

基于精英区域学习的动态差分进化算法

彭 虎^{1,2}, 吴志健^{1,2}, 周新宇^{1,2}, 邓长寿³

(1. 武汉大学软件工程国家重点实验室, 湖北武汉 430072; 2. 武汉大学计算机学院, 湖北武汉 430072;
3. 九江学院信息科学与技术学院, 江西九江 332005)

摘 要: DE算法简单高效,但对复杂问题也存在收敛效率较低的问题,为提高DE算法的全局勘探能力和收敛精度,提出了一种新的精英区域学习动态差分进化算法,算法首先将历史精英保存在精英池中,然后采用正弦函数对精英池中的精英进行区域学习,最后利用动态DE模式有效提高收敛的速度,并从理论上证明了算法的收敛性.通过对包括单峰函数、多峰函数和偏移函数的20个基准测试函数的仿真实验和分析,验证了新算法的有效性和适用性,其能在保持较高的收敛速度的同时也能保持较好的收敛精度,经与多种知名的DE算法在统计学上的分析比较,证明了该算法是一种具有竞争力的新算法.

关键词: 差分进化; 精英池; 精英区域学习; 动态差分进化

中图分类号: TP18 **文献标识码:** A **文章编号:** 0372-2112 (2014)08-1522-09

电子学报 URL: <http://www.ejournal.org.cn> **DOI:** 10.3969/j.issn.0372-2112.2014.08.010

Dynamic Differential Evolution Algorithm Based on Elite Local Learning

PENG Hu^{1,2}, WU Zhi-jian^{1,2}, ZHOU Xin-yu^{1,2}, DENG Chang-shou³

(1. State Key Lab of Software Engineering, Wuhan University, Wuhan, Hubei 430072, China;

2. Computer School, Wuhan University, Wuhan, Hubei 430072, China;

3. School of Information Science and Technology, Jiujiang University, Jiujiang, Jiangxi 332005, China)

Abstract: DE algorithm is simple and efficient, but for complex problems also exist the problem of low efficiency of convergence. In order to improve the global exploration ability and convergence precision, this paper proposes a novel elite local learning dynamic differential evolution algorithm. Firstly the history elites are preserved in the elite pool, and then the elites in the pool conduct local learning by sine functions, finally dynamic DE model is used to effectively improve the speed of convergence, and the convergence of the algorithm is proved in theory. Algorithm has been tested on 20 benchmark functions including unimodal functions and multimodal functions and shift functions, experiments result verified the effectiveness and applicability, and the new algorithm can maintain higher convergence speed while maintaining better convergence accuracy. Comparison with the state-of-the-art DE in statistical analysis proves that the algorithm is a kind of new competitive algorithm.

Key words: differential evolution; elite pool; elite local learning; dynamic differential evolution

1 引言

差分进化算法(Differential Evolution, DE)简单、高效且容易实现,自Storn^[1]提出之后受到广泛关注,并长时间成为进化算法研究中的一个热点. DE已成功应用到如工业设计、工程优化和机器学习等多个领域并取得很好的效果^[2]. 国内外很多学者对DE算法进行了大量的改进^[2],算法的改进可分为三类,一是对差分变异策略进行研究^[3],提出新的变异策略或组合变异策略,其中Wang^[4]提出了一种经典的对变异策略和控制参数进行

随机组合的CoDE;二是研究DE的差分变异策略和控制参数对问题的适应性,提出了各种自适应算法, Qin^[5]提出一种自适应DE算法SaDE,算法根据历史经验自适应选择成功率较高的变异策略和参数,张晓伟^[6]提出了一种免缩放因子F的差分进化算法DEF,此类算法还有JADE^[7], jDE^[8];三是研究高效的区域搜索策略,搞高DE算法的收敛能力, Rahnamayan^[9]提出了反向差分进化算法ODE,在评价当前向量的同时也评价其反向向量, Wang^[10]对反向问题一般化提出了GODE. 针对学习对象

的选取问题,近年来基于精英信息的群智能算法陆续被提出,周新宇^[11]提出了基于精英的反向粒子群优化算法,Takahama^[12]提出了基于可行解精英集的差分进化算法用于求解约束优化问题.他们利用精英所携带的丰富信息,有效提高了算法的收敛速度和精度,但也存在易于陷入局部极值的问题.

为进一步拓展向量学习的区域,提高算法的收敛精度和收敛速度,本文提出了一种新的面向精英的区域学习动态差分进化算法(Elite Local Learning Dynamic Differential Evolution, ELDDE),算法首先采用精英池策略,池中保存有算法的历史最优向量,然后采用精英区域学习策略(Elite Local Learning, ELL),利用正弦函数对 DE 算法在迭代过程中产生的历史精英进行学习,学习的区间为从当前向量到其反向向量的整个搜索区域,从而更好的利用整个学习区域的信息,提高算法的收敛精度;接着算法采用了动态 DE 模式(Dynamic DE, DDE),在计算出每一个试验向量后立即与当前向量进行比较并对当前向量进行更新,提高算法的收敛速度.实验证明 ELDDE 算法简单高效、鲁棒性强,不仅对普通的单峰和多峰问题表现出非常强的收敛能力,而且对偏移的优化问题收敛效果也较其它算法要好.

2 标准 DE

作为一种基于随机搜索的群智能算法,DE 首先根据预设的种群大小和问题规模随机生成包含 N 个 D 维向量 $X_i = (x_{i,1}, x_{i,2}, \dots, x_{i,D})$ 的种群,然后迭代进行 DE 的三个基本操作:差分变异、交叉和选择,直到达到预设的停止条件,下面分别对标准 DE 算法^[1]的三个基本操作进行介绍.

2.1 差分变异

差分变异是 DE 最主要的操作,也是 DE 差异进化思想的体现,其根据当前向量 $X_{i,G}$,在种群中随机选择几个向量进行差分操作并产生一个差分向量 $V_{i,G}$,最常见的差分变异策略有以下五种:

(1) DE/rand/1

$$V_{i,G} = X_{i1,G} + F \cdot (X_{i2,G} - X_{i3,G}) \quad (1)$$

(2) DE/best/1

$$V_{i,G} = X_{best,G} + F \cdot (X_{i1,G} - X_{i2,G}) \quad (2)$$

(3) DE/current-to-best/1

$$V_{i,G} = X_{i1,G} + F \cdot (X_{best,G} - X_{i,G}) + F \cdot (X_{i1,G} - X_{i2,G}) \quad (3)$$

(4) DE/rand/2

$$V_{i,G} = X_{i1,G} + F \cdot (X_{i2,G} - X_{i3,G}) + F \cdot (X_{i4,G} - X_{i5,G}) \quad (4)$$

(5) DE/best/2

$$V_{i,G} = X_{best,G} + F \cdot (X_{i1,G} - X_{i2,G}) + F \cdot (X_{i3,G} - X_{i4,G}) \quad (5)$$

其中 i 表示当前向量的下标, G 表示第 G 次迭代, $i1, i2, i3, i4, i5$ 分别表示从当前种群中随机选择的 5 个不同的用于差分操作向量的下标, $X_{best,G}$ 表示当前种群中的最优向量, F 为缩放因子.除了上述常用的五种,还有学者提出的很多不同的差分变异策略,可以参考文献[1,3].

