

文章编号: 1001-0920(2011)04-0582-05

具有量子行为的协同粒子群优化算法

周 颀, 孙 俊, 须文波

(江南大学 信息工程学院, 江苏 无锡 214000)

摘 要: 以分布估计算法(EDA)的角度,从理论上指出,具有量子行为的粒子群优化算法(QPSO)本质上是EDA算法与原始粒子群算法(PSO)的综合.针对进化类算法普遍遇到的过早熟问题,将协同搜索策略引入传统的QPSO算法,提出了具有量子行为的协同粒子群优化算法(MQPSO).通过实验确定了最适合MQPSO算法的通信频率以及子种群大小.实验结果表明,该算法较QPSO及PSO算法具有更快的收敛速度和更强的搜索精度,其优势在高维优化问题中更为明显.

关键词: 分布估计算法; 具有量子行为的粒子群优化算法; 协同搜索策略; 通信频率; 子种群大小

中图分类号: TP301.6

文献标识码: A

Quantum-behaved particle swarm optimization algorithm with cooperative approach

ZHOU Di, SUN Jun, XU Wen-bo

(School of Information Technology, Jiangnan University, Wuxi 214000, China. Correspondent: ZHOU Di, E-mail: zhouprivate@msn.com)

Abstract: Quantum-behaved particle swarm optimization algorithm(QPSO) is investigated from the perspective of estimation of distribution algorithms(EDAs) for the first time, which proves that QPSO is a combination of EDAs and original particle swarm optimization. A quantum-behaved particle swarm optimization algorithm based on cooperative search strategy is presented, which helps prevent the evolutionary algorithms' universal tendency to be easily trapped into local optima as a result of the rapid decline in diversity. Communication frequency and the size of each sub-swarm are ensured through experiments to obtain the most effective setting for this algorithm. Experiment results show that this algorithm is able to find better solutions than the original QPSO and particle swarm optimization algorithm with higher efficiency.

Key words: estimation of distribution algorithm; quantum-behaved particle swarm optimization; cooperative search strategy; communication frequency; size of sub-swarm

1 引 言

粒子群算法是由Eberhart和Kennedy^[1]于1995年提出的.在标准粒子群算法(PSO)中,一个包含 M 个粒子的群体,每个粒子的维数为 D , $X_i(t) = (X_{i1}(t), X_{i2}(t), \dots, X_{iD}(t))$ 为粒子 i 的当前位置, $V_i(t) = (V_{i1}(t), V_{i2}(t), \dots, V_{iD}(t))$ 和 $p_{best_i} = (p_{best_{i1}}, p_{best_{i2}}, \dots, p_{best_{iD}})$ 分别为该粒子当前速度及个体最优位置, $g_{best} = (g_{best_1}, g_{best_2}, \dots, g_{best_D})$ 为所有粒子当前的全体最好位置.则PSO算法的进化公式可归结为如下形式:

$$V_{ij}(t+1) = \omega V_{ij}(t) + c_1 r_1 (p_{best_{ij}}(t) - X_{ij}(t)) +$$

$$c_2 r_2 (g_{best_j} - X_{ij}(t)),$$

$$X_{ij}(t+1) = X_{ij}(t) + V_{ij}(t+1). \quad (1)$$

其中: $i = 1, 2, \dots, M, j = 1, 2, \dots, D$; ω 为惯性因子,一般可设为随着进化而线性减少; c_1 和 c_2 为加速因子,通常在 $0 \sim 2$ 之间取值; $r_1, r_2 \sim U(0, 1)$.为了减少进化过程中微粒离开搜索空间的可能性,可以限定一个速度范围 $V_i \in [V_{min}, V_{max}]$.

PSO算法自提出以来,得到了广泛的研究并应用于大量实际工程中.然而文献[2]证明,PSO算法不能保证以概率1收敛到全局最优解,甚至不能收敛于局部最优解.对此,文献[3]从量子力学角度提出一

收稿日期: 2010-02-22; 修回日期: 2010-05-04.

基金项目: 国家自然科学基金项目(60474030).

