郭婉青,叶东毅.帝国竞争算法的进化优化[J].计算机科学与探索,2014,8(4):473-482.

ISSN 1673-9418 CODEN JKYTA8
Journal of Frontiers of Computer Science and Technology
1673-9418/2014/08(04)-0473-10
doi: 10.3778/j.issn.1673-9418.1307016

E-mail: fcst@vip.163.com
http://www.ceaj.org
Tel: +86-10-89056056

帝国竞争算法的进化优化*

郭婉青+,叶东毅
福州大学 数学与计算机科学学院,福州 350108

Evolutionary Optimization of Imperialist Competitive Algorithm*

GUO Wanqing+, YE Dongyi
College of Mathematics and Computer Science, Fuzhou University, Fuzhou 350108, China
+ Corresponding author: E-mail: 252698387@qq.com

GUO Wanqing, YE Dongyi. Evolutionary optimization of imperialist competitive algorithm. Journal of Frontiers of Computer Science and Technology, 2014, 8(4): 473-482.

Abstract: To deal with the problem of premature convergence and low precision of the traditional imperialist competitive algorithm (ICA), this paper proposes two improved ICAs based on biological evolution. In the traditional ICA, colony revolution will lead to low precision because the operator may make the strong colony lost. To overcome this shortcoming, a differential evolution operator is introduced, which makes use of the interaction among colonies to produce new colonies. The operator will enhance the population diversity and keep the excellent individuals at the same time. Furthermore, on account of strengthening the interaction among empires, a clone evolution operator is introduced, which includes the following steps: clonal reproduction of the stronger countries; high frequency variation and random crossover of clonal populations; the stronger countries take place of the weaker ones. The operator can guide the search for global optimum efficiently. The proposed methods are applied to six benchmark functions and six typical complex function optimization problems, and the performance comparison of the proposed methods with other ICAs is experimented. The results indicate that the proposed methods can significantly speed up the convergence and improve the precision and stability.

Key words: imperialist competitive algorithm; premature convergence; differential evolution; clone evolution

---

* The National Natural Science Foundation of China under Grant No. 71231003 (国家自然科学基金); the Natural Science Foundation of Fujian Province of China under Grant No. 2012J01262 (福建省自然科学基金).

Received 2013-06, Accepted 2013-08.

CNKI网络优先出版:2013-09-30, http://www.cnki.net/kcms/detail/11.5602.TP.20130930.0914.001.html

摘 要:为了改善帝国竞争算法(imperialist competitive algorithm,ICA)易早熟收敛、精度低等缺点,提出了两种基于生物进化的改进ICA算法。针对殖民地改革算子可能使势力较强的殖民地丢失,导致寻优精度降低的不足,引入了一种微分进化算子,利用殖民地之间的信息交互产生新的殖民地,在增强群体多样性的同时保留了优秀个体。另外,针对帝国之间缺乏有效的信息交互这一情况,引入了克隆进化算子,对势力较强的国家进行克隆繁殖,并经过克隆群体的高频变异和随机交叉,选择势力较强的国家取代势力较弱的国家,从而有效地引导算法向最优解方向搜索。将算法应用于6个基准函数和6个经典复合函数优化问题,并与其他ICA改进算法进行比较,结果表明,基于生物进化的ICA算法在收敛精度、收敛速度及稳定性上有显著提高。

关键词:帝国竞争算法;早熟收敛;微分进化;克隆进化

文献标志码:A **中图分类号:**TP18

1 引言

帝国竞争算法(imperialist competitive algorithm,ICA)[1]是Atashpaz-Gargari和Lucas于2007年提出的一种基于帝国主义殖民竞争机制的进化算法,属于社会启发的随机优化搜索方法。目前,ICA已被成功应用于多种优化问题中,如调度问题[2-5]、分类问题[6-7]和机械设计问题[8-9]等。ICA的收敛速度快,但是容易陷入局部最优。为了克服这一问题,相关学者已提出了很多改进算法。例如,Bahrami等人利用混沌映射决定同化算子中殖民地的移动方向[10];Zhang等人在殖民地移动之后,随机选择一部分殖民地进行位置更新[11];Lin等人提出了扰动ICA算法(perturbed ICA,PICA)[12],并且利用人工的帝国主义国家取代较弱的帝国主义国家,以此来增强帝国之间的信息交互(ICA using artificial imperialist,ICAAI)[13]。在文献[14]中,ICA算法的提出者Atashpaz-Gargari给出了改进的算法源码,本文将此改进算法记为OICA(original ICA)。这些改进算法都只是在一定程度上提高了算法的局部寻优能力,早熟现象仍然存在,在求解精度和计算效率上有待进一步改进。