2.2 交叉操作

交叉操作主要是根据交叉概率 CR 对当前向量和差分向量进行重组,产生一个试验向量,如公式(6)所示,其中 $rand$ 是随机生成的介于 1 到 D 之间的整数.如果生成的随机数小于 CR 或是当前指示器 j 等于 $rand$ 则当前向量的第 j 维使用差分向量的值,否则用原值.

$$u_{i,j,G} = \begin{cases} v_{i,j,G}, & \text{if } rand_j(0,1) \leq CR \text{ or } j = rand \\ x_{i,j,G}, & \text{otherwise} \end{cases} \quad (6)$$

2.3 选择操作

当前向量种群所对应的试验向量全部产生后,两个向量群对应位置的向量根据适应度函数值进行优劣比较,获胜的进入下一代向量群进行新的迭代.

3 精英区域学习的动态差分进化

3.1 精英区域学习策略 ELL

反向学习策略是近年来兴起了的一个研究热点^[9,10],Rahnamayan^[9]将反向学习策略用于 DE 中并证明了其有效性,其主要方法为评估当前向量的同时也评估向量在搜索空间相反位置的反向向量.Wang^[10]进一步分析了反向学习的效果,提出了反向学习一般化处理的方法,如图 1 所示,反向向量可在当前向量的搜索区域所对应的反向搜索区域 $[L, U]$ 内取值,用动态搜索边界取代了固定搜索边界,保存了搜索经验,然而如图 1 所示,反向搜索过程中搜索的只是一个很小的区域,忽略掉了从当前向量 X 到反向向量 X^o 的全搜索区域 $[X, X^o]$.

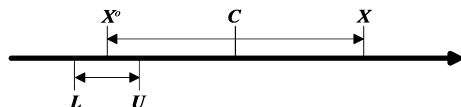


图1 ELL策略示意图

如图 1 所示,如果当前最优向量在几何中心 $[c, c, \dots, c]$,反向向量不断靠近的同时,全搜索区域中的大部分候选向量也在不断向最优向量靠近,那么将区域搜索的范围由反向搜索区域扩大到从当前向量到反向向量的全搜索区域,将能搜索到更多的更靠近最优向量的可行向量.

为此,引入三角函数用于全区域搜索,由于正弦函

数 \sin 和余弦函数 \cos 在定义域 $[-\pi, \pi]$ 的值域为 $[-1, 1]$, 反正弦函数 \arcsin 在定义域 $[-1, 1]$ 的值域为 $[-\pi/2, \pi/2]$, \sin 和 \cos 的取值与全区域搜索吻合, \arcsin 的取值将搜索区域放大了 1.57 倍, 分析发现它们都适合作为 ELL 策略的学习函数, 算法以 \sin 函数作为首选的学习函数. 正弦函数 $y = \sin(\alpha)$, $\alpha \in [-\pi, \pi]$, $y \in [-1, 1]$, 它的取值服从均匀分布, 在整个区域搜索范围内所有候选解的机会均等. 下面给出区域学习变异解的定义.

定义 1 区域学习变异解, 设 \mathbf{X} 为 D 维空间中的一个可行解, $\mathbf{X} = (x_1, x_2, \dots, x_D)$, 在搜索区域从 \mathbf{X} 到 \mathbf{X}° , 通过三角函数进行变异后, 对应的区域学习变异解 $\mathbf{X}^* = (x_1^*, x_2^*, \dots, x_D^*)$ 可定义为:

$$\mathbf{X}^* = \sin(\alpha) \cdot \mathbf{X} + r \cdot (\mathbf{U} + \mathbf{L}), \alpha \in [-\pi, \pi] \quad (7)$$

其中, r 为介于 0 和 1 之间的一个随机数, \mathbf{U} 和 \mathbf{L} 为当前种群搜索区域的上下界.

为了进行区域学习, 学习对象的选取是关键, 靠近实际最优解的向量能为区域学习提供丰富的信息, 而对那些远离实际最优解的向量进行学习可能降低算法的收敛速度, 为此我们的学习存在两种方案: 方案 A, 对当前种群中的普通向量进行学习. 方案 B, 对当前种群中的精英即最优解进行学习.

方案 A 的优点是能保持种群的多样性, 缺点是对种群中那些偏离最优解的向量的学习将导致收敛精度

和速度的下降. 方案 B 的优点是离实际最优解最近, 包含的有效信息量大, 缺点是当前最优解如果是局部极值, 对它的学习将导致种群多样性的丧失, 并容易陷入收敛停滞.

为了平衡两种方案, 提出了精英池的概念, 将历史最优解推入精英池中, 算法对池中的向量进行学习, 通过历史精英来保持学习的多样性, 如果当前精英是局部极值, 算法也可以通过对它其它历史精英的学习来逃离局部极值点. 这样既保证了学习的质量和效率, 也可避免因对单一向量学习而陷入局部极值. 图 2 给出了针对二维 Ackley 函数的搜索状态图, 其中小圆圈和黑色方框分别表示普通个体和精英个体, 图 2(a) 中给出了向量群的初始状态, 图 2(b)、(c)、(d) 中分别给出了第 5、10、15 次迭代后的状态, 其中包括 30 个普通个体, 5 个历史精英. 分析可以发现随着搜索进程的推进, 普通个体逐渐向最优解靠近, 历史精英有效分布于搜索空间中但比普通个体更靠近最优解, 在保持了种群多样性的同时能有效引导普通个体向最优解收敛. 下面给出精英区域学习变异解的定义.