作者简介: 周颀(1983-),女,博士生,从事智能算法及图像处理的研究; 须文波(1946-),男,教授,博士生导师,从事人工智能及计算机控制等研究.

种具有量子行为的粒子群优化算法(QPSO). 该算法以 δ 势阱为基础, 认为粒子具有量子行为, 目前已广泛地用于金融分析、图形处理、生物信息等领域, 并取得了较稳定的应用效果.

无论SPSO还是QPSO算法, 与其他进化算法一样, 在搜索过程中都容易因早熟而陷入局部最优. 这往往是由于迭代后期粒子群多样性过小所导致的. 协同搜索策略是解决该问题的一个有效方法. 协同搜索的主要思想是使用多个模块同时搜索问题空间, 这些模块之间互相交换信息, 以提高算法的效率, 它在研究大规模复杂优化问题中得到了广泛的应用. 本文从分布估计算法的角度分析了QPSO算法的本质, 通过引入协同搜索策略, 提出一种新型的具有量子行为的协同粒子群优化算法. 通过实验对算法中的两个重要参数: 通信频率以及子种群大小进行了讨论.

2 具有量子行为的粒子群算法与分布估计算法之间的关系

分布估计算法(EDA)是最近几年在进化计算领域内产生的一类新型优化算法, 它结合了统计学习理论与随机优化算法的优点. EDA建立了一种基于概率模型的学习和采样进化模式. 首先通过统计学习手段从群体宏观的角度建立一个描述解分布的概率模型; 然后对概率模型随机采样以产生新的种群. 如此反复, 直至满足终止条件, 以此实现种群的进化^[4-5]. 因为EDA算法是基于对整个群体建立数学模型, 直接描述了整个群体的进化趋势, 属于宏观层面的数学建模, 所以能更加有效地解决高维问题, 降低时间复杂性.

通过对SPSO算法的轨迹分析证明^[6], 为了保证算法的收敛性, 每个粒子必须收敛于各自的局部吸引子 $q_i = (q_{i1}, q_{i2}, \dots, q_{iD})$, $j = 1, 2, \dots, D$, 其中

$$q_{ij}(t) = \frac{c_1 r_1 p_{\text{best}_{ij}}(t) + c_2 r_2 g_{\text{best}_j}(t)}{c_1 r_1 + c_2 r_2}. \quad (2)$$

从动力学角度看, SPSO算法中粒子 i 的收敛过程是以 q_i 点为吸引中心, 随着速度的变小而不断地接近于 q_i 点, 直至最后收敛, 因此粒子群能保持聚集性. 但是在SPSO中, 粒子的收敛是以轨道形式实现的, 且粒子速度总是有限的, 因此进化过程中粒子的搜索空间是一个有限区域, 不能覆盖整个可行空间. 所以SPSO算法不能保证以概率1收敛到全局最优解, 甚至不能收敛于局部最优解. 而EDA算法中通过对概率模型采样生成新种群的方法, 可保证粒子依照概率分布于整个搜索空间, 具有更大的灵活性, 因此有可能找到具有更优适应值的位置.

将SPSO算法的发生环境转移到量子空间, 其中每个粒子的行为都将符合量子动力学; 即: 由波函数

描述粒子位置, 由薛定谔方程决定粒子的状态变化. 假设每个粒子都在一个以各自的局部吸引子 q_i 为中心的 δ 势阱中移动, 而粒子在各自 δ 势阱中的具体位置则由一个概率密度函数 $P(x)$ 决定. 进一步的实验表明, 当 $P(x)$ 为二次分布时算法具有最佳性能. 因此通过求解薛定谔方程, 可以得到粒子的概率分布函数为

