

解决动态多中心问题的自学习差异进化算法

刘星宝^{1,2}, 殷建平², 胡春华¹, 陈荣元^{1,2}

(1. 湖南商学院 计算机与信息工程学院, 湖南 长沙 410205; 2. 国防科学技术大学 计算机学院, 湖南 长沙 410073)

摘要: 为解决动态环境下的多中心优化问题, 提出自学习差异进化算法。通过评估特定个体检测到环境变化, 自学习算子将群体引至新的环境, 并保持群体的拓扑结构不变, 以继续当前的进化趋势。采用邻域搜索机制加快算法的收敛速度, 引入随机个体迁入机制增加群体多样性。实验以周期动态函数为测试对象, 比较自学习差异进化算法与部分智能优化算法的性能, 结果表明, 新算法有更快的收敛速度和更好的环境适应能力。

关键词: 进化计算; 动态优化; 自学习机制; 多中心动态优化问题; 差异进化

中图分类号: TP18

文献标识码: A

Self-learning differential evolution algorithm for dynamic polycentric problems

LIU Xing-bao^{1,2}, YIN Jian-ping², HU Chun-hua¹, CHEN Rong-yuan^{1,2}

(1. School of Computer & Information Engineering, Hunan University of Commerce, Changsha 410205, China;

2. School of Computer, National University of Defense Technology, Changsha 410073, China)

Abstract: A novel self-learning differential evolution algorithm is proposed to solve dynamical multi-center optimization problems. The approach of re-evaluating some specific individuals is used to monitor environmental changes. The proposed self-learning operator guides the evolutionary group to a new environment, meanwhile maintains the stable topology structure of group to maintain the current evolutionary trend. A neighborhood search mechanism and a random immigrant mechanism are adapted to make a tradeoff between algorithmic convergence and population diversity. The experiment studies on a periodic dynamic function set suits are done, and the comparisons with peer algorithms show that the self-learning differential algorithm outperforms other algorithms in term of convergence and adaptability under dynamical environment.

Key words: evolutionary computation; dynamic optimization; self-learning mechanism; multi-center dynamic optimization problems; differential evolution

1 引言

在过程追踪、生产调度、交通运输、目标跟踪和资本预算等领域中, 存在大量具有时序特征的动态模型, 即模型的设计变量、估价函数、约束条件等因素与时间相关, 称此类问题为动态问题。根据构造方式的不同, 动态问题大体可分 3 类^[1]: 第 1 类是基于二进制编码的异或运算生成的动态函数,

如 OneMax 函数、Royal Road 函数和 Deceptive 函数等; 第 2 类是约束条件变化产生的动态问题, 如动态旅行商问题、动态背包问题等; 第 3 类是改变静态函数的拓扑结构生成的动态函数, 如 MPB (moving peaks benchmarks) 函数中, 峰的高度、宽度、位置等属性随时间改变。当最优解位置变化时, MPB 被称为多中心动态优化问题。

智能算法处理动态问题时面临诸多挑战, 大体

收稿日期: 2014-08-06; 修回日期: 2014-10-20

基金项目: 国家自然科学基金资助项目 (61273232, 41101425); 教育部新世纪优秀人才支持计划基金资助项目 (NECT-2013-0785)

Foundation Items: The National Natural Science Foundation of China(61273232, 41101425); The Ministry of Education Program for New Century Excellent Talents (NECT-2013-0785)

可归纳为3个方面^[2]: 1) 因环境的动态变化, 群体历史信息的复用性降低; 2) 优化环境变化前后, 个体间的相对优劣度发生改变, 致使进化策略不再适合环境; 3) 动态环境要求更高的群体多样性、耗费更多的计算资源。当前, 增加群体多样性和引入记忆机制是智能算法处理动态问题的主要方法^[3]。第一, 通过变异和个体迁入机制增加群体多样性。当探测到环境发生变化后, 通过超级变异或者自组织变异, “瞬间”或者逐步增加个体的变异率^[4]; Grefenstette^[5]用随机个体替换进化种群的部分个体, 保证每一代都引入新的基因物质。为提高引入个体的“存活率”, Yang^[6]提出了具有导向性的个体迁入机制, 能够更高效地解决某几类动态优化问题。个体迁入机制对解决某一类问题, 如环境变化、周期动态变化等问题具有较好的作用^[7]。第二, 引入记忆机制提高算法的环境适应能力。利用基因的冗余隐式存储有用信息^[8], 或者划定空间显示存储信息^[9]; Bendsen 提出一种自我调整的动态显示记忆模型, 实验证明该方法优于静态记忆机制, 但因候选解被选中的概率较小, 不适应有较大定义域的动态问题^[10]。

多样性保持和记忆机制增加了进化群体对动态环境的适应能力, 提高了算法的求解精度和效率, 但动态进化算法设计大多针对普遍性的动态问题, 此类问题多种因素同时变化, 极大地增加了算法的设计难度。基于以上考虑并结合多中心动态优化问题的特点, 本文提出具有自学习机制的差异进化 (SeDE, self-learning differential evolution) 算法, 特定环境的精英个体将群体映射到新的环境, 并用该环境下的历史最优解引导群体, 加快算法的收敛速度。群体更新中引入随机个体, 以补偿学习算子引起的群体多样性不足的问题。

2 动态环境构造

动态优化问题 (DOP, dynamic optimization problems) 的数学描述为

$$\begin{aligned} & \min f(x, t) \\ \text{s.t.} & \begin{cases} h_i(x, t) = 0, i = 1, 2, \dots, p \\ g_j(x, t) < 0, j = 1, 2, \dots, q \\ x = (x_1, x_2, \dots, x_d), x_i \in D_i, i = 1, 2, \dots, n \end{cases} \end{aligned} \quad (1)$$

其中, $f(x, t)$ 是与时间相关的目标函数, $h_i(x, t) = 0$ 是第 i 个与时间 t 相关的等式约束条件, $i = 1, 2, \dots, p$;

$g_j(x, t) < 0$ 是第 j 个与时间 t 相关的不等式约束条件, $j = 1, 2, \dots, q$ 。时间因素 t 变化时, 函数 $f(x, t)$ 的维度、极值、极值的位置等因素均可能发生变化。

