

文章编号: 1001-0920(2012)09-1406-05

## 拥挤距离排序的多目标文化粒子群优化算法

李中凯, 李艾民, 朱真才

(中国矿业大学 机电工程学院, 江苏 徐州 221116)

**摘要:** 为解决工程优化设计问题, 引入文化进化框架, 提出一种拥挤距离排序的多目标文化粒子群算法. 采用拥挤距离排序算子, 并删除密集区域的多余粒子, 以保证 Pareto 前沿的分布均匀性; 基于拥挤距离值, 从精英知识和条件知识中选择处于最分散区域的粒子, 并将其分别作为全局和局部最优, 以增强算法全局寻优能力; 依据拥挤距离的变化, 动态调整粒子群飞行参数, 以提高算法收敛效率. 通过标准测试问题以及与其他算法的对比, 表明了所提出算法的有效性和鲁棒性.

**关键词:** 文化粒子群算法; 多目标进化算法; 自适应参数调整; 拥挤距离

**中图分类号:** TP18

**文献标志码:** A

## Cultural based multi-objective particle swarm optimization algorithm using crowding distance sorting method

LI Zhong-kai, LI Ai-min, ZHU Zhen-cai

(School of Mechatronics Engineering, China University of Mining and Technology, Xuzhou 221116, China.

Correspondent: LI Zhong-kai, E-mail: lizk@cumt.edu.cn)

**Abstract:** To solve the engineering design problems, by introducing cultural evolution framework, a cultural based multi-objective particle swarm optimization algorithm with crowding distance sorting is proposed. The redundant particles in the crowded area are deleted with the distance sorting operator to guarantee the elitism's uniform distribution. With the distance value, the global and local best of the particles are selected from the most disperse region in the elitism and situational knowledge, respectively, so as to enhance its global searching capability. The evolution parameters are adjusted dynamically according to the changing of distance to improve the convergence speed. Some standard test problems and the comparison with other algorithms show the effectiveness and robustness of the algorithm.

**Key words:** cultural based particle swarm optimization; multi-objective evolutionary algorithm; paramters self-adaptive adjustment; crowding distance

### 1 引言

现实设计中的优化问题通常包括多个相互冲突的优化目标, 若要满足多约束条件和多设计变量的取值范围, 需采用多目标进化算法<sup>[1]</sup>求得 Pareto 前沿. 粒子群算法 (PSO)<sup>[2]</sup>是一种群体智能算法, 将 PSO 扩展为多目标粒子群算法 (MOPSO) 需解决的主要问题是: 选择粒子局部最优值和群体全局最优值, 以指导粒子群向 Pareto 前沿进化; 同时, 控制优势种群中粒子的拥挤程度, 以获得分布均匀的 Pareto 前沿.

文献 [3] 提出了快速非支配排序 MOPSO, 通过比较粒子当前位置与局部最优位置的支配关系来保证

粒子向 Pareto 前沿进化. 但是, 由于丢失了粒子局部优势信息, 导致最优前沿分布性不足. [4] 提出了拥挤距离计算方法, 将边界个体距离值赋为无穷大来计算内部个体拥挤距离, 可以快速判断个体之间的拥挤程度. [5] 基于拥挤距离排序算子开发了多目标粒子群算法, [6] 开发了量子粒子群算法, 但都存在进化参数无法自适应调整的问题. [7] 提出了基于文化框架的 MOPSO, 使用提取的条件知识、标准知识和拓扑知识来调整粒子飞行参数, 采用网格表示粒子拥挤程度, 但网格划分与调整难度较大. 文化算法是一种模拟社会文化发展过程的计算框架, 在种群空间进

收稿日期: 2011-02-25; 修回日期: 2011-06-16.

基金项目: 国家自然科学基金项目(51005237); 中国博士后科学基金项目(20100471407); 江苏高校优势学科建设工程项目(PAPD).

作者简介: 李中凯(1980—), 男, 副教授, 博士, 从事智能计算方法、产品设计方法等研究; 李艾民(1965—), 男, 教授, 从事现代机械设计理论与方法的研究.

