为 30,算法对应的最大迭代次数为 5 000,粒子数为 40;每个问题的求解都应用算法随机地独立运行 50 次。表 1 给出了 γ 不同取值条件下时,目标测试函数值的平均最优值。

表 1 COQPSO 算法不同 γ 取值下的目标测试函数的平均最优值

γ	平均最优值											
	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12
0.2	2.4783E-27	0.1085	2.6236E+06	449.0741	2.6065E+03	65.3979	0.0128	7.9581E-15	20.5447	105.1491	26.2957	2.0207E+04
0.3	2.4654E-27	0.0939	3.0767E+06	393.2316	2.6032E+03	52.9934	0.0195	8.5265E-15	21.9178	108.3188	28.7797	1.9824E+04
0.4	2.3908E-27	0.0949	2.4050E+06	431.4541	2.7687E+03	44.2702	0.0188	8.1712E-15	21.4726	108.8499	27.4229	2.5453E+04
0.5	2.4560E-27	0.1026	3.1237E+06	339.4429	2.4445E+03	69.8321	0.0182	8.2423E-15	22.5305	118.4556	27.5517	1.6776E+04
0.6	2.5139E-27	0.0751	2.5781E+06	344.0046	2.3794E+03	60.5539	0.0155	7.3896E-15	23.4980	108.8292	29.2903	2.0547E+04
0.7	2.3843E-27	0.0706	2.8447E+06	299.6406	2.5619E+03	62.3032	0.0156	7.9581E-15	22.9414	108.6882	31.4084	2.1101E+04
0.8	2.4910E-27	0.0670	2.6533E+06	294.7308	2.2669E+03	48.3101	0.0165	7.8160E-15	25.3737	126.7743	30.6725	1.6372E+04
0.9	2.4837E-27	0.0703	2.5986E+06	301.6700	2.2453E+03	38.9780	0.0146	0.4078	23.7328	124.8804	31.9503	2.9553E+04
1.0	2.3646E-27	0.0712	3.2698E+06	313.3528	2.3347E+03	54.4467	0.0164	0.4067	23.7341	125.2802	30.6272	2.1602E+04

同样,为了测试提出的具有交叉算子的 QPSO 算法是否提高了算法的性能,本文还和文中提出的线性递减参数调整策略 (LDQPSO) 和非线性递减的参数调整策略 (开口向上 UPQPSO, 开口向下 DNQPSO) 的算法进行了比较^[6,15], 测试条件仍设置相同。表 2 给出了几种算法的在不同测试函数下的平均最优值。图 1 给出了在优化测试函数时,不同算法的平均收敛曲线。

3.2 仿真结果与讨论

由表 1 中的仿真结果可以看出,群体中粒子的交叉率 γ 的取值对算法性能是有影响的,不同的取值在优化不同的测试函数时算法的表现也有差别。对于 Shifted Sphere 函数(F1)、Schwefel's Problem 2.6 with Global Optimum on the Bounds 函数

(F5) 和 Shifted Rastrigin's 函数(F9),不同的 γ 取值对仿真的结果影响不大,仿真结果比较接近;对于 Shifted Schwefel's Problem 1.2 函数(F2)、Shifted Schwefel's Problem 1.2 with Noise in Fitness(F4) 和 Shifted Rotated Griewank's Function without Bounds 函数(F7), γ 取值对算法的影响成规律变化,随着 γ 值的增加,对应测试函数的仿真结果越来越好;而对于 Shifted Rotated High Conditioned Elliptic 函数(F3)、Shifted Rotated Ackley's 函数(F8)、Shifted Rotated Rastrigin's(F10) 和 Shifted Rotated Weierstrass 函数(F11),情况正好相反,随着 γ 值的增加,对应测试函数的仿真结果越来越差;对于 Shifted Rosenbrock 函数(F6) 和 Shifted Schwefel's Problem 1.2 函数(F2),则 γ 取值在 0.6 附近时,仿真结果较好。