$$D(x) = \int_{-\infty}^x P(t) dt = e^{-2|q-x|/L}. \quad (3)$$

对式(3)采用蒙特卡罗随机模拟的方式即可采样得到粒子的位置方程为

$$x = q \pm L/2 \ln(1/\mu), \quad \mu \sim U(0, 1). \quad (4)$$

从方程(4)可以看出, 粒子 x 的位置分布在以局部吸引子 q 为中心, L 为分布范围的区域内. 通过分析可知, 如果粒子的分布范围 L 过大, 则会导致粒子在空间中的分布过于分散, 从而降低整个粒子群的收敛速度, 甚至造成整个粒子群发散; 反之, 如果 L 过小, 又会导致粒子的分布每次都过于靠近其局部吸引子 q , 从而降低了粒子的随机性, 导致收敛速度过快而易于陷入局部最优值. 为了更好地调节 L 的取值, 文献[3]将一个称为“平均最优位置”的全局变量 m_{best} 引入传统的PSO算法, 规定 m_{best} 等于所有个体最优位置的平均值, 即

$$m_{\text{best}}(t) = \left(\frac{1}{M} \sum_{i=1}^M p_{\text{best}_{i1}}(t), \frac{1}{M} \sum_{i=1}^M p_{\text{best}_{i2}}(t), \dots, \frac{1}{M} \sum_{i=1}^M p_{\text{best}_{iD}}(t) \right). \quad (5)$$

每个粒子的分布范围被设为 $L_{ij} = 2\beta|m_{\text{best}_j}(t) - x_{ij}(t)|$, 这样可以保证远离群体中心的粒子搜索范围相对较大, 而靠近群体中心的粒子搜索范围较小. 因此粒子的位置方程可改写为

$$x(t+1) = q(t) \pm \beta|m_{\text{best}}(t) - x(t)| \ln(1/\mu), \quad (6)$$

其中 β 为收缩扩张因子, 用来控制整个算法的收敛速度. 文献[7]证明, 当 $\beta < 1.782$ 时, 算法可保证收敛. 以式(6)作为进化公式的SPSO算法即为具有量子行为的粒子群优化算法(QPSO).

通过上述分析可知, 在QPSO算法中, 每个粒子的每一维 x_{ij} 都服从一个以局部吸引子 q_{ij} 为中心, L_{ij} 为范围的二次分布. q_{ij} 直接决定了该粒子分布区域的位置, 而且每个 q_{ij} 都位于其个体最优位置与群体最优位置之间, 随迭代的进行逐渐靠近群体最优位置. SPSO的进化机制可以保证群体最优位置随迭代不断更新, 因此 q_{ij} 可指引粒子在保持随机性的同时逐步向全局最优值靠拢. L_{ij} 起到了概率分布中类似于“方差”的作用, 决定了粒子的分布范围, 且该范围由该粒子在群体中的位置决定. 粒子越接近于群体

的外围,其搜索范围越大;反之,粒子越接近于群体中心,其搜索范围越小.因此,从本质上来说,QPSO算法与SPSO算法的区别在于后者采用了“速度+位置”的轨道方式生成新群体;而前者则借鉴群体中所有粒子的位置建立了分布概率模型,通过随机采样完成了对群体的更新.因此QPSO算法可以理解为一种将分布估计思想装入PSO框架中的综合型算法.这也正是QPSO算法搜索性能优于SPSO算法及很多遗传算法的原因.

3 具有量子行为的协同粒子群优化算法

3.1 协同搜索策略

协同进化算法是一类模拟生物协同进化机制的优化算法,尤其适合于多目标、复杂的动态大规模优化问题.其核心思想是先将复杂问题分解成多个简单的子问题,然后采用不同的进化算法求解每个子问题,最后从整体上进行协调与综合.子问题的求解与整体综合交替进行,直至找到最优解.按照协同系统所达到的空间分解层次进行分类,协同搜索算法可以分为隐式空间分解和显式空间分解^[10].为了解决QPSO算法在迭代后期存在种群多样性过低而导致早熟的问题,将协同搜索策略引入QPSO算法,提出一种具有量子行为的协同粒子群优化算法(MQPSO).将整个粒子群划分为 N 个子种群,采用隐式空间分解的协同策略,让这 N 个子种群各自独立地采用QPSO算法进行进化,并以某一频率共享各自有用的信息.