定义 1 $f(x)$ 是 n 维空间的静态函数, $o = \varphi(t)$ 是时间 t 与点 o 间的双射函数, 以 $f(x)$ 为基的多中心动态问题 $F(x, t)$ 定义为

$$F(x, t) = f(x, \varphi(t)) \quad (2)$$

算法中的迭代次数、运算平台的物理时间等因素均可驱动函数 $F(x, t)$ 动态变化。以二维 Sphere 函数 $f(x) = x_1^2 + x_2^2, x \in R^2$ 为例构造动态函数如下

$$F(x, t_i) = (x_1 - o_{i1})^2 + (x_2 - o_{i2})^2, i = 1, 2, \dots, K \quad (3)$$

当 t 变化时, 函数中心在 o_1, o_2, \dots, o_K 间循环变化, K 是动态环境的数目。当 $\varphi(t)$ 沿着不同的曲线运动时, 如弦函数、指数函数和多项式函数 $p_n(t)$ 等, 函数 $F(x, t)$ 的中心有不同的运动轨迹。

多中心动态函数的最优解随时间 t 变化, 增加了算法的搜索难度。主要体现在 2 个方面: 1) 进化过程中群体逐渐逼近最优解, 但环境动态变化改变了最优解位置, 则群体需要重新适应环境, 减缓了算法的收敛速度; 2) 当进化群体陷入局部极值点时, 失去了持续进化的能力, 无法搜索新环境的最优解。具体到多中心动态函数, 其特点是最优解由函数 $\varphi(t)$ 的运动轨迹决定, 这是算法设计时可以利用的前验知识。

定义 2 进化群体 $P = \{x_1, x_2, \dots, x_{NP}\}$, $x_i = (x_{i1}, x_{i2}, \dots, x_{in}), x_{ij} \in D_j, i = 1, 2, \dots, NP, j = 1, 2, \dots, n$, P 在问题空间 $D = (D_1, D_2, \dots, D_n)$ 中的分布称作群体 P 的拓扑结构。

定义 3 进化群体 $P = \{x_1, x_2, \dots, x_{NP}\}$, $x_i = (x_{i1}, x_{i2}, \dots, x_{in}), x_{ij} \in D_j, i = 1, 2, \dots, NP, j = 1, 2, \dots, n$, 称 $M(P)$ 为群体 P 的欧几里德距离矩阵, $M(P) = (m_{ij})_{NP \times NP}$, 其中, m_{ij} 为个体 x_i 与 x_j 间的欧几里德距离。

定义 4 设 τ 是双射函数, P 是进化群体, 若 $\tau(P)$ 与 P 有相同的欧几里德距离矩阵, 则称 τ 为群体 P 的同态映射。

定理 1 P 是进化群体, $\forall x \in P$, 参考点 $a, b \in D$, $D = (D_1, D_2, \dots, D_n)$, 双射函数 $\tau(x) = x + (b - a)$ 将群体从点 a 映射到点 b , 则 τ 不改变群体 P 的拓扑结构。

证明 设群体 $P = \{x_1, x_2, \dots, x_{NP}\}$, $x_i = (x_{i1}, x_{i2}, \dots, x_{in}), x_{ij} \in D_j, i = 1, 2, \dots, NP, j = 1, 2, \dots, n$ 。映射

前后的欧几里德距离矩阵分别为 $M(P)$ 和 $M'(P)$ ，对应的第 i 行第 j 列元素分别记作 m_{ij} 和 m'_{ij} 。根据定义 3 有

$$\begin{aligned}
 m_{ij} &= \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \dots + (x_{in} - x_{jn})^2} \\
 &= \sqrt{\sum_{k=1}^n (x_{ik} - x_{jk})^2} \\
 m'_{ij} &= \sqrt{(\alpha(x_{i1}) - \alpha(x_{j1}))^2 + (\alpha(x_{i2}) - \alpha(x_{j2}))^2 + \dots + (\alpha(x_{in}) - \alpha(x_{jn}))^2} \\
 &= \sqrt{\sum_{k=1}^n [(x_{ik} - a_k + b_k) - (x_{jk} - a_k + b_k)]^2} \\
 &= \sqrt{\sum_{k=1}^n (x_{ik} - x_{jk})^2} \\
 &= m_{ij}
 \end{aligned}$$

即映射前后群体 P 的拓扑结构不变。

证毕。

3 SeDE 算法

与静态优化问题最优解固定不同，多中心动态优化问题的最优解与时间相关。据此 SeDE 算法的设计目标有 2 个：良好的跟踪能力和较快的收敛速度。新算法主要由 3 部分构成：1) 动态环境检测机制检测优化环境的变化，并确定群体所在环境的信息；2) 学习机制引导群体从当前环境映射到下一个环境，利用历史信息提高算法的环境适应能力；3) 个体随机迁入机制为群体提供更多的多样性。算法框架如下。

Step1 在定义域内初始化群体 P ， NP 个体、 D 维；初始化参数 F 和 CR 。

Step2 动态环境检测：检测环境变化，若优化环境改变，则执行 Step3~Step7，否则，执行 Step4~Step7。

Step3 学习操作：判定当前优化环境所在的中心，用当前中心的历史最优解引导群体 P 学习适应环境。

Step4 进化操作：通过变异和交叉操作产生后代 *Children*。

Step5 评估群体 *Children*，从父代和对应的子代中选择优秀个体。

Step6 随机迁入，若最优解连续 N 代没有更新，则用随机个体替换掉群体中 $p\%$ 随机选择的个体。

Step7 若满足结束条件，则输出相关统计数

据；否则，执行 Step2。

说明 该算法的“结束条件”有多种表达方式，如限定群体的进化代数、进化过程中群体的评价次数、优化结果的相对误差等。

3.1 环境检测机制

环境检测机制包含 2 个阶段，首先检测环境是否发生变化，其次确定当前环境所处的中心。在 DOP 中检测环境变化的方法主要有 2 种^[11]，即重新评估环境监测器和监测算法行为，优化问题的特点决定了环境检测方式的选择。本文处理的动态优化问题有 2 个不同的中心，当环境从一个中心转变到另一个中心后，个体适应值发生明显变化。图 1 以二维 Sphere 函数 $f(x_1, x_2) = x_1^2 + x_2^2$ 最小化为例，当时间变化时产生 2 个中心点 o_1 和 o_2 ，个体 u 和 v 是进化群体中的随机个体。当优化环境处于 o_2 时，个体 u 接近中心点 o_2 ，优于远离 o_2 的个体 v ；但当优化环境从 o_2 变化到 o_1 时，个体 u 的适应值迅速增加，劣于个体 v 。所以中心点的动态变化改变了个体的适应值，本文以特定个体作为环境变化监测器，如进化群体的最优个体、问题空间中随机生成的个体等。