行进化计算,在信念空间进行优势知识积累,以实现进化过程的自适应调整和全局寻优.本文将粒子群算法、文化进化框架与拥挤距离算子相融合,提出一种拥挤距离排序的多目标文化粒子群算法,以保证算法的收敛性和多样性特征.

## 2 文化粒子群进化模型

将PSO的快速收敛特性与文化算法的自适应调节能力相结合来建立多目标文化粒子群算法框架,如图1所示.种群空间进行粒子群飞行运算,选择优势粒子更新文化空间.文化空间包括精英知识( $E$ ),条件知识( $S$ )和标准知识( $N$ ).其中: $E$ 存储粒子群进化中产生的所有优势粒子,进而将其作为最终的Pareto前沿,并从中选择全局最优位置; $S$ 存储每个粒子进化过程中的所有优势位置,从而选择处于最分散区域的局部最优; $N$ 存储粒子群飞行速度的最大和最小边界,以动态调整粒子速度惯量.两空间的交互方式如下:种群空间通过优势粒子更新文化空间,而文化空间则通过调节函数更新粒子群飞行参数,从而提高算法的收敛效率和全局寻优能力.

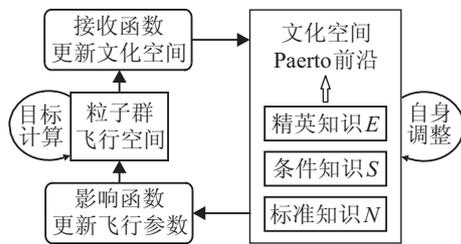


图1 文化粒子群进化框架示意图

## 3 多目标文化粒子群算法

### 3.1 总体运算流程

基于文化粒子群的进化模型,采用拥挤距离排序算子指导精英知识的筛选和粒子最优值选择.拥挤距离排序多目标文化粒子群算法(Cultural MOPSO)的运算步骤描述如下:

Step 1: 初始化粒子群空间和文化空间.

Step 2: 计算内部粒子群的优化目标值.

Step 3: 使用约束支配关系<sup>[4]</sup>标记内部粒子群的支配情况,选择非支配的优势粒子.

Step 4: 使用优势粒子分别更新信念空间中的精英知识、条件知识和标准知识.

Step 5: 进行内部粒子群劣势粒子的变异运算.

Step 6: 基于拥挤距离排序更新粒子群的全局最优值.

Step 7: 基于拥挤距离排序更新每个粒子的局部最优值.

Step 8: 基于拥挤距离和标准知识调整粒子群飞行参数,包括全局飞行系数 $c_g$ ,局部飞行系数 $c_p$ 和粒

子速度惯量 $\omega$ .

Step 9: 进行粒子群的一代飞行运算.

Step 10: 若达到最大迭代次数,则输出精英知识作为最终解;否则,返回Step 2继续运算.

### 3.2 文化空间的更新方法

#### 3.2.1 精英知识的更新策略

设第 $t$ 代精英知识 $E_t$ 包含 $m_1$ 个粒子,最大允许的粒子数为 $p^{\max}$  $(m_1 \leq p^{\max})$ .内部粒子群进化后产生 $n_1$ 个优势粒子,保存在集合 $Q_{t+1}$ 中.合并集合 $E_t$ 和 $Q_{t+1}$ ,形成精英知识 $E'_t$ .

标记并删除 $E'_t$ 中的重复和劣势粒子,此时 $E'_t$ 包含 $m_2$  $(m_2 \leq m_1 + n_1)$ 个互不支配的粒子.计算 $E'_t$ 中所有粒子的拥挤距离,并按降序排列形成精英知识 $E''_t$ .

判断 $m_2$ 与 $p^{\max}$ 的数值关系.若 $m_2 \leq p^{\max}$ ,则将 $E''_t$ 记为新精英知识 $E_{t+1}$ ,此时 $E_{t+1}$ 的后 $p^{\max} - m_2$ 个位置为空;否则,调用精英知识的缩减过程,即仅保留 $E''_t$ 中的前 $p^{\max}$ 个,删除最后 $m_2 - p^{\max}$ 个拥挤距离小的粒子,形成缩减的精英知识 $E_{t+1}$ .