早期的ICA改进算法中,殖民地改革算子可能使势力较强的殖民地丢失,导致寻优精度降低。针对这个不足,本文借鉴微分进化思想,引入一种微分进化算子,利用殖民地之间的信息交互产生新的殖民地,在增强群体多样性的同时保留了优秀个体。另外,帝国之间缺乏有效的信息交互,针对这个情况,本文借鉴免疫算法思想,引入克隆进化算子,增强了帝国之间的交互能力,有效地引导算法向最优解方向搜索。在6个基准优化函数和6个国际上通用的复合优化函数上进行测试,实验结果表明,针对高维优化问题,本文算法具有很好的全局寻优能力和求解精度,与现有的ICA改进算法相比,具有明显的优势。

2 帝国竞争算法

为便于描述,首先简要介绍基本的ICA过程。最原始的ICA主要包括以下几个部分:产生初始帝国,同化机制,竞争机制,帝国灭亡。

2.1 产生初始帝国

ICA的个体是国家,相当于遗传算法中的染色体,对于一个N维的优化问题,国家可以表示成如下形式:

$$country=[p_1,p_2,\cdots,p_N]$$

国家的势力大小通过代价函数来衡量:

$$cost=f(country)=f(p_1,p_2,\cdots,p_N)$$

国家的势力和代价函数值成反比,即代价函数值越小,国家势力越大。初始帝国的产生分为以下几个步骤:

首先,随机产生 N_{pop} 个国家,从中选出势力较大的前 N_{imp} 个国家作为帝国主义国家,剩下的 N_{col} 个国家作为殖民地。

其次,根据帝国主义国家的势力大小划分殖民地。每个帝国的殖民地个数按照式(1)~(3)计算:

$$C_n=c_n-\max_i\{c_i\} \tag{1}$$

$$p_n=\left|\frac{C_n}{\sum_{i=1}^{N_{imp}}C_i}\right| \tag{2}$$

$$N.C._n = round\{p_n \times N_{col}\} \quad (3)$$

其中，c_n 是第 n 个帝国主义国家的代价函数值；C_n 是它的标准化代价；p_n 是它的标准化势力大小；$N.C._n$ 是第 n 个帝国的初始殖民地个数。

最后，对于每个帝国主义国家，从 N_{col} 个殖民地中随机选择相应的个数分配给它，最终形成初始的 N_{imp} 个帝国。

2.2 同化机制

帝国主义国家为了更好地控制其殖民地国家，将自己的思想模式及文化风俗推广到殖民地国家的过程，称为同化。ICA中通过所有殖民地向其所属帝国主义国家移动来模拟同化过程。

殖民地向帝国主义国家移动的距离 x 定义如下：

$$x \sim U(0, \beta \times d)$$

其中，$\beta > 1$，d 是殖民地国家和帝国主义国家之间的距离。

同时，为了扩大搜索范围，增加了一个偏移方向 θ，其定义如下：

$$\theta \sim U(-\gamma, \gamma)$$

其中，$0 < \gamma < \pi$，用来调整殖民地的移动方向。

当一个殖民地国家移动到一个新的位置后，殖民地的代价函数值可能比帝国主义国家小，也就是说殖民地的势力更大。此时，交换殖民地和帝国主义国家的位置，即殖民地成为该帝国的帝国主义国家，而原来的帝国主义国家则沦为殖民地。

2.3 竞争机制

帝国竞争机制模拟的是现实社会中势力较强的帝国占有并控制势力较弱帝国的殖民地的过程。首先，需要计算帝国的总代价函数值，即势力大小。帝国主义国家对整个帝国的势力影响较大，而殖民地国家的影响非常小，因此ICA采用如下公式计算一个帝国的总代价：

$$T.C._n = f(imp_n) + \xi \times \frac{\sum_{i=1}^{N.C._n} f(col_i)}{N.C._n} \quad (4)$$

其中，imp_n 是第 n 个帝国的帝国主义国家；$T.C._n$ 是第 n 个帝国的总代价；$0 < \xi < 1$，ξ 的大小决定了殖民地国家对整个帝国势力的影响程度。

选择最弱的帝国中最弱的殖民地作为帝国竞争的对象，势力越大的帝国越有可能占有该殖民地。

2.4 帝国灭亡

帝国之间的竞争，使势力大的帝国通过占有其他帝国的殖民地变得越来越强大，而势力弱的帝国殖民地个数不断减少，当一个帝国丢失所有的殖民地时，帝国覆灭。随着帝国的灭亡，最终剩下一个帝国，此时算法终止。