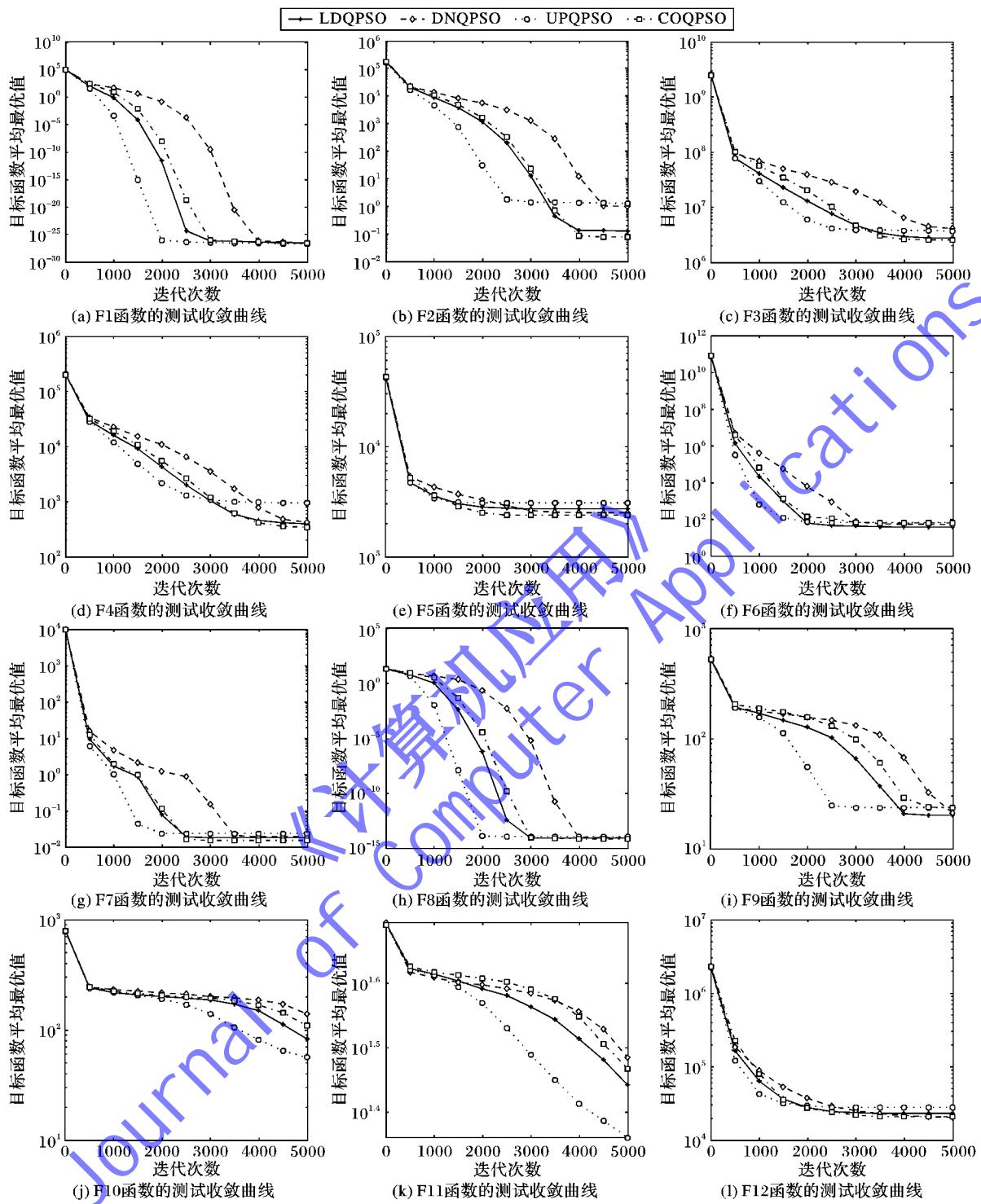


图1 4种算法优化不同标准测试函数的目标函数值的收敛曲线比较

表2中的COQPSO算法和其他几种算法的仿真结果比较显示,在测试函数F2、F3、F4、F5、F7、F8、F12中COQPSO算法取得了明显的优势,DNQPSO在F6、F9中获得了最好的仿真数值,而UPQPSO在F1、F10、F11测试中取得了最优的表现。在本文提出的COQPSO算法中,收缩扩张因子 β 采用的是1.0到0.5线性下降方法,和LDQPSO算法设置相同,因此两者更具有比较性。而通过对两者的仿真结果比较,COQPSO算法在12个测试函数的10个中取得了优势,只有在F6、F9函数中,劣于LDQPSO算法,说明交叉操作算法能够显著提高QPSO算法的性能。另外从图1中(a)~(l)的收敛曲线可以得知,仿真实验