#### 3.2.2 条件知识的更新策略

文化粒子群算法设置条件知识,存储单粒子进化过程中产生的所有非支配位置,可以扩展粒子局部最优的范围,选择处于分散区域的局部最优,以增强算法全局寻优能力.条件知识的更新策略是,首先将粒子的当前位置拷贝入条件知识库,进而标记该条件知识库的粒子支配情况,并排除劣势粒子;然后计算粒子拥挤距离并按降序排列,支持后续粒子局部最优值的更新.由于每个粒子条件知识库的大小没有限制,随着进化运算的进行,粒子条件知识会逐渐增多.

#### 3.2.3 标准知识的更新策略

在标准知识 $N$ 中,设置 $vel_j^l$ 和 $vel_j^u$ 分别记录第 $j$ 个变量的速度最小值和最大值.标准知识的更新采用下式:

$$vel_j^l(t+1) = \begin{cases} v_{i,j}(t), & v_{i,j}(t) < vel_j^l(t); \\ vel_j^l(t), & \text{otherwise.} \end{cases} \quad (1)$$

$$vel_j^u(t+1) = \begin{cases} v_{i,j}(t), & v_{i,j}(t) > vel_j^u(t); \\ vel_j^u(t), & \text{otherwise.} \end{cases} \quad (2)$$

其中: $i=1,2,\dots,m$ ;  $j=1,2,\dots,n$ ;  $m$ 为内部种群粒子数,  $n$ 为设计变量数;  $v_{i,j}(t)$ 为第 $i$ 个粒子的第 $j$ 个变量在第 $t$ 代循环的速度值.

### 3.3 全局和局部最优的选择

在文化粒子群算法Step 6和Step 7中,采用拥挤距离排序方法更新粒子群的全局和局部最优.由于二者更新方法类似,下面以全局最优值 $g_{\text{best}}$ 的更新为

例进行阐述.

设精英知识  $E$  中粒子已按拥挤距离降序排列. 若  $E$  中所有粒子的拥挤距离均为无穷大 ( $CD = INF$ ), 即 Pareto 前沿中仅包括数量较少的边界粒子, 则在边界粒子中随机选择一个作为  $g_{best}$ ; 若  $E$  中包括拥挤距离不为  $INF$  的粒子, 则选择一个拥挤距离最大且不为  $INF$  的粒子作为  $g_{best}$ . 因此选出的  $g_{best}$  处于 Pareto 前沿的最分散区域, 有助于扩大粒子群的全局寻优范围.

### 3.4 粒子群进化参数调整

#### 3.4.1 飞行系数调整

对于全局飞行系数  $c_g$ , 如果新  $g_{best}$  的拥挤距离大于当前  $g_{best}$  的距离值, 则说明最优粒子向分散区域飞行, 作为奖励应增大  $c_g$  值. 如果拥挤距离变小, 则说明粒子飞向拥挤区域, 作为惩罚应减小  $c_g$  值; 否则,  $c_g$  值不变. 粒子全局飞行系数也应保持在  $c_g^{\min}$  与  $c_g^{\max}$  之间. 因此,  $c_g$  的更新公式为

$$c_g(t+1) = \begin{cases} c_g(t) + \alpha \cdot |cd_g(t) - cd_g(t+1)|, & cd_g(t) < cd_g(t+1); \\ c_g(t) - \alpha \cdot |cd_g(t) - cd_g(t+1)|, & cd_g(t) > cd_g(t+1); \\ c_g(t), & \text{otherwise.} \end{cases} \quad (3)$$

$$c'_g(t+1) = \begin{cases} c_g^{\max}, & c_g(t+1) > c_g^{\max}; \\ c_g^{\min}, & c_g(t+1) < c_g^{\min}; \\ c_g(t+1), & \text{otherwise.} \end{cases} \quad (4)$$

其中:  $cd_g(t)$  为全局最优粒子第  $t$  代运算拥挤距离值,  $\alpha$  为调节常量. 对于局部飞行系数  $c_{p,i}$ ,  $i = 1, 2, \dots, m$ , 其调节原理与全局飞行系数  $c_g$  相同, 调节系数为  $\beta$ .