比如这里就是说的我的ICAmatlab代码中的revolve操作

3 基于生物进化的改进ICA

3.1 基于微分进化的改进ICA

在现有的ICA改进算法中，为了提高群体的多样性，引入了殖民地改革操作，其做法是随机选择一部分殖民地更新位置，这就可能使得势力较强的殖民地丢失，导致寻优精度降低。因此，本文借鉴微分进化思想，引入一种微分进化算子，利用殖民地之间的信息交互产生新的殖民地，在增强群体多样性的同时保留了优秀个体。

基于微分进化的改进ICA算法步骤如下：

(1)根据式(1)、(2)、(3)初始化帝国；
(2)当不满足算法终止条件时，循环进行第3~10步；
(3)对每个帝国 i，循环进行第4~7步；
(4)对帝国 i 内的殖民地进行同化操作；
(5)对帝国 i 内的殖民地进行微分进化；
(6)计算帝国 i 内殖民地的代价函数值；
(7)更新帝国 i 的帝国主义国家；
(8)根据式(4)计算帝国的总代价；
(9)相似帝国的合并操作；
(10)算法终止。

利用微分进化替换殖民地改革

下面讨论第5步微分进化的具体做法。

①每一个殖民地 Col 以 MR 的概率根据式(5)进行微分变异：

$$C = Col_{r_3} + F(Col_{r_1} - Col_{r_2}) + rand(imp_{best} - Col) \quad (5)$$

其中，Col_{r_1}、Col_{r_2}、Col_{r_3} 是随机选择的3个殖民地；F 是缩放因子；imp_{best} 是最好的帝国主义国家，即全局最优值；MR 是变异概率，适当的 MR 取值可以提高算

法的时间效率。

②对每一维度根据式(6)进行微分交叉：

$$D_i=\begin{cases}C_i, \text{if } rand<CR\\ Col_i, \text{otherwise}\end{cases} \tag{6}$$

其中，CR是交叉概率；$rand$是[0,1]之间的随机数。

③选择采用贪婪策略，当新产生的殖民地D的势力大于原来殖民地的势力，即$cost(D)<cost(Col)$时，更新殖民地位置。

本文将基于微分进化的改进ICA称为ICADE（ICA based on differential evolution）。

3.2 基于克隆进化的改进ICA

在现有的ICA改进算法中，帝国竞争操作体现了帝国之间的信息交互，然而帝国竞争在每一次迭代中只是将最弱的殖民地归于最强的帝国，该过程对每个帝国的势力大小影响很小，需要多次迭代才能体现出来，帝国之间缺乏更有效的信息交互。因此，为了提高帝国之间的信息交互能力，借鉴免疫算法思想，引入克隆进化算子。

基于克隆进化的改进ICA算法步骤如下：

(1)根据式(1)、(2)、(3)初始化帝国；

(2)当不满足算法终止条件时，循环进行第3~11步；

(3)对每个帝国i，循环进行第4~7步；

(4)对帝国i内的殖民地进行同化操作；

(5)殖民地改革操作；

(6)计算帝国i内殖民地的代价函数值；

(7)更新帝国i的帝国主义国家；

(8)根据式(4)计算帝国的总代价；

(9)相似帝国的合并操作；

(10)帝国之间的克隆进化操作；

(11)算法终止。

下面讨论第10步克隆进化的具体做法。

①克隆。将当前的m个帝国主义国家作为代克隆群体，并根据势力大小从大到小进行排序，每个帝国主义国家的克隆个数NC_i按照式(7)进行计算：

$$NC_i=floor(\lambda\times NCol_i) \tag{7}$$

其中，λ是克隆系数；$NCol_i$是第i个帝国的殖民地个数，势力越大的帝国主义国家克隆群体个数越多。

②变异。第i个帝国的克隆群体Clo_i根据式(8)~(9)进行变异，产生变异群体Mut_i：

$$Mut_i=Clo_i+\psi_i\times rand(NC_i,n)\times imp_{\text{best}} \tag{8}$$

$$\psi_i=\frac{0.9^{m-i}}{\sum_{j=1}^{m}0.9^j} \tag{9}$$

其中，n是问题维度；m是当前的帝国个数；$rand(NC_i,n)$是一个$NC_i\times n$维的矩阵，其元素是[0,1]之间的随机数；ψ是变异概率，势力越小的帝国主义国家变异的概率越大；imp_{best}是势力最大的帝国主义国家。

③交叉。将变异后的Mut_i随机分为4个一组，每一组按照式(10)进行交叉：

$$Cro_i=Mut_{i,r_1}-Mut_{i,r_2}+Mut_{i,r_3}-Mut_{i,r_4} \tag{10}$$

其中，r_1、r_2、r_3、r_4互不相同。

④选择。从Mut_i和Cro_i中选择势力最大的k个取代当前帝国中最弱的k个帝国主义国家。

另外，帝国合并算子虽然加快了算法的收敛速度，却也使算法容易陷入局部最优。因此，本文提出在算法的每一次迭代中以UR的概率进行帝国合并。适当的UR取值能够平衡收敛速度以及早熟现象之间的矛盾，进一步提高算法的整体性能。