中的几种算法在群体进化初期(500~800次迭代)收敛速度是相似的,没有明显的区别;在算法搜索的中后期,有了一些差别,对于COQPSO和LDQPSO基本保持了一致的收敛速度,但在最后的局部搜索过程中,COQPSO算法表现更加优异。以上说明,提出的具有交叉算子的QPSO算法大大提高了算法的整体性能,并且在大多数测试函数中表现了优异的性能。

4 结语

文章分析了QPSO算法的进化方程,研究了QPSO算法进化过程中采用了粒子的整体评价策略,算法在多维问题中

的局部寻优能力受到一定影响。在不增加算法复杂度的情况下,增加保留群体中单个粒子的维信息,提出了引入交叉算法的思想增强 QPSO 算法的寻优能力,并给出了改进后的算法执行过程。最后应用 CEC2005 测试函数对改进后的算法和

其他几种不同的控制参数选择策略进行了仿真比较,仿真结果显示改进后的算法在大部分测试函数上能够取得更优的结果,其他函数也能够取得和其他方法相近的结果,因此改进后的算法能够提高 QPSO 算法的整体性能。

表 2 COQPSO($\gamma=0.6$)与其他几种算法运行 50 轮后的平均最优值

算法	平均最优值											
	F1	F2	F3	F4	F5	F6	F7	F8	F9	F10	F11	F12
COQPSO ($\gamma=0.6$)	2.5139E - 27	0.07510	2.5781E + 06	344.0046	2.3794E + 03	60.5539	0.0155	7.3896E - 15	23.4980	108.8292	29.2903	2.0547E + 04
LDQPSO	2.9823E - 27	0.96404	4.1751E + 06	4.3066E + 02	2.5180E + 03	49.5838	0.0191	9.8054E - 15	21.0647	1.3977E + 02	30.4810	2.0659E + 04
DNQPSO	2.5252E - 27	0.12780	2.7708E + 06	3.9470E + 02	2.7399E + 03	37.1565	0.0184	8.1001E - 15	20.0990	82.9169	27.5797	2.3213E + 04
UPQPSO	2.0898E - 27	1.29110	3.7056E + 06	9.5279E + 02	3.0922E + 03	64.9420	0.0234	1.1510E - 14	23.6679	56.7124	22.9094	2.7928E + 04

参考文献:

- [1] KENNEDY J, EBERHART R. Particle swarm optimization [C]// Proceedings of the 1995 IEEE International Conference on Neural Network. Piscataway: IEEE, 1995: 1942 – 1948.
- [2] van den BERGH F. An analysis of particle swarm optimizers[D]. Pretoria: University of Pretoria, 2001.
- [3] ZHENG Y, MA L, ZHANG L. Empirical study of particle swarm optimizer with an increasing inertia weight [C]// Proceedings of the 2003 IEEE Congress on Evolutionary Computation. Piscataway: IEEE, 2003: 221 – 226.
- [4] CLERC M. Discrete particle swarm optimization, illustrated by the traveling salesman problem, new optimization techniques in engineering [M]. Berlin: Springer, 2004: 219 – 239.
- [5] SUN J, XU W, FENG B. A global search strategy of quantum-behaved particle swarm optimization [C]// Proceedings of the 2004 IEEE Conference on Cybernetics and Intelligent Systems. Piscataway: IEEE, 2004: 111 – 116.
- [6] SUN J, WU X, PLADE V, et al. Convergence analysis and improvements of quantum-behaved particle swarm optimization [J]. Information Sciences, 2012, 193: 81 – 103.
- [7] XI M, SUN J, XU W. An improved quantum-behaved particle swarm optimization algorithm with weighted mean best position [J]. Applied Mathematics and Computation, 2008, 205(2): 751 – 759.
- [8] SUN J, FANG W, XU W, et al. Quantum-behaved particle swarm optimization analysis of individual particle behavior and parameter selection [J]. Evolutionary Computation, 2012, 20(3): 349 – 39.
- [9] OMRANPOUR H, EBADZADEH M, SHIRY S, et al. Dynamic particle swarm optimization for multimodal function [J]. International Journal of Artificial Intelligence, 2012, 1(1): 1 – 10.
- [10] OMKAR S, KHANDELWAL R, ANANTH T, et al. Quantum behaved Particle Swarm Optimization (QPSO) for multi-objective design optimization of composite structures [J]. Expert Systems with Applications, 2009, 36(8): 11312 – 11322 .
- [11] INDIRAL K, KANMANI S, JAGAN R, et al. An evolutionary quantum behaved particle swarm optimization for mining association rules [J]. International Journal of Scientific and Engineering Research, 2014, 5(5): 379 – 388.
- [12] FARZI S, SHAVAZI A, PANDARI A. Using quantum-behaved particle swarm optimization for portfolio selection problem [J]. The International Arab Journal of Information Technology, 2013, 10 (2): 111 – 119.
- [13] CLERC M, KENNEDY J. The particle swarm: explosion, stability and convergence in a multi-dimensional complex space [J]. IEEE Transactions on Evolutionary Computation, 2002, 6: 58 – 73.
- [14] SUGANTHAN P, HANSEN N, LIANG J J, et al. Problem definitions and evaluation criteria for the CEC 2005 special session on real-parameter optimization [EB/OL]. [2014-08-16]. http://www.researchgate.net/publication/235710019_Problem_Definitions_and_Evaluation_Criteria_for_the_CEC_2005_Special_Session_on_Real_Parameter_Optimization.
- [15] FANG W. Swarm intelligence and its application in the optimal design of digital filters [D]. Wuxi: Jiangnan University, 2008. (方伟. 群体智能算法及其在数字滤波器优化设计中的研究[D]. 无锡: 江南大学, 2008.)