由于  $c_g$  和  $c_{p,i}$  值介于  $[1, 3]$  区间, 拥挤距离之差的绝对值介于  $[0, 1]$  区间, 为保证良好的进化收敛效果, 通过运算实验, 可取  $\alpha = \beta = 0.01$ .

#### 3.4.2 速度惯量调整

使用标准知识调整每个粒子每个变量的飞行速度惯量  $\omega_{i,j}$ , 其中  $i = 1, 2, \dots, m, j = 1, 2, \dots, n$ , 同时保持粒子速度惯量在  $[\omega^{\min}, \omega^{\max}]$  之间. 取  $\Delta\omega$  为调节系数,  $\omega_{i,j}$  的更新算式为

$$\omega_{i,j}(t+1) = \begin{cases} \omega_{i,j}(t) + \Delta\omega, & v_{i,j}(t) < vel_j^l(t); \\ \omega_{i,j}(t) - \Delta\omega, & v_{i,j}(t) > vel_j^u(t); \\ \omega_{i,j}(t), & \text{otherwise.} \end{cases} \quad (5)$$

$$\omega'_{i,j}(t+1) = \begin{cases} \omega^{\max}, & \omega_{i,j}(t+1) > \omega^{\max}; \\ \omega^{\min}, & \omega_{i,j}(t+1) < \omega^{\min}; \\ \omega_{i,j}(t+1), & \text{otherwise.} \end{cases} \quad (6)$$

由于  $\omega$  的取值范围位于  $(0, 1)$  区间内,  $\Delta\omega$  取值过大, 会对算法产生振动影响. 依据运算实验, 可取  $\Delta\omega$  为较小值, 如  $\Delta\omega = 0.1$ .

### 3.5 内部粒子群变异机制

在粒子群算法中加入变异机制, 对粒子位置产生小范围扰动, 有助于增强算法的全局寻优能力. 但是, 固定的变异概率会随进化运算增多对粒子群产生振荡影响, 变异概率应随着迭代次数的增加逐步减小. 因此, 设置内部粒子群劣势个体的变异概率  $MP$  为

$$MP = (1 - t/G^{\max})^{5/u}. \quad (7)$$

其中:  $t$  为当前迭代次数,  $G^{\max}$  为算法最大迭代次数,  $u$  为预设的变异常数. 对于浮点数粒子编码方式, 采用多项式规则<sup>[8]</sup>进行变异运算.

### 3.6 算法的时间与空间复杂度

Cultural MOPSO 的主要运算工作在于粒子拥挤距离计算, 因此算法的时间复杂度为  $O(N \log N)$ ,  $N$  为内部粒子群数量, 低于传统多目标进化算法的  $O(N^2)$  或  $O(N^3)$  时间复杂度, 显著提高了多目标进化的运算效率. 同时, 文化粒子群算法由于需存储粒子群精英知识和每个粒子的条件知识等信息, 空间复杂度比单目标粒子群有所增加, 但没有影响算法的收敛速度.

## 4 实验对比与分析

### 4.1 优化测试问题求解

将所提出的算法与非支配排序多目标遗传算法 NSGA-II<sup>[4]</sup> 和拥挤距离排序多目标粒子群算法 DSMOPSO<sup>[5]</sup> 进行对比. 使用 C 语言编程实现 3 种算法, 运行于双核 2.6 GHz, 2 G 内存微型 PC. 利用趋近前沿标准  $C$ <sup>[9]</sup> 和多样性标准  $\delta$ <sup>[4]</sup> 比较各算法获得 Pareto 前沿的收敛性和多样性特征.

由于对设计约束的处理采用了 Pareto 约束支配方式, 无需额外参数便可将搜索范围限制在可行空间内. 约束问题的优化过程与无约束问题类似, 下面分别测试 3 种算法对于两目标和三目标无约束优化问题求解的效果.