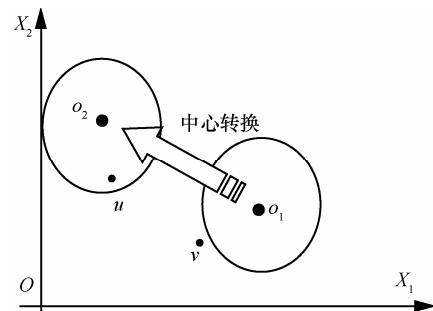


图 1 动态环境中最优解从 o_1 变化到 o_2

3.2 自学习机制

环境从中心 o_i 变化到中心 o_{i+1} 时，破坏了个体在中心 o_i 下形成的模式，这种“破坏”作用持续进行，直至环境从中心 o_{i+1} 转变回中心 o_i 。同样个体在环境 o_i 下的搜索学习过程也是对中心 o_{i+1} 下个体模式的破坏。所以环境动态变化既是当前环境下个体模式的“重建”过程，也是前一个环境下个体模式的“破坏”过程。个体模式的“破坏”和“重建”减缓了算法的收敛速度，在给定的评价次数、进化代数等资源限制下，算法不容易获得较好的优化结果。另一方面，进化过程中群体会获得有价值的历史信息，如对应于环境的历史最优解等，能够很好地指导群体的进化过程。

3.2.1 环境的自适应机制

根据多中心问题最优值随时间周期变化的特点，设计自学习算子以降低环境动态变化对算法性能的影响。从 i 环境转变到 j 环境的“瞬间”，环境监测器发现环境变化后，则学习算子自动将 i 环境下的个体映射到 j 环境。由于该过程使用同态映射，保证了映射前后群体的拓扑结构相似。具体设计中，2 个环境下的历史最优解 $stageBestIndi(i)$ 和 $stageBestIndi(j)$ 同时作用于个体 x ，同时引入参数 $omiga$ 控制新环境下群体结构的变化。当 $omiga > 1$ 时，映射前后群体的整体结构不变，但个体间的相对距离增加；当 $omiga = 1$ 时，群体的整体结构不变，但个体间的相对距离不变；当 $omiga < 1$ 时，群体的整体结构不变，但个体间的相对距离减小。另外，进入新环境后，个体向该环境下的历史最优解和 2 个随机选择的个体学习，执行贪婪的邻域搜索，即个体的变异操作。自适应机制描述如下

$$\forall x \in P, x^* = x + omiga(stageBest(j) - stageBest(i)) \quad (4)$$

其中， x^* 是 x 的学习后得到的新个体， $omiga$ 是控制群体拓扑结构的缩放因子。图 2 以二维空间中 2 个中心动态函数为例说明个体 x 的学习过程。当前环境下群体 P 搜索中心 o_1 的邻域， $x \in P$ 。当优化环境从 o_1 变化到 o_2 时，个体 x 被映射到 x^* 。算法 1 具体描述了个体 x 在优化环境变化“瞬间”的学习过程，确定从第 i 个环境变化到第 j 个环境，历史最优解分别为 $stageBestIndi(i)$ 和 $stageBestIndi(j)$ ，进化群体为 P 。

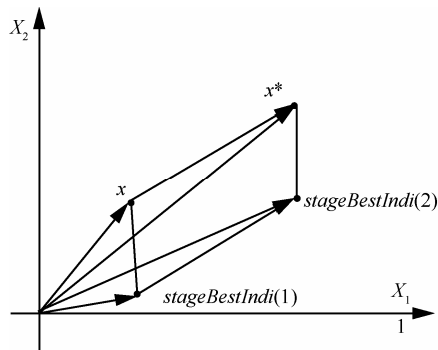


图 2 个体学习机制

算法 1 个体的自学习机制

Step1 如果 P 在 i 环境下的最优解 $bestIndi$ 优于历史最优解 $stageBestIndi(i)$ ，则更新 $stageBestIndi(i)$

Step2 用 j 环境下历史最优解引导 P 中个体：

$\forall x \in P, x^* = x + omiga(stageBest(j) - stageBest(i))$ ；
得到新群体 P^* 。

Step3 在 j 环境下重新评估群体 P^* ，并找出 P^* 中的当代最优解 $bestIndi$ 。

Step4 如果 $bestIndi$ 优于 j 环境下的历史最优解 $stageBestIndi(j)$ ，则替换；否则 $stageBestIndi(j)$ 不变。

Step5 群体 P^* 在 j 环境下进入进化过程。

对学习机制的说明：1) 个体的学习机制应用于中心点转换的瞬间，利用历史信息引导个体更好地适应新环境；2) 学习机制类似于精英保留策略，能加速算法在新环境中的收敛速度，但要付出快速丧失群体多样性的代价。

3.2.2 个体的自学习机制

在智能算法设计中，进化操作不仅驱动群体逐渐逼近最优解，还要始终保持一定的多样性避免算法早熟。动态问题中优化环境随时间改变，要求算法在有限的进化代数内尽快逼近最优解；同时群体需要保持较高的多样性以适应改变后的环境。因此，进化操作中采用当代最优个体引导的贪婪变异算子和随机个体迁入机制，以平衡收敛速度和多样性。

变异算子： $pbest(gen)$ 表示第 gen 代群体的最优解， $v(gen, i)$ 表示第 i 个体 $P(gen, i)$ 在 gen 代生成的测试向量，第 i 个体的变异后代生成方式如下

$$v(gen, i) = pbest(gen) + F(pbest - P(gen, r_1)) + k(P(gen, r_2) - P(gen, r_3)) \quad (5)$$

其中， F 是标量变异步长， k 是 $(0, 1]$ 间的随机分布， $P(gen, r_j), j = 1, 2, 3$ 是从第 gen 代群体 P 中随机选择的 3 个互不相同个体，且 $P(gen, r_j) \neq P(gen, i), j = 1, 2, 3$ 。该变异操作在当代最优解 $pbest(gen)$ 引导下的变异策略，生成的测试向量主要分布在 $pbest(gen)$ 周围，用以搜索当代最优解所在的邻域，加快算法的收敛速度。