由上节讨论可知,局部吸引子 q 和分布范围 L 对整个QPSO算法的性能起着重要作用.因为 $q_i = \phi p_{best_i} + (1 - \phi)g_{best}$, $L_i = 2\beta|m_{best} - x_i|$,所以在各子种群之间通信时,可将每个子种群的 g_{best} 和 m_{best} 同时进行共享,并将各子种群的 g_{best} 和 m_{best} 都调整为与具有最好 g_{best} 的子种群一样的值.所有子种群之间的通信频率可由一个固定的、以迭代次数 t 为自变量的概率控制函数 $F(t)$ 进行调节, $F(t) \in [0, 1]$.在每次迭代开始之前都要判断一下各子种群间是否需要通信.取随机数 $\text{rand}(t) \sim U(0, 1)$,当 $\text{rand}(t) \leq F(t)$ 时,共享 g_{best} 及 m_{best} ;否则,各子种群间不通信,而按原始QPSO进化规则各自进行进化.因此,

MQPSO算法可描述如下:

Step 1: 给每个子种群的所有粒子随机赋初始位置,并求出每个子种群的 g_{best} , $\text{mark} = 0$.

Step 2: 根据概率控制函数 $F(t)$ 计算通信概率 $\text{exchange_possible} = F(t)$.

Step 3: 如果 $\text{rand}(0, 1) \leq \text{exchange_possible}$,则选出具有最好 g_{best} 的子种群 l ,并将其他子种群的 $m_{best_i} = m_{best_l}$, $g_{best_i} = g_{best_l}$,并且 $\text{mark} = 1$;否则, $\text{mark} = 0$.

Step 4: 若 $\text{mark} == 0$,则所有子种群按式(5)计算 m_{best} ;若 $\text{mark} == 1$,则不再重新计算 m_{best} .

Step 5: 根据式(6)计算粒子的新位置,并更新 p_{best} 和 g_{best} .

Step 6: 返回Step 2,直至迭代完成.

Step 7: 选出所有子种群中最好的 g_{best} 并输出.

在所有基于协同搜索的优化算法中,各子种群之间的通信频率以及子种群大小是影响整个算法性能优劣的重要因素.如果通信频率过高,则会导致整个粒子群的多样性迅速降低,从而较易于陷入局部最优;如果通信频率过低,则会使各子种群合作的优势不明显.对于子种群大小而言,如果每个子种群中包含的粒子数过少,则会影响各子种群的搜索能力;如果过多,则会导致整个算法开销过大,无法体现协同搜索算法的优点.因此应采用实验方法确定出最适合MQPSO算法的通信频率控制函数以及子种群大小.

3.2 通信频率控制函数

通信频率控制函数 $F(t)$ 一般可设为随迭代次数 t 单调递增或单调递减.令 $A(t)$ 代表递增函数 $A(t) = t/\text{maxiter}$,其中 maxiter 表示最大迭代次数;令 $B(t)$ 代表递减函数 $B(t) = (\text{maxiter} - t)/\text{maxiter}$.它们的幂函数递增递减性质不变.现设置一个包含4个种群的粒子群,分别采用 $y = A(t)^k$ 和 $y = B(t)^k$ ($k = 0.1, 0.2, 0.4, 0.6, 0.8, 1, 2, 4, 6, 8, 10, 12, 14, 16$)作为子种群间通信频率控制函数,对取自标准测试函数集^[8]的4个函数进行测试.这4个函数的初始化范围、搜索范围、理论最优值以及函数类型如表1所示.