Cultural MOPSO 的运算参数为:  $\omega^{\min} = 0.1, \omega^{\max} = 0.9, c_g^{\min} = 1, c_g^{\max} = 3, c_p^{\min} = 1, c_p^{\max} = 3, \omega^{\text{ini}} = 0.4, c_g^{\text{ini}} = c_p^{\text{ini}} = 2, \eta_m = 20, \Delta\omega = 0.1, \alpha = \beta = 0.01, u = 10$ ; NSGA-II 的运行参数为:  $p_c = 0.9, p_m = 0.01, \eta_c = \eta_m = 20$ ; DSMOPSO 的运行参数为:  $\omega = 0.3, c_g = c_p = 2, p_m = 0.1$ . 3 种算法的内部种群数为  $m = 100, G^{\max} = 200$ , 两种粒子群算法的精英种群数量为  $p^{\max} = 100$ .

选择两目标测试问题 ZDT3<sup>[4]</sup> 和三目标测试问题 DTLZ1<sup>[9]</sup> 进行算法求解效果的对比. ZDT3 具有 30 个设计变量, DTLZ1 包括 7 个变量. 分别采用 Cultural MOPSO, NSGA-II 和 DSMOPSO 求解上述 2 个问题得到的最优前沿如图 2 和图 3 所示. 每个算法运算 20 次取平均值, 消除初始化随机扰动因素的影响后得到的趋近前沿  $C$  与多样性  $\delta$  的比较分别如表 1 和表 2 所示.

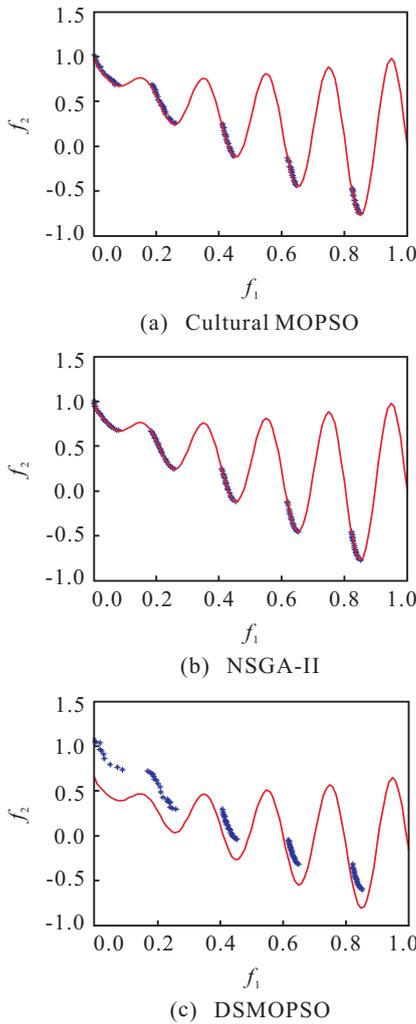


图2 测试问题 ZDT3 的 Pareto 前沿对比

表1 测试问题的趋近前沿 C 比较结果

趋近指标	ZDT3	DTLZ1
$C$ (Cultural MOPSO,NSGA-II)	0.770 0	1
$C$ (NSGA-II,Cultural MOPSO)	0.010 0	0
$C$ (Cultural MOPSO,DSMOPSO)	1	1
$C$ (DSMOPSO,Cultural MOPSO)	0	0

表2 测试问题的多样性  $\delta$  比较结果

算法	ZDT3	DTLZ1
Cultural MOPSO	0.558 7	0.616 0
NSGA-II	0.630 2	0.820 7
DSMOPSO	0.752 5	1.081 8

从图2可看出,对于ZDT3问题,Cultural MOPSO,NSGA-II都接近真实前沿,DSMOPSO的前沿偏离度较大,且分布不均匀.如表1所示,ZDT3的 $C$ (Cultural MOPSO,DSMOPSO)指数为0.77,表明文化粒子群的Pareto前沿中有77%的个体优于NSGA-II的Pareto前沿,而 $C$ (NSGA-II,Cultural MOPSO)=0.01则表示NSGA-II的解仅有1%优于Cultural MOPSO,表明对于ZDT3问题,Cultural MOPSO比NSGA-II具有更好的收敛性.在ZDT3的Pareto前沿分布方面,Cultural MOPSO的 $\delta$ 值比NSGA-II减小12.8%,比DSMOPSO减小16.3%,具有更好的分布均匀性.