交叉算子：交叉操作采用指数交叉。变异操作生成的测试向量 $v(gen, i)$ 和 gen 代的第 i 个体 $P(gen, i)$ 交叉生成目标向量 $u_i, u_i = (u_{i1}, u_{i2}, \dots, u_{iD}), i = 1, 2, \dots, np, np$ 是群体规模。

$$u_{ij} = \begin{cases} v_{ij} & , k < CR, j = j_rand \\ P_{ij} & , \text{其他} \end{cases} \quad (6)$$

其中， k 是区间 $[0, 1]$ 上的随机分布， CR 是交叉概率，

j_rand 是 $[1, \dots, D]$ 中的随机整数, 条件 $j = j_rand$ 以保证 u_i 和 $P(gen, i)$ 中至少有一维不同。变异和交叉算子中的 2 个参数 F 和 CR 采用自适应的控制策略^[4]。

3.3 多样性与搜索效率的均衡机制

目的是维持进化群体规模的稳定性, 平衡贪婪变异算子导致的群体多样性丧失问题。更新操作的参与对象是群体 $P(x_{i1}, \dots, x_{iD})$ 和 $u_i = (u_{i1}, u_{i2}, \dots, u_{iD})$, $i = 1, 2, \dots, np$, 群体更新过程分 2 个步骤。

Step1 从 u_i 和 x_i 中选择优秀的个体 x'_i , 记 $P' = (x'_{i1}, x'_{i2}, \dots, x'_{iD}), i = 1, 2, \dots, np$

$$x'_i = \begin{cases} u_i, u_i \text{ 优于 } P(gen, i) \\ P(gen, i), \text{其他} \end{cases}, i = 1, 2, \dots, np \quad (7)$$

Step2 根据群体的进化情况, 用随机个体替换群体 P' 中的部分个体, 以增加群体的多样性。在群体进化过程中, 若连续 k 代最优值没有更新, 则生成 t 个随机个体替换群体中 t 个非最优个体。根据前期的实验结果设定 $k = 10$, $t = \lfloor 0.2np \rfloor$, $\lfloor x \rfloor$ 表示不大于 x 的整数。

4 实验与分析

采用 4 个可扩展静态函数生成两中心动态环境。在函数 $f(x)$ 的定义域范围内随机生成固定点 $a = (a_1, \dots, a_d)$, 令 $f^*(x) = f(x - a)$ 。容易得到函数 $f^*(x)$ 最优值为 0^d , 最优解为 a 。优化环境在 $f(x)$ 和 $f^*(x)$ 间循环转换, 构成动态变化环境。生成动态测试环境的静态基函数如表 1 所示, f_1 、 f_2 是单模函数, 测试算法适应环境变化的能力和邻域搜索能力。 f_3 、 f_4 是多模函数, 测试动态环境中算法跳出局部极值点的能力。

4.1 自学习机制对算法性能的影响

自学习机制是 SeDE 算法应对环境变化的重要

算子, 在环境变化的瞬间引导群体从当前中心进入新的中心。在环境变化时缩放参数 $omega$ 控制群体结构放大、缩小或者保持不变, 是影响算法收敛性的重要参数。首先, 以函数 f_1 为例模拟进化过程中群体学习前后拓扑结构的变化, 如图 3 所示。其次, 以单模函数 f_1 和多模函数 f_3 为例, 测试控制参数 $omega$ 不同取值对算法收敛速度的影响, 如图 4 所示。

图 3 是进化过程中环境变化前后的个体分布。“+”和“o”点分别表示学习前后的个体在二维空间中的分布, 符号“■”表示群体中心, “☆”表示对应于 2 个中心的历史最优解。如图 3(a)所示, 在第一次环境变化时个体分布较为分散。随着进化过程的深入, 个体逐渐分布在历史最优解的周围, 形成对其的局部精细搜索, 提高了算法的邻域搜索能力。就进化个体而言, 在每一次环境变化时, 个体向新环境下的历史最优解学习, 被引导至新的环境; 环境变化前后, 个体相互间的位置关系没有变化, 即保持逼近历史最优解的趋势, 削弱环境变化对算法收敛速度的影响。另一方面, 图 3 也表明了自学习机制是贪婪的邻域搜索, 群体多样性随着进化过程的深入丢失严重, 需要引入其他机制平衡群体的多样性。

在图 4 中, 当 $omega = 1$ 时, 算法对函数 f_1 和 f_3 的搜索性能提高很多。但图 4 中的收敛性曲线都出现了非常大的波动, 算法得到的优化结果很差, 表明进化群体因多样性的丧失而陷入局部极值, 失去了适应环境变化的能力, 此时触发了随机个体迁入机制, 算法生成随机个体替代部分不能继续进化的个体, 增加了群体多样性, 进化过程得以继续进行。当 $omega$ 为 0.5 和 1.5 时, 收敛曲线呈现细微锯齿形, 算法得到的函数最优值出现反复, 表明环境变化破

表 1 生成动态环境的静态基函数集

函数表达式	状态空间	中心 1 近似最优值	中心 2 近似最优值
$f_1(x) = \sum_{i=1}^D x_i^2$	$-10 \leq x_i \leq 10$	0	0
$f_2(x) = \sum_{i=1}^D x_i + \prod_{i=1}^D x_i $	$-10 \leq x_i \leq 10$	0	0
$f_3(x) = -20 \exp\left(-0.2 \sqrt{\frac{1}{D} \sum_{i=1}^D x_i^2}\right) - \exp\left(\frac{1}{D} \sum_{i=1}^D \cos 2\pi x_i\right) + 20 + e$	$-32 \leq x_i \leq 32$	0	0
$f_4(x) = \frac{1}{4000} \sum_{i=1}^D x_i^2 - \prod_{i=1}^D \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$-20 \leq x_i \leq 20$	0	0