表1 标准测试函数

测试函数	函数表达式	搜索范围	初始化范围	函数理论最优值	函数类型
Unimodal Schwefel	$f_1(x) = 10^6 x_1^2 + \sum_{i=2}^n (x_i^2)^2$	$[-100, 100]$	$[-50, 100]$	$f_1(x) = 0, x(i) = 0$	单峰
Rosenbrock	$f_2(x) = \sum_{i=1}^{n-1} (100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2)$	$[-30, 30]$	$[15, 30]$	$f_1(x) = 0, x(i) = 0$	单峰
Griewank	$f_3(x) = \frac{1}{4000} \sum_{i=1}^n x_i^2 - \prod_{i=1}^n \cos(2\pi x_i) + 10$	$[-600, 600]$	$[300, 600]$	$f_1(x) = 0, x(i) = 0$	多峰
Multimodal Schwefel	$f_4(x) = 418.9829n - \sum_{i=1}^n (x_i \sin \sqrt{ x_i })$	$[-500, 500]$	$[250, 500]$	$f_1(x) = 0, x(i) = 420.9687$	多峰

实验中每个子种群包含 20 个粒子, 每个粒子维数均为 10, 迭代次数上限为 500 次. β 随迭代次数增加在 1.0 ~ 0.5 之间线性减小^[3]. 实验环境为 Visual C++6.0, Intel Core2 Duo 2.00 GHz. 对各测试函数采用不同通信频率控制函数而产生的测试结果用 3 次样条插值显示, 见图 1. 图 1 的横坐标表示幂指数 k ; 由于 f_1 的测试结果大部分在 $10^{-30} \sim 10^{-60}$ 之间, 为了清晰起见, 图 1(a) 和图 1(b) 中的纵坐标表示函数平均最优值的数量级; 图 1(c)~图 1(h) 的纵坐标表示函数平均最优值.

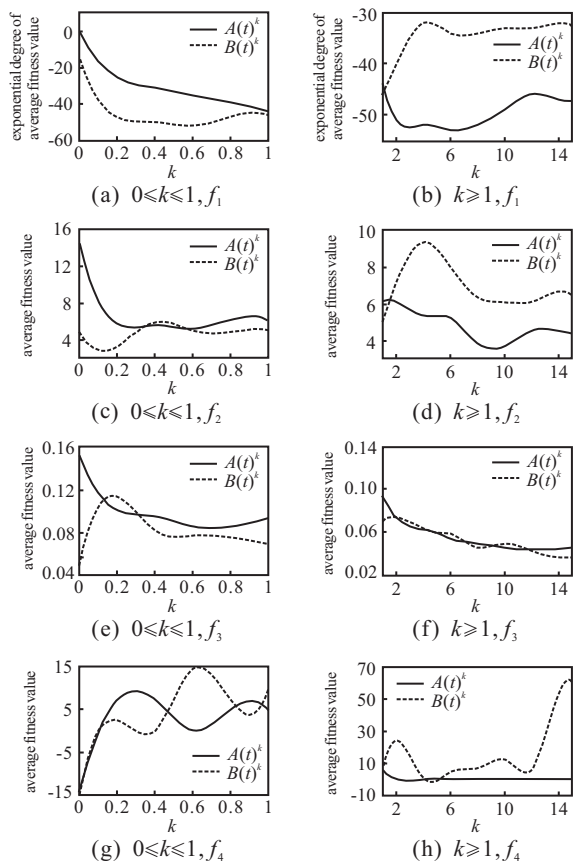


图 1 通信频率控制函数中 k 的变化对 MQPSO 性能的影响

由图 1 可知, 对于单峰函数, 在 $k \geq 1$ 的情况下, 用递增类函数 $A(t)$ 作为通信频率控制函数所得的函数最优值优于递减类函数 $B(t)$; 当 $k \in [0, 1]$ 之间时, 情况正好相反. 对于多峰函数, 无论采用哪类通信频率控制函数, $k \in [1, +\infty]$ 时的测试结果均优于 $k \in [0, 1]$ 时. 综合所有测试函数, 采用 $y = (t/\maxiter)^8$ 作为通信频率控制函数, 能使 MQPSO 算法具有最好的搜索精度和稳定性.