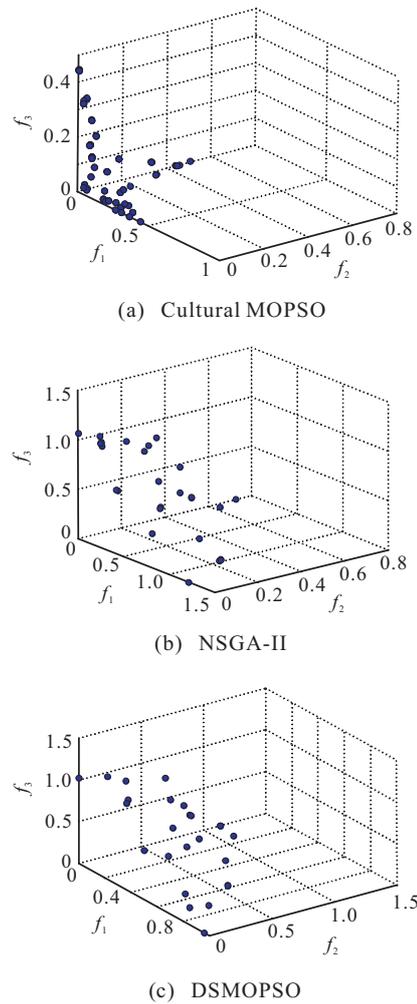


图3 测试问题 DTLZ1 的 Pareto 前沿对比

从图3可见,对于DTLZ1,文化粒子群的Pareto前沿显著优于另外两种算法,如表2中DTLZ1的趋近前沿 $C$ 值所示.同时,表2中Cultural MOPSO的分布均匀性比NSGA-II提高24.9%,比DSMOPSO提高43.1%.

### 4.2 敏感度分析

Cultural MOPSO的运算参数包括速度惯量调整系数 $\Delta\omega$ ,飞行系数调节常量 $\alpha$ 和 $\beta$ 等.以ZDT3测试问题为例,Cultural MOPSO使用不同的 $\Delta\omega$ 值所得到的Pareto前沿对比如图4所示.

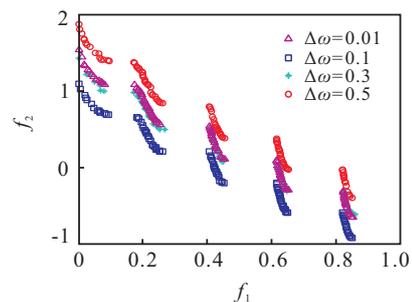


图4 Cultural MOPSO 不同  $\Delta\omega$  求解 ZDT3 的结果对比

分析 $\Delta\omega$ 值对Pareto前沿收敛性的影响,可以得出: $\Delta\omega$ 过大(如 $\Delta\omega = 0.5$ ),会对粒子飞行造成扰动,

导致最优解收敛性不足;而  $\Delta\omega$  过小(如  $\Delta\omega = 0.01$ ), 则不利于提高算法全局寻优能力;  $\Delta\omega$  的最优值应为 0.1.

设调节系数  $\alpha = \beta$ , 分析其不同取值对算法性能的影响. Cultural MOPSO 使用不同  $\alpha$  值求解 ZDT 3 问题获得的 Pareto 前沿与图 4 类似. 通过图形分析可以得出:  $\alpha$  取值过大(如  $\alpha = 1.0$ ), 会对粒子飞行造成扰动;  $\alpha$  取值过小(如  $\alpha = 0.001$ ), 则会造成粒子飞行系数的调节程度微弱, 最优前沿收敛性不足;  $\alpha = \beta = 0.01$  时, Pareto 前沿的收敛性最好.

## 5 结 论

拥挤距离排序多目标文化粒子群算法使用文化进化框架存储粒子群进化过程中的优势知识, 引入拥挤距离算子实现精英粒子数量控制、全局和局部最优粒子选择以及粒子飞行系数的动态调整, 有助于增强算法的全局寻优能力, 保持 Pareto 前沿多样性特征. 通过对两目标和三目标标准测试问题的求解, 并与 NSGA-II 和 DSMOPSO 算法进行对比, 表明了所提出算法的有效性. 通过变量敏感度分析, 获得了运算参数的最优取值, 从而证明了所提出的算法具有良好的鲁棒运算效果. 进一步, 应研究文化框架与其他多目标进化算法的结合方式, 开发用于工程设计问题求解的多目标优化软件平台.