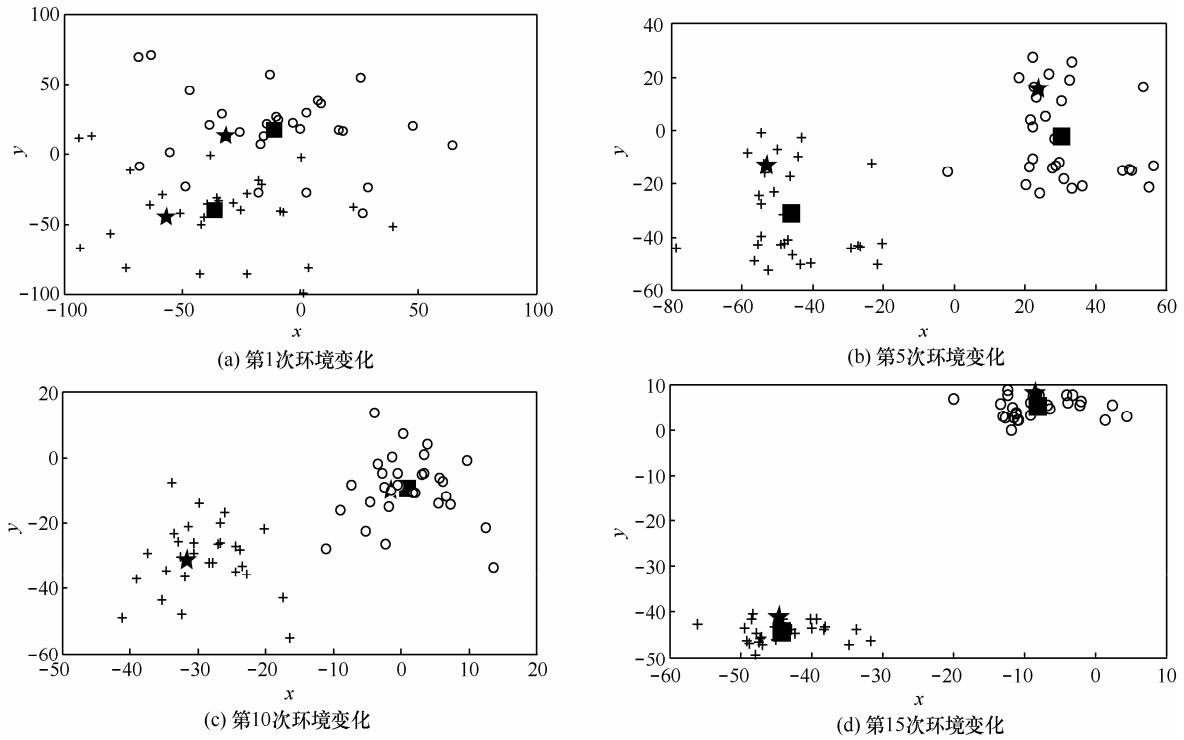


图 3 环境变化前后个体的相对位置不变

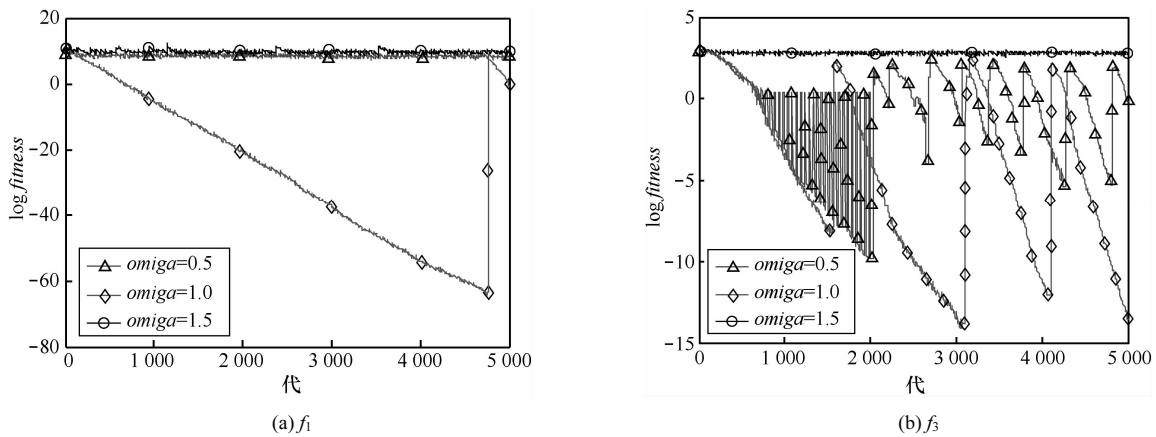


图 4 不同控制参数值对算法收敛性的影响

坏了群体向最优解进化的趋势。

4.2 算法性能比较实验

将 SeDE 算法与均匀粒子群优化算法(UPSO)^[12]、差异进化(DE)^[13]和参数自适应差异(jDE)^[14]进化算法进行对比,测试 4 种优化算法对动态环境的适应能力。UPSO 算法将个体的社会认知和个体认知能力做了线性化处理,仍具有较强的邻域搜索能力。jDE 算法具有 4 种算法中最好的全局搜索能力,而 DE 算法在随机选择变异策略的情况下,具有较为平衡的全局和局部搜索能力。实验中每个测试函数独立运行 30 次,进化代数限定为 1 000 次。

由于已知被优化函数在各个中心的最优值,本文采用离线误差评价算法应对环境变化的适应能力,适应性评价(APE, adaptive performance evaluation)描述如下

$$APE(i) = \frac{|f^*(i) - f_{best}(i)| \cdot i \cdot genLmt}{\max Gen}, i = 1, 2, \dots, m \quad (8)$$

其中,APE 是算法在第 i 次环境变化中的适应能力, m 是环境变化的总次数, $genLmt$ 环境变化频率, $\max Gen$ 是算法的最大进化代数, $f_{best}(i)$ 是第 i 次环境变化中算法获得的最优值和, $f^*(i)$ 该环境下的

理论最优值。 $|f^*(i) - f_{best}(i)|$ 计算第 i 次环境中算法获得的最优解与理论最优解的绝对距离， $\frac{i \cdot genLmt}{\max Gen}$ 平衡不同进化阶段的最优值。

适应性评价 $APE(i)$ 定量给出第 i 次环境变化中，群体向理论最优点的逼近程度，即算法在该“时间片段”内体现出的搜索能。若每次环境变化后算法都能搜索到理论最优值，即 $|f^*(i) - f_{best}(i)| = 0$ ，则 $APE(i) = 0$ 。

4.2.1 算法对中心距离变化适应能力

设被优化函数 f 第 i 维分量的定义域是 $[lb_i, ub_i]$ ，两中心点 O_1 和 O_2 之间的距离定义如下

$$|O_{2i} - O_{1i}| = \frac{(ub_i - lb_i)}{2r}, \quad r = 2, 3, 4, 5, i = 1, 2, \dots, d \quad (9)$$