定义 2 精英区域学习变异解, 设 \mathbf{X}^e 为当前迭代计算的精英解, $\mathbf{X}^e = (x_{1e}, x_{2e}, \dots, x_{De})$, 通过三角函数进行变异后, 对应的三解函数变异解 $\mathbf{X}^{e*} = (x_{1e*}, x_{2e*}, \dots, x_{De*})$ 可定义为:

$$\mathbf{X}^{e*} = \sin(\alpha) \cdot \mathbf{X}^e + r \cdot (\mathbf{U} + \mathbf{L}), \alpha \in [-\pi, \pi] \quad (8)$$

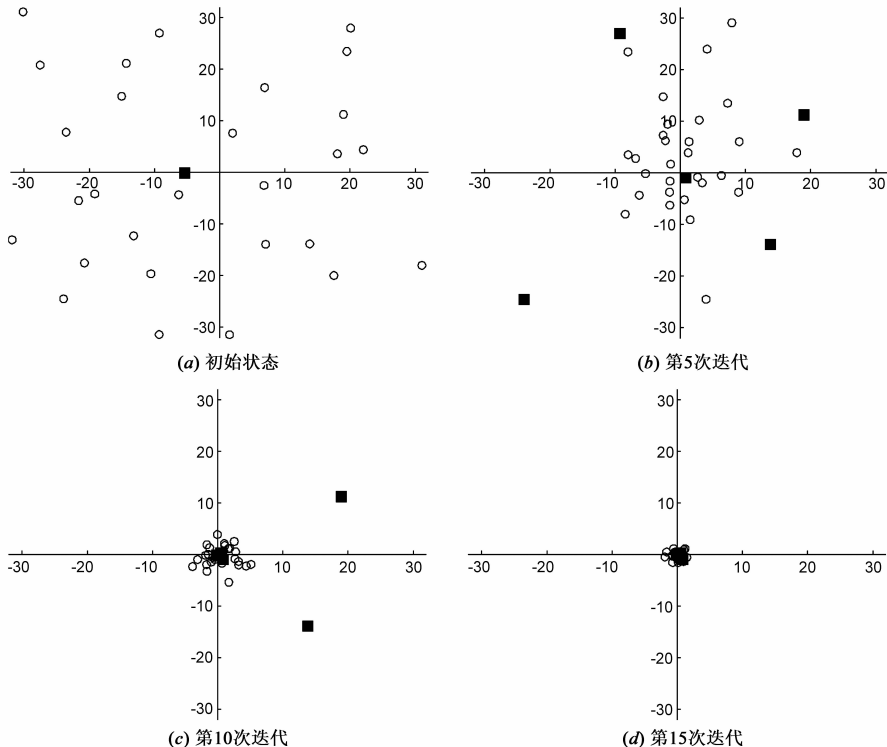


图2 精英个体与普通个体二维分布图

3.2 动态 DE

区域学习过程是对选定的向量进行局部寻优,而寻优时所耗费的迭代次数,必将在一定程度上减缓算法的收敛进程.动态 DE 可以有效的克服这一问题,加快算法的收敛速度.

标准 DE 算法的一个主要特征是算法在差分变异和交叉操作执行完后会产生一个和原向量群相同规模的试验向量群,原向量群和试验向量群中对应位置的向量通过适应度函数值进行一对一的比较,如果试验向量优于原向量,那么试验向量将在原向量群中取代对应位置的原向量.然而这种方案存在一个延缓收敛速度的问题,当前某个向量,如果优于原向量,但这个新的优质向量不能立即参与到其它向量的演化过程中,必需等到下一次迭代开始才能参与其中.

相对于标准 DE 算法,动态 DE 模式^[13]减去了试验向量群这一中间环节,每个向量根据差分变异和交叉操作产生新的向量后,立即与原向量进行比较.如果新向量优于原向量,则立即取代原向量更新到向量群中. DDE 加速了算法的收敛过程,任何新产生的优质向量能立即参与到其它向量的更新过程,不会出现迟滞的现象. ELDDE 算法采用这一模式来加速算法的收敛进程.

3.3 ELDDE 算法

算法 1 描述了 ELDDE,它和标准 DE 的框架基本相似,但主要有 3 个不同之处,在第(6)步,如果产生的随机数小于学习概率 LP,则从精英池选择一个精英进行区域学习,否则根据 DE/rand/1 执行标准的差分变异和交叉.在第(12)至(17)步,根据 DDE 模式,在每一个向量产生试验向量后,立即判断优劣并进行更新.在第(19)至(21)步,对精英池进行更新.

算法 1 ELDDE 算法

Input: NP: population size, LP: mutation probability

- (1) Randomly initialize population P
- (2) Evaluate the objective function value
- (3) FEs = NP
- (4) While FEs < MaxFEs
- (5) For $i = 1 : NP$
- (6) If $\text{rand}(0,1) < LP$
- (7) Execute ELL strategy according to Eq. (8)
- (8) Else
- (9) Execute the DE strategy DE/rand/1 and crossover according to Eq. (1) and Eq. (6)
- (10) End if
- (11) Evaluate the trial vector $U_{i,G}$
- (12) If $f(U_{i,G}) \leq f(X_{i,G})$
- (13) $X_{i,G+1} = U_{i,G}$
- (14) If $f(U_{i,G}) \leq f(X_{best,G})$
- (15) $X_{best,G+1} = U_{i,G}$

- (16) End if
 - (17) End if
 - (18) End for
 - (19) If $f(X_{best,G}) \leq f(X_{elite,G})$
 - (20) Update the elite pool
 - (21) End if
 - (22) End while
- Output: Global optimum

3.4 算法收敛性分析

Zaharie^[14]对收缩因子 F 添加了一些限定条件,从而证明了 DE 算法能以概率 1 收敛,贺毅朝^[15]通过分析 DE 的差分算子和选择算子,用随机压缩映射原理证明了 DE 是渐近收敛的.本文在这些收敛性分析的成果上,对 ELDDE 算法的收敛性进行分析.

引理 1 设 S 为解空间, $\Psi: \Omega \times S \rightarrow S$ 为 DE 形成的随机压缩算子,则 Ψ 是具有唯一性的随机不动点,即 DE 是渐近收敛的.

证明:见参考文献[15],略.

根据引理 1 可知 DE 算法具有渐近收敛性,ELDDE 算法是在 DE 算法的基础上增加了 ELL 策略,其试验向量的产生分为两种情况,一是通过 DE 的差分变异和交叉操作产生的,二是对历史精英进行区域学习产生的,依据 DE 的渐近收敛性可以证明 ELL 策略产生的向量也将渐近收敛到一个全局最优解,于是得到 ELDDE 的渐近收敛性定理如下.

定理 1 DE 算法具有渐近收敛性,则 ELDDE 算法也具有渐近收敛性.