## 参考文献(References)

- [1] Goh C K, Tan K C, Liu D S, et al. A competitive and cooperative co-evolutionary approach to multi-objective particle swarm optimization algorithm design[J]. *European J of Operational Research*, 2010, 202(1): 42-54.
- [2] Kennedy J, Eberhart R C. Particle swarm optimization[C].

Proc of IEEE Int Conf on Neural Networks. Perth: IEEE, 1995, 4: 1942-1948.

- [3] Li X D. A nondominated sorting particle swarm optimizer for multiobjective optimization[C]. Proc of Genetic and Evolutionary Computation. Berlin: Springer, 2003, 7: 37-48.
- [4] Deb K, Pratap A, Agarwal S. A fast and elitist multi-objective genetic algorithm: NSGA-II[J]. *IEEE Trans on Evolutionary Computation*, 2002, 6(2): 182-197.
- [5] Li Z K, Zhu Z C, Zhang H Q. DSMOPSO: A distance sorting based multiobjective particle swarm optimization algorithm[C]. Proc of the 6th Int Conf on Natural Computation. Yantai, IEEE Computer Society, 2010, 5: 2749-2753.
- [6] 施展, 陈庆伟. 基于QPSO和拥挤距离排序的多目标量子粒子群优化算法[J]. *控制与决策*, 2011, 26(4): 540-547. (Shi Z, Chen Q W. Multi-objective quantum-behaved particle swarm optimization algorithm based on QPSO and crowding distance sorting[J]. *Control and Decision*, 2011, 26(4): 540-547.)
- [7] Daneshyari M, Yen G G. Cultural MOPSO: A cultural framework to adapt parameters of multiobjective particle swarm optimization[C]. *IEEE Congress on Evolutionary Computation*. Hong Kong, 2008: 1325-1332.
- [8] Ricardo L B, Carlos A C C. Cultured differential evolution for constrained optimization[J]. *Computer Methods in Applied Mechanics and Engineering*, 2006, 195(7): 4303-4322.
- [9] Zitzler E, Deb K, Thiele L. Comparison of multi-objective evolutionary algorithms: Empirical results[J]. *IEEE Trans on Evolutionary Computation*, 2000, 18(2): 173-195.

(上接第1405页)

- [4] 罗钧, 李研. 具有混沌搜索策略的蜂群优化算法[J]. *控制与决策*, 2010, 25(12): 1913-1916. (Luo J, Li Y. Artificial bee colony algorithm with chaotic-search strategy[J]. *Control and Decision*, 2010, 25(12): 1913-1916.)
- [5] Quan H Y, Shi X L. On the analysis of performance of the improved artificial-bee-colony algorithm[C]. The 4th Int Conf on Natural Computation. Ji'nan, 2008: 654-658.
- [6] Alatas B. Chaotic bee colony algorithms for global numerical optimization[J]. *Expert Systems with Applications*, 2010, 37(8): 5682-5687.
- [7] Karaboga D, Basturk B. Artificial bee colony(ABC) optimization algorithm for solving constrained optimization problems[C]. *Foundations of Fuzzy Logic and*

*Soft Computing*. Cancun, 2007: 789-798.

- [8] Karaboga D, Akay B, Ozturk C. Artificial bee colony(ABC) optimization algorithm for training feed-forward neural networks[J]. *Modeling Decisions for Artificial Intelligence*, 2007, 4617: 318-329.
- [9] 高尚, 杨静宇. 群智能算法及其应用[M]. 北京: 中国水利水电出版社, 2006: 6-10. (Gao S, Yang J Y. Swarm intelligence algorithm and its application[M]. Beijing: China WaterPower Press, 2006: 6-10.)
- [10] Bao L, Zeng J C. Comparison and analysis of the selection mechanism in the artificial bee colony algorithm[C]. 2009 9th Int Conf on Hybrid Intelligent Systems(HIS 2009). Shenyang, 2009: 411-416.