在环境变化频率为 25 时，各算法独立运行 30 次，统计每次最优值的平均值，测试两中心间距离对算法性能的影响，实验结果如表 2 所示。在 $r=2,3,4,5$ 几种情况下，SeDE 都优于其他 3 种算法，如表 2 的“+”行所示。只有在 $r=2$ 时，UPSO 算法优化函数 f_3 的结果好于 SeDE 的搜索结果。从单个算法看，UPSO 比 DE、jDE 的搜索结果更好，这是因为粒子群算法具有更好的邻域搜索能力，能在有限的进化代数（环境变化频率为 25 代）内更快地逼近最优解。DE 算法和 jDE 算法比 PSO 算法有更好的群体多样性，相应的个体在空间中的分布区域更广泛，逼近最优解需要的进化代数更多，因此算法的收敛速度比 SeDE 和 UPSO 算法慢。

图 5 选取函数单模函数 f_1 和多模函数 f_4 为实验对象，测试 4 种算法在不同中心距离下的适应能力，横坐标是环境变化的次数，纵坐标基于 APE 准则的算法适应性能力值。如图 5 所示，SeDE 算法的适应能力高于其他 3 种算法，表明每次环境变

化后，SeDE 算法能快速逼近最优解，具有良好的环境适应性。除 SeDE 算法外，UPSO 算法表现优于 DE 和 jDE 算法，表明该算法有更好的环境适应能力，比较适合处理动态环境中的优化问题。

优化环境以 $genLmt=25、50、100$ 和 150 代为周期时，几种算法在动态环境中的适应能力 APE 得分如表 3 所示。可以看出在 4 种变化频率下，算法 SeDE 对环境的适应能力都优于其他 3 种算法，如表 3 中“+”行所示。整体上看，环境变化频率对算法的适应能力影响较大。随着 $genLmt$ 从 25 增加到 150，4 种算法的环境适应能力得分逐渐变小，如算法 SeDE 在 4 种频率下优化函数 f_1 ，APE 值分别为 1.34、0.74、0.41 和 0.24，呈明显的下降趋势，表明 SeDE 的环境适应能力逐步增强，在一个变化周期内搜索精度逐渐增加。另一方面也表明，优化环境的动态变化会改变群体逼近最优解的趋势，导致算法的搜索精度下降。

4.2.2 算法对频率变化的适应能力

实验中群体规模设定为 30，进化代数限定为 1 000 代（相应评价次数为 31 030）、 $r=2$ ，环境以 $genLmt=25、50、100$ 和 150 代为频率变化。算法在 $genLmt=25$ 时优化 4 个函数的收敛性曲线如图 6 所示。

图 6 是 $genLmt=25、r=2$ 时算法优化 4 个函数得到的收敛性曲线。其中，SeDE 算法的搜索能力最好，优于其他 3 种算法。这是因为 SeDE 算法中自学习机制将个体等位置映射到新环境中，保持了向最优解逼近的趋势。同时，4 种算法的收敛性曲线都呈现明显的锯齿状，这是由动态优化问题的特点决定的。以 SeDE 算法为例，贪婪的变异算子加快了算法的收敛速度，提高了算法在一个变化周期内的优化精度，但群体的多样性变差，群体陷入局部极值点。当满足触发条件时，

表 2 两中心点距离变化时，不同优化算法搜索得到的平均最优值

算法	Mean(bestFit), r=2				Mean(bestFit), r=3				Mean(bestFit), r=4				Mean(bestFit), r=5			
	f_1	f_2	f_3	f_4	f_1	f_2	f_3	f_4	f_1	f_2	f_3	f_4	f_1	f_2	f_3	f_4
SeDE	1.6×10^{-4}	2.4×10^{-2}	3.5	6.2×10^{-3}	8.9×10^{-6}	1.5×10^{-2}	1.9×10^{-2}	3.7×10^{-3}	5.6×10^{-5}	1.3×10^{-2}	3.0×10^{-2}	2.3×10^{-5}	9.5×10^{-6}	9.5×10^{-3}	3.2×10^{-2}	5.7×10^{-6}
DE	1.3×10^2	4.0×10^1	1.6×10^1	1.1	6.3×10^1	3.9×10^1	1.5×10^1	1.1	5.5×10^1	3.4×10^1	1.3×10^1	1.1	2.7×10^1	3.5×10^1	1.4×10^1	1.0
jDE	8.7×10^1	4.7×10^1	1.5×10^1	1.1	3.8×10^1	3.0×10^1	1.2×10^1	1.0	2.2×10^1	2.3×10^1	1.0×10^1	9.8×10^{-1}	1.2×10^1	1.8×10^1	8.6	9.2×10^{-1}
UPSO	9.2×10^{-2}	1.3	1.4	5.4×10^{-2}	5.5×10^{-1}	1.0×10^1	3.5	9.6×10^{-2}	2.8×10^1	1.9×10^1	6.3	7.5×10^{-1}	3.9	3.7×10^1	1.5	4.1×10^{-1}
+	+3	+3	+2	+3	+3	+3	+3	+3	+3	+3	+3	+3	+3	+3	+3	+3