(上接第 674 页)

- [11] XU X, ZHENG K, LI D, et al. New chaos-particle swarm optimization algorithm [J]. Journal on Communications, 2012, 33(1): 25 – 30. (胥小波, 郑康锋, 李丹, 等. 新的混沌粒子群优化算法 [J]. 通信学报, 2012, 33(1): 25 – 30.)
- [12] TIAN D. Research of chaos particle swarm optimization algorithm [J]. Computer Engineering and Applications, 2013, 49(17): 43 – 46. (田东平. 混沌粒子群优化算法研究 [J]. 计算机工程与应用, 2013, 49(17): 43 – 47.)
- [13] LI P, WANG H, YANG Y. Random particle swarm optimization algorithm and its application [J]. Computer Systems and Applications, 2012, 21(2): 245 – 249. (李盼阳, 王海英, 杨雨. 一种随机粒子群算法及应用 [J]. 计算机系统应用, 2012, 21(2): 245 – 249.)
- [14] ZHANG L, ZHOU C, LIU X, et al. Application of particle swarm optimization for solving optimization problems [J]. Journal of Jilin University: Information Science Edition, 2005, 23(4): 385 – 388. (张利彪, 周春光, 刘小华, 等. 粒子群算法在求解优化问题中的应用 [J]. 吉林大学学报: 信息科学版, 2005, 23(4): 385 – 388.)
- [15] CHEN G, JIA J, HAN Q. Study on the strategy of decreasing inertia weight in particle swarm optimization algorithm [J]. Journal of Xi'an Jiaotong University, 2006, 40(1): 53 – 56. (陈贵敏, 贾建援, 韩琪. 粒子群优化算法的惯性权值递减策略研究 [J]. 西安交通大学学报, 2006, 40(1): 53 – 56.)
- [16] SHI Y, EBERHART R. A modified particle swarm optimizer [C]// Proceedings of the 1998 IEEE International Conference on Evolutionary Computation. Piscataway: IEEE, 1998: 69 – 73.
- [17] ZHAN Z, ZHANG J. Adaptive particle swarm optimization [C]// Proceedings of the 6th International Conference on Ant Colony Optimization and Swarm Intelligence. Berlin: Springer, 2008: 227 – 234.
- [18] HUANG X, ZHANG J, ZHAN Z. Faster particle swarm optimization with random inertia weight [J]. Computer Engineering and Design, 2009, 30(3): 647 – 650. (黄轩, 张军, 詹志辉. 基于随机惯量权重的快速粒子群优化算法 [J]. 计算机工程与设计, 2009, 30(3): 647 – 650.)
- [19] SUGANTHAN P N. Particle swarm optimizer with neighborhood operator [C]// Proceedings of the Congress on Evolutionary Computation. Piscataway: IEEE, 1999: 1958 – 1962.
- [20] RATNAWECRA A, HALGAMUGE S K, WATSON H C. Self-organizing hierarchical particle swarm optimizer with time-varying acceleration coefficients [J]. Evolutionary Computation, 2004, 8(3): 240 – 255.