证明:假设 DE 算法收敛到全局最优解 X^{gopt} ,则对于第 t 次迭代的个体 $X(t)$ 有:

$$\lim_{t \rightarrow \infty} X_i(t) = X_i^{gopt}, i \in (1, 2, \dots, D) \quad (9)$$

当前种群搜索区域的边界满足:

$$\lim_{t \rightarrow \infty} U_i(t) = \lim_{t \rightarrow \infty} L_i(t) = X_i^{gopt}, i \in (1, 2, \dots, D) \quad (10)$$

那么对于 ELL 策略产生的变异解 X^e 则有:

$$\begin{aligned} \lim_{t \rightarrow \infty} X_i^{e*} &= \lim_{t \rightarrow \infty} (\sin(\alpha) \cdot X_i^e + r \cdot (U + L)) \\ &= \sin(\alpha) \cdot \lim_{t \rightarrow \infty} X_i^e + r \cdot \lim_{t \rightarrow \infty} (U + L) \\ &= \sin(\alpha) \cdot X_i^{gopt} + 2 \cdot r \cdot X_i^{gopt} \\ &= (\sin(\alpha) + 2 \cdot r) \cdot X_i^{gopt} \end{aligned} \quad (11)$$

分析上述推理结果,可知精英区域学习的结果将收敛到 $(\sin(\alpha) + 2 \cdot r) \cdot X^{gopt}$,那么 ELDDE 算法中的所有向量不管是差分进化的个体,还是 ELL 策略产生的个体都是渐近收敛的.所以 ELDDE 算法也具有渐近收敛性.

3.5 算法复杂度分析

根据算法 1 中各步骤分析 ELDDE 的时间复杂度,其中种群规模为 NP,问题维数为 D ,迭代次数为 T .第(1)步种群初始化和第(2)步初始适应度函数评价分别

为 $O(NP \cdot D)$, 第(7)步 ELL 学习为 $O(D)$, 第(9)步 DE 差异变异和交叉操作为 $O(D)$, 第(11)步适应度函数评价和第(12)至(17)步向量更新为 $O(D)$, 则从第(5)到第(18)步的每一次迭代计算为 $O(NP \cdot D)$. 第(19)至(21)步中如果精英池规模与种群规模同为 NP, 将当前精英推入池的顶端, 其它历史精英后移所需时间复杂度为 $O(NP \cdot D)$, 则从第(4)到(22)步总的时间复杂度为 $O(T \cdot NP \cdot D)$. 综上所述, 略去低阶项后 ELDDE 算法总的时间复杂度为 $O(T \cdot NP \cdot D)$, 仅与 NP、D 和 T 相关. ELDDE 算法与标准 DE 算法的时间复杂度一致, 对算法的改进没有增加过多的计算开销.

4 仿真实验与分析

4.1 测试函数

为了验证 ELDDE 算法的有效性、正确性和适用性, 选择了 20 个标准测试函数进行实验分析. 其中 $f_1 \sim f_7$ 为单峰函数^[16], $f_8 \sim f_{13}$ 为多峰函数^[16], $f_{14} \sim f_{20}$ 为偏移函数^[17], 各函数如表 1 所示. ELDDE 算法中, 参数 NP = 30, F = 0.5, CR = 0.5, 区域学习概率 LP = 0.1, 其它各参与比较算法的参数与原文献相同, 最大函数评价次数 FE_s = 200000, 各算法独立运行 25 次.

为了更好的分析 ELDDE 的收敛速度、收敛强度和收敛精度等性能, 将 ELL 策略构建在标准 DE 之上形成 ELDE, 比较分析动态 DE 模式对算法收敛速度的贡献.

将 ELL 减去精英池策略形成 ELDDE-NEP, 比较分析精英池策略对提高种群多样性以逃离局部极值的贡献.

ELDDE 算法与 DE^[1], CODE^[10], ELDE, ELDDE-NEP 的平均最优值和方差在表 2 中, 为了公正客观的评价

表 1 基准测试函数

Type	Function	Name	$f(X^*)$	Initial Range
Unimodal Functions	f_1	Sphere	0	[-100, 100]
	f_2	Schwefel2.22	0	[-10, 10]
	f_3	Schwefel1.2	0	[-100, 100]
	f_4	Schwefel2.21	0	[-100, 100]
	f_5	Rosenbrock	0	[-30, 30]
	f_6	Step	0	[-100, 100]
	f_7	Quartic with Noise	0	[-1.28, 1.28]
Multimodal Functions	f_8	Schwefel2.26	-12569.5	[-500, 500]
	f_9	Rastrigin	0	[-5.12, 5.12]
	f_{10}	Ackley	0	[-32, 32]
	f_{11}	Griewank	0	[-600, 600]
	f_{12}	Penalized1	0	[-50, 50]
	f_{13}	Penalized2	0	[-50, 50]
Shift Functions	f_{14}	Shift Sphere	0	[-100, 100]
	f_{15}	Shift Schwefel1.2	0	[-100, 100]
	f_{16}	Shift Schwefel1.2 With noise	0	[-100, 100]
	f_{17}	Shift Griewank	0	[-600, 600]
	f_{18}	Shift Ackley	0	[-32, 32]
	f_{19}	Shift Penalized1	0	[-50, 50]
	f_{20}	Shift Penalized2	0	[-50, 50]