“+”表示 SeDE 算法相比于其他 3 种算法占优的个数。

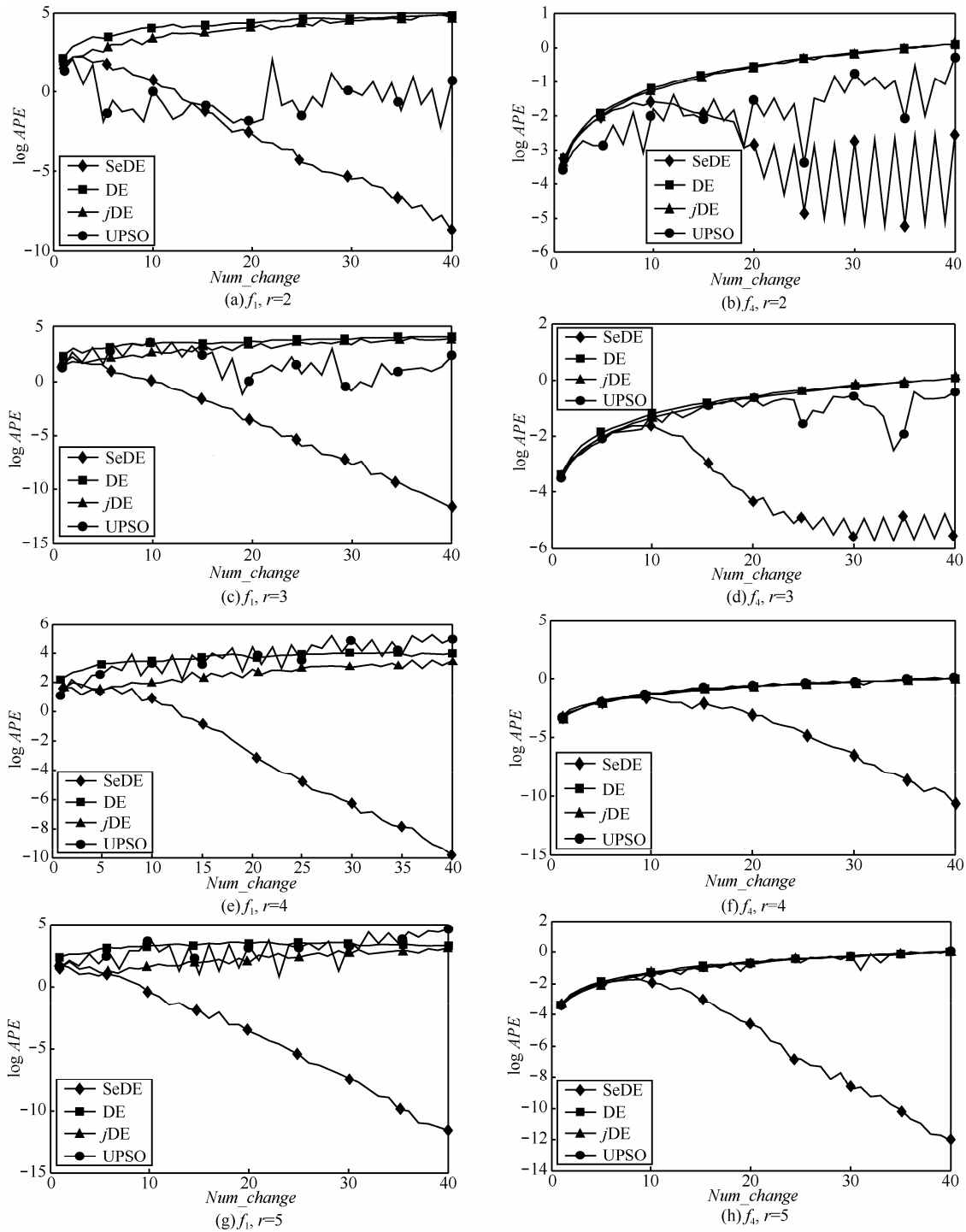


图 5 中心点距离变化时，不同算法的适应性曲线

算法生成随机个体替代群体中部分非最优个体，增加了多样性，从而群体能够跳出局部极值点继续进化过程。

5 结束语

多中心动态问题在诸多领域有重要的应用，但

其优化方法的研究相对较少。本文以两中心动态问题为切入点研究优化该类问题的智能优化方法，提出了具有较强环境适应性的自学习进化算法。根据两中心动态优化问题的特点，环境变化的“瞬间”，新环境的历史最优个体引导群体学习，将当前环境下的群体映射到新环境中，且保持个体间的相对位