3.3 子种群大小

为了确定最佳的子种群大小, 设定一个粒子数为 80 的粒子群, 将之分别划分为 2 个包含 40 个粒子的子种群 (MQPSO_40), 4 个包含 20 个粒子的子种群 (MQPSO_20), 8 个包含 10 个粒子的子种群 (MQPSO_

10), 16 个包含 5 个粒子的子种群 (MQPSO_5), 仍然对表 1 中的 4 个测试函数进行测试. 同时也对标准 QPSO 算法和 SPSO 算法进行对比测试, 其中粒子群同样包含 80 个粒子. 测试时, $\maxiter = \text{dimension} \times 100$. SPSO 算法中, ω 从 0.9 ~ 0.4 线性减少^[9]. QPSO 算法与 MQPSO 算法参数的设置均与第 3.2 节相同. 采用 $y = (t/\maxiter)^8$ 控制 MQPSO 算法的通信频率. 分别记录维数为 10, 30 和 50 时, 运行 50 次得到的平均最优值以及运行 50 次的总耗时 (单位 ms), 并列于表 2 中. 维数为 30 时, 各算法的收敛曲线如图 2 所示.

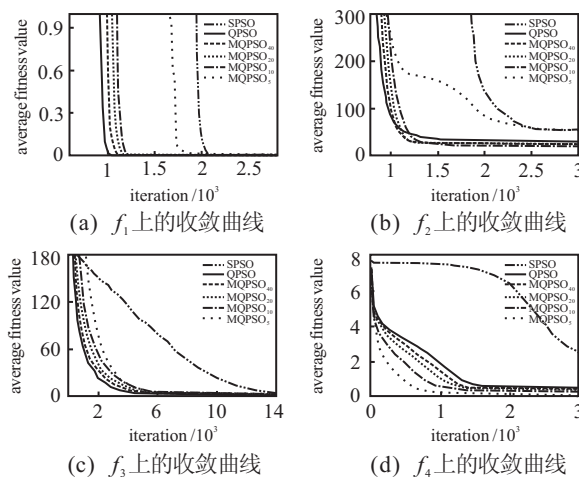


图 2 各算法收敛曲线

由表 2 可知, 对于单峰函数, MQPSO 的搜索精度随子种群中粒子数减少而下降; 而对于多峰函数, MQPSO 的搜索精度则随子种群粒子数减少而略有增强. 这说明多个子种群协同搜索有效防止了整个群体多样性的过快下降, 从而减少了陷入局部最优值的可能性. 各类 MQPSO 算法的搜索精度大都强于 QPSO 算法, 仅对维数为 10 的函数 f_1 进行测试时, QPSO 搜索得到的函数最优值好于 MQPSO 算法, 说明 MQPSO 算法的优势在高维优化问题上更为明显. 另外, SPSO 算法耗时最少, QPSO 算法耗时最多, 这是因为 Visual Studio C++ 中运行随机数函数消耗了大量时间. 各类 MQPSO 算法的耗时介于 SPSO 与 QPSO 之间, 且随子种群粒子数减少而增加.

由图 2 的收敛曲线可以看出, 在大部分情况下, SPSO 的收敛速度最慢, 而 QPSO 速度最快. 各 MQPSO 算法的收敛速度随子种群粒子数减少而依次下降. 据分析所知, 对于算法而言, 如果收敛速度过慢, 则该算法的搜索效率相对较低; 如果收敛速度过快, 则该算法陷入局部最优的概率较高. 因此, MQPSO 之所以能找到比 QPSO 和 SPSO 更接近函数理想最优值的原因在于它具有合适的收敛速度.