表 2 DE, CODE, ELDE, ELDDE-NEP, ELDDE 算法的平均值、标准差和 Wilcoxon 秩和检验结果

	DE 平均值 ± 方差	CODE 平均值 ± 方差	ELDE 平均值 ± 方差	ELDDE-NEP 平均值 ± 方差	ELDDE 平均值 ± 方差
f_1	9.66E-37 ± 2.80E-36 -	1.16E-39 ± 1.18E-39 -	0.00E+00 ± 0.00E+00 ≈	0.00E+00 ± 0.00E+00 ≈	0.00E+00 ± 0.00E+00
f_2	1.57E-20 ± 9.86E-21 -	5.25E-21 ± 2.59E-21 -	0.00E+00 ± 0.00E+00 ≈	0.00E+00 ± 0.00E+00 ≈	0.00E+00 ± 0.00E+00
f_3	1.21E+03 ± 6.17E+02 -	4.26E+03 ± 1.11E+03 -	0.00E+00 ± 0.00E+00 ≈	0.00E+00 ± 0.00E+00 ≈	0.00E+00 ± 0.00E+00
f_4	9.20E-02 ± 3.11E-01 -	2.66E-02 ± 1.67E-02 -	0.00E+00 ± 0.00E+00 ≈	0.00E+00 ± 0.00E+00 ≈	0.00E+00 ± 0.00E+00
f_5	2.21E+01 ± 3.77E-01 ≈	2.32E+01 ± 3.69E-01 -	2.30E+01 ± 3.66E-01 -	2.30E+01 ± 2.86E-01 -	2.23E+01 ± 3.88E-01
f_6	0.00E+00 ± 0.00E+00 ≈	0.00E+00 ± 0.00E+00 ≈	0.00E+00 ± 0.00E+00 ≈	0.00E+00 ± 0.00E+00 ≈	0.00E+00 ± 0.00E+00
f_7	5.92E-03 ± 1.49E-03 -	5.94E-03 ± 2.28E-03 -	1.92E-04 ± 1.25E-04 ≈	6.46E-05 ± 3.69E-05 +	1.51E-04 ± 1.10E-04
f_8	-5.35E+03 ± 3.80E+02 -	-5.08E+03 ± 3.50E+02 -	-9.56E+03 ± 2.28E+03 -	-1.13E+04 ± 4.09E+02 -	-1.25E+04 ± 9.98E+01
f_9	1.54E+02 ± 2.35E+01 -	1.34E+02 ± 2.08E+01 -	0.00E+00 ± 0.00E+00 ≈	0.00E+00 ± 0.00E+00 ≈	0.00E+00 ± 0.00E+00
f_{10}	6.25E-15 ± 1.52E-15 -	6.25E-15 ± 1.52E-15 -	0.00E+00 ± 0.00E+00 ≈	0.00E+00 ± 0.00E+00 ≈	0.00E+00 ± 0.00E+00
f_{11}	7.89E-04 ± 2.76E-03 ≈	0.00E+00 ± 0.00E+00 ≈	0.00E+00 ± 0.00E+00 ≈	0.00E+00 ± 0.00E+00 ≈	0.00E+00 ± 0.00E+00
f_{12}	4.15E-03 ± 2.03E-02 ≈	6.54E-21 ± 7.31E-21 -	1.31E-24 ± 1.17E-24 -	1.58E-32 ± 2.53E-34 ≈	1.57E-32 ± 5.47E-48
f_{13}	1.35E-32 ± 5.47E-48 ≈	1.42E-20 ± 9.62E-21 -	3.89E-03 ± 1.91E-02 -	8.87E-01 ± 8.00E-01 -	1.35E-32 ± 5.47E-48
f_{14}	0.00E+00 ± 0.00E+00 ≈	1.19E-20 ± 1.31E-20 -	3.34E-22 ± 3.40E-22 -	0.00E+00 ± 0.00E+00 ≈	0.00E+00 ± 0.00E+00
f_{15}	1.34E+03 ± 6.27E+02 -	3.65E+03 ± 1.48E+03 -	3.04E+00 ± 2.06E+00 -	2.63E-04 ± 1.96E-04 +	4.32E-01 ± 3.19E-01
f_{16}	1.58E+03 ± 1.20E+03 -	4.05E+03 ± 1.93E+03 -	5.04E+00 ± 3.07E+00 -	4.54E-01 ± 1.39E+00 -	4.53E-01 ± 2.86E-01
f_{17}	7.89E-04 ± 2.67E-03 ≈	2.96E-04 ± 1.45E-03 -	5.92E-04 ± 2.01E-03 ≈	3.16E-03 ± 3.91E-03 -	0.00E+00 ± 0.00E+00
f_{18}	6.39E-15 ± 1.42E-15 -	5.59E-11 ± 2.81E-11 -	1.31E-11 ± 4.35E-12 -	6.33E+00 ± 2.62E-01 -	4.97E-15 ± 1.74E-15
f_{19}	1.58E-32 ± 2.58E-34 ≈	6.57E-21 ± 8.15E-21 -	8.40E-25 ± 7.18E-25 -	1.58E-32 ± 2.53E-34 ≈	1.57E-32 ± 6.32E-35
f_{20}	1.67E-32 ± 1.55E-32 ≈	5.85E-20 ± 7.52E-20 -	4.39E-04 ± 2.15E-03 -	4.25E+00 ± 7.23E+00 -	1.35E-32 ± 5.47E-48
-	11	18	10	7	
+	0	0	0	2	
≈	9	2	10	11	

ELDDE 算法的性能,采用 Wilcoxon 秩和检验方法对实验结果进行统计分析,显著性水平为 0.05,在表的底部,符号“-”,“+”,“≈”分别表示劣于、优于和相当于 ELDDE 的结果.表 3 为 ELDDE 与知名的 DE 算法 CoDE^[4],SaDE^[5],JADE^[7],jDE^[8]的实验和统计结果.图

3 描述了各算法的收敛曲线,画图所用数据为各算法独立运行 25 次的第 15 次计算结果,受篇幅限制,仅从单峰函数、多峰函数和偏移函数中各选择了 2 个有代表性的函数.