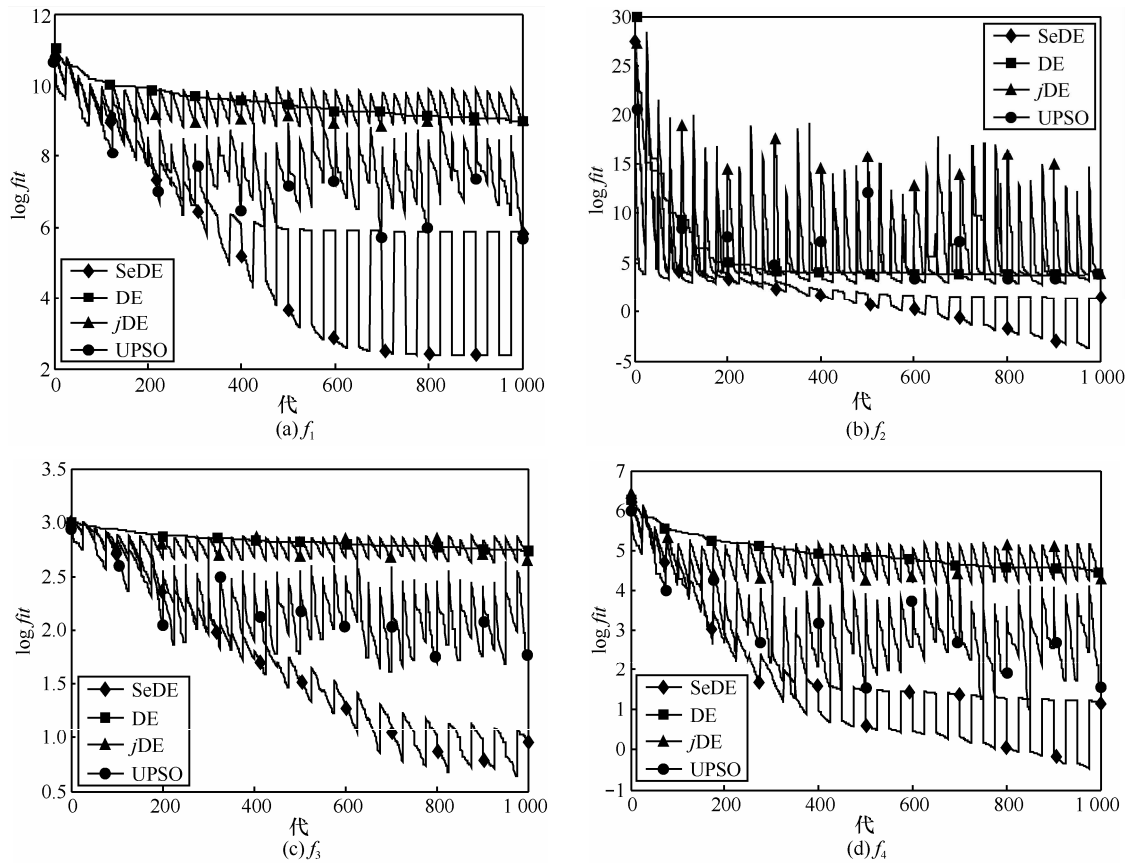


图 6 在环境变化频率为 25 代时，不同算法的收敛曲线

表 3 环境变化频率发生变化时，不同算法的适应能力

算法	mean(APE), genLmt=25				mean(APE), genLmt=50				mean(APE), genLmt=100				mean(APE), genLmt=150			
	f_1	f_2	f_3	f_4	f_1	f_2	f_3	f_4	f_1	f_2	f_3	f_4	f_1	f_2	f_3	f_4
SeDE	1.34	2.59	1.13	0.27	0.74	1.00	0.65	0.04	0.41	3.16	0.50	0.03	0.24	3.62	0.90	0.04
DE	73.53	394.77	8.612	0.57	48.42	26.71	8.10	0.58	8.18	16.8	5.65	0.50	4.51	13.56	4.09	0.37
jDE	38.45	24.67	7.35	0.55	23.31	16.56	6.59	0.55	8.25	9.00	5.03	0.52	2.28	3.82	5.68	0.26
UPSO	11.26	15.58	3.79	0.36	9.69	11.87	2.87	0.30	13.16	10.33	5.66	0.23	16.80	8.67	4.97	0.15
+	+3	+3	+3	+3	+3	+3	+3	+3	+3	+3	+3	+3	+3	+3	+3	+3

“+”表示 SeDE 算法相比于其他 3 种算法占优的个数。

置不变，以保持持续的进化过程。系列实验结果表明算法能尽快适应环境的动态变化，具有良好的全局和局部搜索能力。本文只设计了解决两中心的动态优化问题，而现实生活中存在大量离散多中心动态优化问题，此类问题的研究不仅具有极高的挑战性，也具有实用的实用价值，今后将着重研究此类问题。

参考文献:

[1] NGUYEN T T, YANG S, BRANKE J. Evolutionary dynamic optimization: a survey of the state of the art[J]. Swarm and Evolutionary

Computation, 2012, 6: 1-24.
 [2] CRUZ C, GONZÁLEZ J R, PELTA D A. Optimization in dynamic environments: a survey on problems, methods and measures[J]. Soft Computing, 2011, 15(7): 1427-1448.
 [3] 陈莉, 丁立新. 动态优化算法综述[J]. 武汉大学学报(理学版), 2011, 57(3): 255-264.
 CHEN L, DING L X. Survey on dynamic optimization algorithms[J]. Journal of Wuhan University(Natural Science Edition), 2011, 57(3): 255-264.
 [4] FERNANDES C M, MERELO J J, RAMOS V, et al. A self organized criticality mutation operator for dynamic optimization problems[A]. Proceedings of the 10th Annual Conference on Genetic and Evolutionary Computation[C], New York: ACM Press, 2008.937-944
 [5] GREFENSTETTE J. Genetic Algorithms for Changing Environ-

- ments[M]. Navy Center for Applied Research in Artificial Intelligence: Navy Research Laboratory, 1992. 137-144.
- [6] YANG S. Genetic algorithms with memory and elitism-based immigrants in dynamic environments[J]. *Evolutionary Computation*, 2008, 16(3): 385-416.
- [7] YAO X, YANG S. Population-based incremental learning with associative memory for dynamic environments[J]. *IEEE Transactions on Evolutionary Computation*, 2008, 12(5): 542-561.
- [8] GOLDBERG D E, SMITH R E. Nonstationary function optimization using genetic algorithms with dominance and diploidy[A]. *Proceedings of the 2nd International Conference on Genetic Algorithms and Their Applications*[C]. Hillsdale: Lawrence Erlbaum Associates, 1987. 59-68.
- [9] MAVROVOUNIOTIS M, YANG S. Memory-based immigrants for ant colony optimization in changing environments[A]. *Proceedings of the 2011 International Conference on Applications of Evolutionary Computation-Volume Part I, EvoApplications'11*[C]. Springer-Verlag, Berlin, Heidelberg, 2011. 324-333.
- [10] BENDTSEN C N, KRINK T. Dynamic memory model for nonstationary optimization[A]. *Proceeding of the 2002 Congress on Evolutionary Computation*[C]. New York, USA, 2002. 145-150.
- [11] 周传华, 谢安世. 一种机遇动态小生境的自组织学习算法[J]. *软件学报*, 2011, 57(3): 255-264.
- ZHOU C H, XIE A S. Dynamic niche-based self-organizing learning algorithm[J]. *Journal of Software*, 2011, 22(8): 1738-1748.
- [12] YANG S. A clustering particle swarm optimizer for locating and tracking multiple optima in dynamic environments[J]. *IEEE Transactions on Evolutionary Computation*, 2010, 14(6): 959-974.
- [13] 吴晓军, 杨战中, 赵明. 均匀搜索粒子群算法[J]. *电子学报*, 2012, 40(6): 1115-1120.
- WU X J, YANG Z Z, ZHAO M. The convergence analysis of the uniform search particle swarm optimization[J]. *Acta Electronica Sinica*, 2012, 40(6): 1115-1120.
- [14] STORN R, PRICE K. Differential evolution—a simple and efficient heuristic for global optimization over continuous spaces[J]. *Journal of Global Optimization*, 1997, 11(4): 341-359.
- [15] BREST J, GREINER S, BOSKOVIC B, *et al.* Self-adapting control parameters in differential evolution: a comparative study on numerical benchmark problems[J]. *IEEE Transactions on Evolutionary Computation*, 2006, 10(6): 646-657.

作者简介:



刘星宝 (1977-), 男, 山东临沂人, 博士, 湖南商学院讲师, 主要研究方向为智能计算、投资组合优化。

殷建平 (1963-), 男, 湖南益阳人, 博士, 国防科学技术大学教授, 主要研究方向为人工智能、模式识别和信息安全。

胡春华 (1973-), 男, 湖南新化人, 博士, 湖南商学院教授, 主要研究方向为云计算, 电子商务。

陈荣元 [通信作者] (1976-), 男, 江苏泰州人, 博士, 湖南商学院高级研究员, 主要研究方向为图形图像处理。E-mail: chenrongyuan@126.com。