表 2 函数测试结果

函数	粒子维数		算 法					
			SPSO	QPSO	MQPSO_40	MQPSO_20	MQPSO_10	MQPSO_5
f_1	10	均值	1.387e-40	3.471e-131	6.556e-125	1.593e-109	2.325e-83	7.871e-50
		耗时	4 797	8 938	8 250	8 390	8 296	8 875
	30	均值	5.217e-25	5.643e-95	1.944 6e-114	1.110e-94	1.502e-73	3.049e-50
		耗时	41 544	61 187	70 665	72 421	72 188	73 594
	50	均值	4.631e-16	5.339e-69	3.109e-93	1.748e-82	2.857e-66	1.304e-46
		耗时	111 603	206 288	196 287	197 618	197 968	201 212
f_2	10	均值	4.670	4.061	3.236	3.096	2.327	3.314
		耗时	4906	8 922	8 375	8 422	8 422	8 609
	30	均值	42.837	38.349	31.809	23.912	22.572	40.826
		耗时	44 298	63 031	73 555	75 585	74 785	75 703
	50	均值	80.658	51.672	46.035	45.214	67.297	104.319
		耗时	119 885	214 570	205 665	206 290	206 328	209 608
f_3	10	均值	0.083 92	0.040 63	0.038 30	0.037 47	0.036 24	0.028 66
		耗时	8 953	12 140	12 094	12 250	12 141	12 438
	30	均值	0.013 49	9.654e-3	3.489e-3	1.226e-3	2.464e-4	7.660e-4
		耗时	79 121	87 969	107 070	108 868	108 360	109 531
	50	均值	8.493e-3	2.341e-3	3.698e-4	8.93e-5	0	0
		耗时	217 305	298 008	299 493	301 523	301 563	304 063
f_4	10	均值	245.966	10.659	0.000 127	0.000 127	0.000 127	0.000 127
		耗时	10 063	13 766	12 594	12 813	12 687	12 985
	30	均值	2 471.491	518.171	484.016	464.473	321.765 9	88.862 9
		耗时	86 796	96 156	113 223	111 855	111 856	112 461
	50	均值	5 269.350	1 349.740	1 333.493	1 326.042	1 159.084	1 185.467
		耗时	237 188	324 140	307 850	307 890	309 650	311 250

4 结 论

本文以分布估计算法的角度,从理论上详细分析了QPSO算法的原理及其优于原始粒子群算法及许多进化算法的原因.将协同搜索策略引入传统的QPSO算法并用于解决过早熟问题,提出了MQPSO算法.实验表明,当以 $y = (t/\maxiter)^8$ 作为通信频率控制函数、每个子种群大小控制为10个粒子时,MQPSO算法具有最好的搜索性能和稳定性,远远优于原始QPSO算法及SPSO算法.其优势在高维优化问题中更为明显.

参考文献(References)

- [1] Kennedy J, Eberhart R C. Particle swarm optimization[C]. Proc of IEEE Int Conf on Neural Networks. Perth, 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 Jun, Xu Wenbo, Feng Bin. A global search strategy of quantum-behaved particle swarm optimization[C]. Proc of IEEE Conf on Cybernetics and Intelligent Systems. Singapore: IEEE Press, 2004: 111-116.
- [4] Larrañaga P, Lozano J A. Estimation of distribution algorithms: A new tool for evolutionary computation[M]. Boston: Kluwer Academic Publishers, 2002.
- [5] Pelikan M, Goldberg D E, Lobo F. A survey of optimization by building and using probabilistic models[J]. Computational Optimization and Applications, 2002, 21(1): 5-20.
- [6] Clerc M, Kennedy J. The particle swarm: Explosion, stability and convergence in a multi dimensional complex space[J]. IEEE Trans on Evolutionary Computation, 2002, 6(1): 58-73.
- [7] Sun Jun, Xu Wenbo, Feng Bin. Adaptive parameter control for quantum-behaved particle swarm optimization on individual level[C]. Proc of IEEE Int Conf on Systems, Man and Cybernetics. Hawaii: IEEE Press, 2005: 3049-3054.
- [8] Yao X, Liu Y, Lin G. Evolutionary programming made faster[J]. IEEE Trans on Evolutionary Computation, 1999, 3(2): 82-102.
- [9] Shi Y, Eberhart R C. A modified particle swarm[C]. Proc of IEEE Int Conf on Evolutionary Computation. Alaska: IEEE Press, 1998: 1945-1950.
- [10] 周树德, 孙增圻. 分布估计算法综述[J]. 自动化学报, 2007, 33(2): 113-124.
(Zhou S D, Sun Z Q. A survey on estimation of distribution algorithms[J]. Acta Automatica Sinica, 2007, 33(2): 113-124.)