表 3 CoDE, SaDE, JADE, jDE, ELDDE 算法的平均值、标准差和 Wilcoxon 秩和检验结果

	CoDE 平均值 + 方差	SaDE 平均值 + 方差	JADE 平均值 + 方差	jDE 平均值 + 方差	ELDDE 平均值 + 方差
f_1	6.42E-43 ± 1.90E-42 -	1.79E-96 ± 8.76E-96 -	5.79E-97 ± 2.83E-96 -	9.80E-130 ± 3.25E-129 -	0.00E+00 ± 0.00E+00
f_2	4.44E-23 ± 5.44E-23 -	5.41E-71 ± 1.81E-70 -	7.89E-53 ± 2.60E-52 -	9.60E-81 ± 4.25E-80 -	0.00E+00 ± 0.00E+00
f_3	8.15E-10 ± 1.00E-09 -	4.80E-03 ± 7.22E-03 -	4.29E-37 ± 1.22E-36 -	5.12E-07 ± 7.70E-07 -	0.00E+00 ± 0.00E+00
f_4	3.24E-10 ± 2.53E-10 -	4.77E-02 ± 1.60E-01 -	2.26E-11 ± 4.80E-11 -	1.74E+01 ± 5.62E+00 -	0.00E+00 ± 0.00E+00
f_5	5.33E-02 ± 7.97E-02 +	3.18E+01 ± 2.72E+01 ≈	4.15E+00 ± 1.71E+01 +	1.50E+00 ± 1.83E+00 +	2.23E+01 ± 3.88E-01
f_6	0.00E+00 ± 0.00E+00 ≈	0.00E+00 ± 0.00E+00 ≈	3.20E-01 ± 8.35E-01 -	8.00E-02 ± 2.71E-01 ≈	0.00E+00 ± 0.00E+00
f_7	4.02E-03 ± 1.52E-03 -	7.07E-03 ± 3.72E-03 -	2.44E-03 ± 1.19E-03 -	3.44E-03 ± 2.12E-03 -	1.51E-04 ± 1.10E-04
f_8	-1.26E+04 ± 1.82E-12 +	-1.25E+04 ± 9.48E+01 ≈	-1.22E+04 ± 1.94E+02 -	-1.25E+04 ± 6.77E+01 ≈	-1.25E+04 ± 9.98E+01
f_9	3.98E-02 ± 1.95E-01 ≈	9.15E-01 ± 8.40E-01 -	0.00E+00 ± 0.00E+00 ≈	7.96E-02 ± 2.70E-01 ≈	0.00E+00 ± 0.00E+00
f_{10}	3.55E-15 ± 0.00E+00 -	9.77E-01 ± 7.96E-01 -	1.44E-01 ± 3.97E-01 -	6.11E-15 ± 1.60E-15 -	0.00E+00 ± 0.00E+00
f_{11}	7.89E-04 ± 2.67E-03 ≈	2.20E-02 ± 1.76E-02 -	2.27E-03 ± 4.40E-03 -	1.97E-03 ± 4.06E-03 -	0.00E+00 ± 0.00E+00
f_{12}	1.57E-32 ± 5.47E-48 ≈	1.38E-01 ± 3.51E-01 -	4.56E-02 ± 1.41E-01 ≈	8.29E-03 ± 2.81E-02 -	1.57E-32 ± 5.47E-48
f_{13}	1.35E-32 ± 5.47E-48 ≈	1.28E-01 ± 4.33E-01 -	4.39E-04 ± 2.15E-03 ≈	4.39E-03 ± 1.94E-02 -	1.35E-32 ± 5.47E-48
f_{14}	0.00E+00 ± 0.00E+00 ≈	2.96E-32 ± 6.39E-32 -	0.00E+00 ± 0.00E+00 ≈	2.96E-32 ± 5.92E-32 -	0.00E+00 ± 0.00E+00
f_{15}	6.72E-10 ± 1.37E-09 +	2.40E-03 ± 3.74E-03 +	1.21E-30 ± 6.59E-31 +	1.74E-07 ± 2.70E-07 +	4.32E-01 ± 3.19E-01
f_{16}	2.82E-08 ± 4.68E-08 +	4.12E+01 ± 6.94E+01 -	1.52E+02 ± 2.89E+02 -	2.75E+01 ± 1.09E+02 ≈	4.53E-01 ± 2.86E-01
f_{17}	0.00E+00 ± 0.00E+00 ≈	1.18E-02 ± 2.36E-02 -	5.61E-03 ± 8.65E-03 -	2.76E-03 ± 4.64E-03 -	0.00E+00 ± 0.00E+00
f_{18}	5.55E-15 ± 0.00E+00 -	7.57E-01 ± 7.09E-01 -	8.35E-02 ± 2.85E-01 ≈	7.25E-15 ± 3.09E-15 -	4.97E-15 ± 1.74E-15
f_{19}	1.57E-32 ± 5.47E-48 ≈	4.15E-03 ± 2.03E-02 ≈	1.57E-32 ± 5.47E-48 ≈	3.32E-02 ± 1.05E-01 -	1.57E-32 ± 6.32E-35
f_{20}	1.35E-32 ± 5.47E-48 ≈	1.61E-01 ± 7.08E-01 -	8.79E-04 ± 2.98E-03 ≈	7.41E-32 ± 2.09E-31 -	1.35E-32 ± 5.47E-48
-	7	15	11	14	
+	4	1	2	2	
≈	9	4	7	4	

4.2 实验结果比较分析

对表 2 和表 3 分析可发现,ELDDE 算法的收敛能力较强,在 $f_1, f_2, f_3, f_4, f_6, f_9, f_{10}, f_{11}, f_{14}, f_{17}$ 共 10 个函数上的优势明显,可以很快并稳定的收敛到最优解 0. ELDDE 算法在收敛精度和收敛能力方面,与各算法相比占有较大优势.

在表 2 中标准 DE 算法在 $f_3, f_8, f_9, f_{15}, f_{16}$ 出现了停滞,而 ELDDE 则在 f_3, f_9 收敛到了最优解, f_8, f_{15}, f_{16} 则接近最优解.对于一般化的反向搜索算法 GODE, ELDDE 占有全面的优势,且收敛的精度方面除 f_5, f_6, f_{11} 之外都大于多个数量级,说明将学习范围从反向区域扩大到全区域的重要意义.

相对于在标准 DE 上增加了 ELL 策略的 ELDE,分析可发现 ELDE 对于各测试函数具有较强的收敛能力,但在相同的函数评价次数的情况下,它的收敛精度要弱于 ELDDE,说明 DDE 模式提高了收敛的速度.对于在 ELDDE 上去掉精英池策略的 ELDDE-NEP,可发现它的

收敛能力也较强,但对于部分函数如 $f_{13}, f_{17}, f_{18}, f_{20}$ 则存在收敛停滞的现象,说明仅仅对当前精英进行学习对于多峰问题和偏移问题容易陷入了局部极值,这证明了精英池对保持种群多样性方面的贡献,对于历史精英的不断学习有助于算法及时跳出局部极值. ELDDE-NEP 在 f_7 和 f_{15} 上的收敛效果好于 ELDDE,因为 f_7 是单峰问题, f_{15} 是单峰的偏移问题, ELDDE-NEP 对当前精英进行学习是一种贪婪的学习策略,其对此类问题的收敛速度比 ELDDE 要快.

从表 3 底部的 Wilcoxon 秩和检验的统计结果可以看出,ELDDE 算法相对于近几年出现的在国际上知名的 DE 算法具有明显的优势,对于单峰函数 $f_1 \sim f_7$, ELDDE 除 f_5 和 f_7 外都收敛到最优解 0,但对函数 f_5 ELDDE 要弱于 CoDE, JADE, jDE. 对于多峰函数 $f_8 \sim f_{13}$, ELDDE 与各算法相当,但 ELDDE 对于 f_8, f_9, f_{10} 收敛到了最优解 0. 对于偏移函数 $f_{14} \sim f_{20}$, ELDDE 在 f_{14} 和 f_{17} 收敛到了最优解, f_{18}, f_{19}, f_{20} 都优于或相当于各算法, f_{15} 要劣于

各算法, f_{16} 要劣于 CoDE.

从图 3 可以看出 ELDDE 算法相对于其它算法具有较强的收敛能力, 在 f_1, f_2, f_9 上能快速的收敛到最优解 0, 在 f_8, f_{19}, f_{20} 上与其它算法的收敛曲线吻合. 表 4 给出了 ELDDE 和各算法的 Friedman 检验排名, ELDDE 排在第一, 从统计学上说明了 ELDDE 算法的优势.

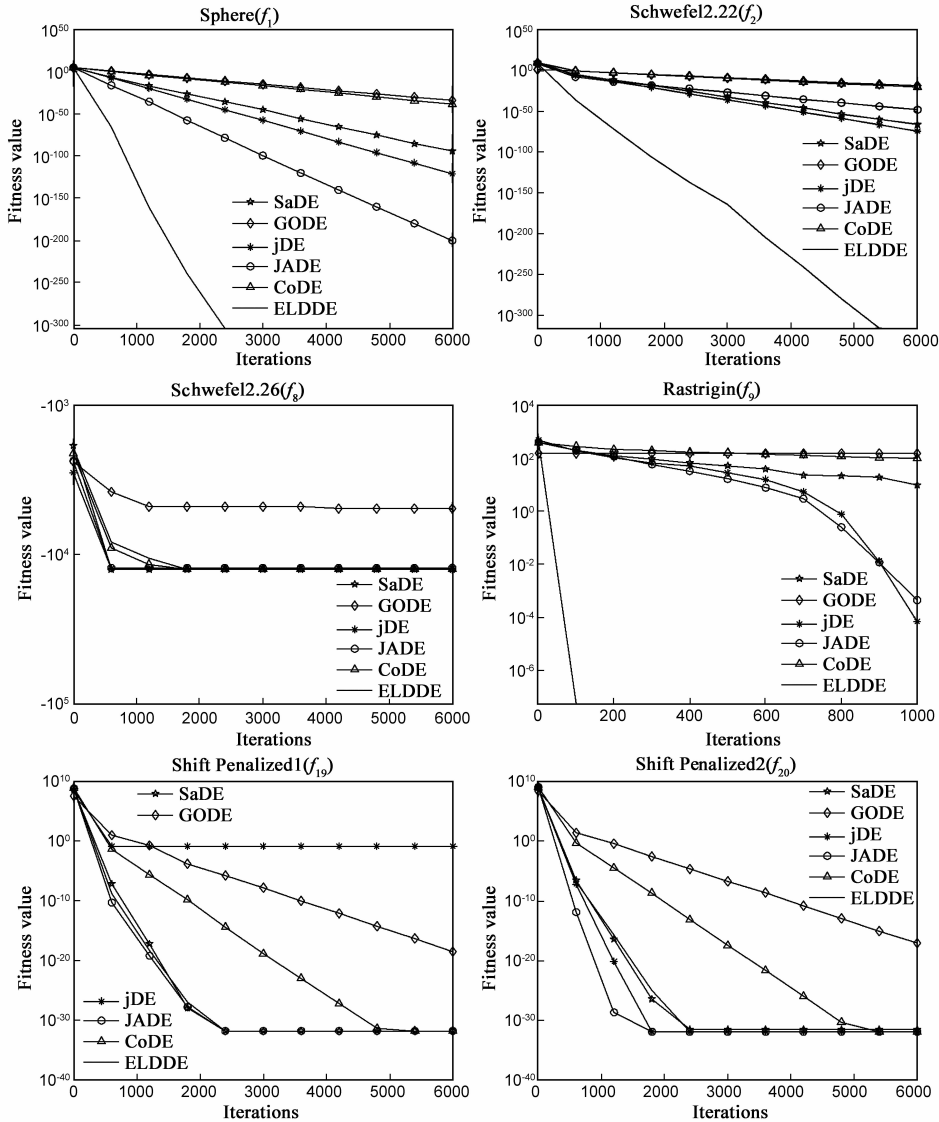


图3 各算法在函数 $f_1, f_2, f_8, f_9, f_{19}, f_{20}$ 上的收敛曲线

4.3 精英学习的可行性分析

表 5 给出了在区域学习的 DDE 算法框架下对精英个体和普通个体分别进行学习的实验结果, 为了全面的比较两者的性能, 分别记录了函数评价次数达到 20 万次和 10 万次时的平均值. 对表 5 分析可发现, 精英学习的收敛速度和精度要优于普通学习, 特别是复杂的偏移问题.

对于单峰问题 $f_1 \sim f_7$ 和多峰问题 $f_8 \sim f_{13}$, 精英学习和普通学习的精度基本一致, 在 f_5 和 f_7 上稍有差异, 在

综合以上实验分析可以说明 ELDDE 算法具有较快的收敛速度、较强的收敛精度和较好的函数适用性, 有效提高了对复杂问题的收敛效率.

表 4 各算法的 Friedman 检验排名

算法	SaDE	GODE	jDE	JADE	CoDE	ELDDE
排名	4.90	4.55	3.68	3.60	2.30	1.98

f_{13} 上精英学习的效果好于普通学习, 但对比函数评价次数达到 10 万次时的结果, 可发现精英学习的收敛速度要更快, 能在较少的函数评价次数达到更高的精度. 对于带偏移的问题 $f_{14} \sim f_{20}$, 精英学习在收敛速度和精度两方面都要好于普通学习. 说明精英池策略在解决复杂问题时, 对保持学习对象的多样性方面发挥了重要作用, 能让算法及时跳出局部极值, 实验结果证明了精英学习以及精英池策略的可行性和有效性.

表 5 精英学习与普通学习的实验结果

FES		f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}
2E5	精英	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.23E+01	0.00E+00	1.51E-04	-1.25E+04	0.00E+00	0.00E+00
	普通	0.00E+00	0.00E+00	0.00E+00	0.00E+00	2.24E+01	0.00E+00	7.92E-05	-1.25E+04	0.00E+00	0.00E+00
1E5	精英	0.00E+00	4.41E-180	0.00E+00	9.94E-181	2.47E+01	0.00E+00	3.44E-04	-1.25E+04	0.00E+00	0.00E+00
	普通	1.98E-235	3.71E-122	4.07E-174	1.30E-102	2.58E+01	0.00E+00	3.32E-04	-1.08E+04	0.00E+00	0.00E+00
FES		f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}	f_{19}	f_{20}
2E5	精英	0.00E+00	1.57E-32	1.35E-32	0.00E+00	4.32E-01	4.53E-01	0.00E+00	4.97E-15	1.57E-32	1.35E-32
	普通	0.00E+00	1.57E-32	4.39E-04	7.89E-33	1.58E+01	2.43E+01	1.78E-03	5.26E-15	1.57E-32	1.35E-32
1E5	精英	0.00E+00	1.57E-32	1.35E-32	0.00E+00	1.28E+01	1.37E+01	0.00E+00	6.25E-15	1.57E-32	1.35E-32
	普通	0.00E+00	1.25E-30	4.39E-04	4.38E-28	2.32E+02	2.84E+02	1.78E-03	2.19E-14	8.99E-31	3.00E-28

4.4 DDE 的有效性分析

为了测试 DDE 模式对算法收敛速度的影响,在表 7 中给出了 ELDE 和 ELDDE 达到如表 6 所示的精度所用的 25 次独立实验所得的平均迭代次数及加速比.如表 2 所示,因为 ELDDE 的收敛精度比 ELDE 高,表 6 所用的精度以 ELDE 能达到的精度为基准进行设定.对表 7

分析可发现,ELDDE 和 ELDE 对单峰函数的平均迭代次数基本相同,但在多峰函数和偏移函数的加速比较为明显,特别是在 $f_{12}, f_{13}, f_{14}, f_{17}, f_{18}, f_{19}, f_{20}$ 都超过了四倍,显示了 DDE 模式在 ELL 算法框架下提高了对复杂问题求解的收敛速度,证明了 DDE 在 ELDDE 算法中的适用性和正确性.

表 6 函数预设精度

函数	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}
精度	0	0	0	0	1E+2	0	1E-3	-1E+4	0	0
函数	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}	f_{19}	f_{20}
精度	0	1E-20	1E-2	1E-20	1E+1	1E+1	1E-3	1E-10	1E-20	1E-3

表 7 ELDDE 与 ELDE 达到预设精度的平均迭代次数及加速比

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}
ELDDE	2665.96	5420.4	2741.64	5498.56	17.2	22.68	1477.92	1024.88	131.84	286.52
ELDE	2749.24	5449.96	2803.84	5481.64	17.96	23.88	1773.04	2298.76	152.68	292.68
加速比	1.03	1.01	1.02	1.00	1.04	1.05	1.20	2.24	1.16	1.02
	f_{11}	f_{12}	f_{13}	f_{14}	f_{15}	f_{16}	f_{17}	f_{18}	f_{19}	f_{20}
ELDDE	151.32	1345.64	230.92	1508.36	3529.84	3513.96	338.52	1471.88	1349.92	346.12
ELDE	165.76	5511.04	961.8	6183.48	4893.04	5296.72	1484.64	6131	5456.44	1439.8
加速比	1.10	4.10	4.17	4.10	1.39	1.51	4.39	4.17	4.04	4.16

5 结论

本文提出了一种新的面向精英进行区域学习的动态差分进化算法 ELDDE,在 ELDDE 算法中,DDE 模式有效提高了收敛速度,精英池策略有效保持了学习对象的多样性,通过正弦函数对历史精英进行区域学习增强了算法的收敛精度,实验数据和统计分析证明了 ELDDE 算法较强的收敛能力以及各参数的适用性.后续工作是将 ELL 策略推广到其它的群智能算法中并用于解决工程优化等实际问题.

参考文献

- [1] 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.
- [2] Das S, Suganthan P N. Differential evolution: A survey of the state-of-the-art [J]. IEEE Transactions on Evolutionary Computation, 2011, 15(1): 4-31.
- [3] Mezura-Montes E, Velázquez-Reyes J, et al. A comparative study of differential evolution variants for global optimization [A]. Proc of the 8th Annual Conference on Genetic and Evo-

- lutionary Computation[C]. Seattle: ACM, 2006. 485 – 492.
- [4] Wang Y, Cai Z, et al. Differential evolution with composite trial vector generation strategies and control parameters[J]. IEEE Transactions on Evolutionary Computation, 2011, 15(1): 55 – 66.
- [5] Qin A K, Huang V L, et al. Differential evolution algorithm with strategy adaptation for global numerical optimization[J]. IEEE Transactions on Evolutionary Computation, 2009, 13(2): 398 – 417.
- [6] 张晓伟, 刘三阳. 免比例因子 F 的差分进化算法[J]. 电子学报, 2009, 36(6): 1318 – 1323.
Zhang Xiao-wei, Liu San-yang. Differentialevolution without the scale factor F[J]. Acta Electronica Sinica. 2009, 36(6): 1318 – 1323. (in Chinese)
- [7] Zhang J, Sanderson A C. JADE: adaptive differential evolution with optional external archive[J]. IEEE Transactions on Evolutionary Computation, 2009, 13(5): 945 – 958.
- [8] Brest J, Greiner S, 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.
- [9] Rahnamayan S, Tizhoosh H R, et al. Opposition-based differential evolution[J]. IEEE Transactions on Evolutionary Computation, 2008, 12(1): 64 – 79.
- [10] Wang H, Wu Z, et al. Enhanced opposition-based differential evolution for solving high-dimensional continuous optimization problems[J]. Soft Computing, 2011, 15(11): 2127 – 2140.
- [11] 周新宇, 吴志健, 等. 一种精英反向学习的粒子群优化算法[J]. 电子学报, 2013, 41(8): 1647 – 1652.
Zhou Xin-yu, Ww Zhi-jian. Elite opposition based particle swarm optimization[J]. Acta Electronica Sinica. 2013, 41(8): 1318 – 1323. (in Chinese)
- [12] Takahama T, Sakai S. Constrained optimization by the ϵ constrained differential evolution with gradient-based mutation and feasible elites[A]. Proc of IEEE Congress on Evolutionary Computation[C]. Vancouver: IEEE, 2006. 1 – 8.
- [13] Tagawa K. A statistical study of the differential evolution based on continuous generation model[A]. Proc of IEEE Congress on Evolutionary Computation [C]. Trondheim: IEEE, 2009. 2614 – 2621.
- [14] Zaharie D. Critical values for the control parameters of differential evolution algorithms [A]. Proc of Mendel 2002, 8th International Conference on Soft Computing [C]. Brno: Faculty of Mechanical Engineering BUT, 2002. 62 – 67.
- [15] 贺毅朝, 王熙照, 等. 差分演化的收敛性分析与算法改进[J]. 软件学报, 2010, 21(5): 875 – 885.
He Yi-chao, Wang Xi-zhao. Convergent analysis and algorithmic improvement of differential evolution[J]. Journal of Software, 2010, 21(5): 875 – 885. (in Chinese)
- [16] Yao X, Liu Y, et al. Evolutionary programming made faster [J]. IEEE Transactions on Evolutionary Computation, 1999, 3(2): 82 – 102.
- [17] Wang H, Wu Z, et al. Space transformation search: a new evolutionary technique [A]. Proc of the first ACM/SIGEVO Summit on Genetic and Evolutionary Computation [C]. New York: ACM, 2009. 537 – 544.

作者简介



彭 虎 男, 1981 年生, 武汉大学计算机学院博士研究生, 研究方向: 智能计算及其在软件工程中的应用.

E-mail: hu_peng@whu.edu.cn



吴志健 男, 1963 年生, 教授, 博士生导师, 武汉大学软件工程国家重点实验室副主任, 研究方向: 智能计算、并行计算和智能信息处理.

E-mail: zhijianwu@whu.edu.cn

周新宇 男, 1987 年生, 武汉大学计算机学院博士研究生, 研究方向: 智能计算、并行计算.

邓长寿 男, 1972 年生, 博士, 教授, 九江学院信息科学与技术学院副院长, 研究方向: 智能计算和数据挖掘.