本文将基于克隆进化的改进ICA称为ICACE（ICA based on clone evolution）。

4 算法实验与分析

4.1 常用基准优化函数实验比较

为了测试ICADE和ICACE算法的性能，采用6个常用的基准优化函数进行比较，其中前4个是多模函数，后2个是单模函数，如表1所示。该实验对ICADE和ICACE与OICA、PICA、ICAAI进行比较。ICA的相关参数设置如表2，其他参数的设定同OICA。为便于比较，国家个数与帝国个数的设置同PICA[12]和ICAAI[13]；同化系数β设为2，使殖民地可以从两个方向向帝国主义国家靠近，局部搜索能力较好；帝国竞争概率ρ在OICA中取0.11，本文设为1，即每一次迭代都进行帝国竞争操作；合并概率UR取0.01可以较好地平衡收敛速度及早熟现象之间的矛盾；微分进化

Table 1 Benchmark functions

表1 基准优化函数

函数名	定义	范围	最小值
Ackley	$f(x)=20+\mathrm{e}-20\mathrm{e}^{-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}}-\mathrm{e}^{-\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)}$	$[-32,32]$	$f(0)=0$
Griewank	$f(x)=\sum_{i=1}^{n}\frac{x_i^2}{4\,000}-\prod_{i=1}^{n}\cos(\frac{x_i}{\sqrt{i}})+1$	$[-600,600]$	$f(0)=0$
Rastrigin	$f(x)=10n+\sum_{i=1}^{n}(x_i^2-10\cos(2\pi x_i))$	$[-10,10]$	$f(0)=0$
Schwefel	$f(x)=-\sum_{i=1}^{n}x_i\sin\sqrt{\lvert x_i\rvert}$	$[-500,500]$	$f(420.97)=-418.982\,9n$
Rosenbrock	$f(x)=\sum_{i=1}^{n-1}(100(x_i^2-x_{i+1})^2+(x_i-1)^2)$	$[-10,10]$	$f(1)=0$
Sphere	$f(x)=\sum_{i=1}^{n}x_i^2$	$[-100,100]$	$f(0)=0$

Table 2 Parameters setting of ICA

表2 ICA参数设置

N_{pop}	N_{imp}	β	ρ	UR	MR	F	CR	λ
88	8	2	1	0.01	0.3	0.5	0.9	0.8

中缩放因子 $F=0.5$,交叉概率 $CR=0.9$ [15];克隆系数λ决定了克隆群体的大小,太小起不到克隆的效果,太大会影响收敛速度,本文取0.8,是一个较好的选择。

ICADE和ICACE与OICA、PICA、ICAAI每次运行的初始国家是相同的,分别独立运行40次,每次运行以函数的总计算次数(*FEs*=25 000)为停止准则,记录40次最优结果的平均值(Mean)及标准差(Std),测试函数的维度 n 分别取30、50和100,实验结果如表3~表5所示。

分析表3~表5可知,ICACE在Griewank、Rastrigin和Sphere函数上都能找到全局最优解,在Ackley和Rosenbrock函数上能达到距离全局最优较近的位置,明显优于其他ICA改进算法;ICADE在Schwefel函数上表现出较好的性能,而在其他函数上的优化能力与PICA和ICAAI相差不大;随着优化函数维度的增加,PICA、ICAAI和ICADE的优化结果越来越差,而ICACE则表现出很好的稳定性,仍然具有较强的寻优能力。

Table 3 Performance comparison among 5 ICAs(*FEs*=25 000, n=30)

表3 5种ICA改进算法性能对比(*FEs*=25 000, n=30)

函数名		OICA	PICA	ICAAI	ICADE	ICACE
Ackley	Mean	4.90E+00	4.96E+00	1.98E+00	3.87E+00	**8.88E−16**
	Std	5.86E+00	6.21E−01	6.19E−01	7.84E−01	0
Griewank	Mean	2.67E−02	2.44E+00	1.08E+00	1.81E+00	**0**
	Std	3.36E−02	8.54E−01	9.26E−02	4.06E−01	0
Rastrigin	Mean	1.92E+02	1.08E+02	6.65E+01	6.68E+01	**0**
	Std	4.23E+01	2.16E+01	1.17E+01	1.74E+01	0
Schwefel	Mean	−8.53E+03	−1.13E+04	−1.12E+04	**−1.14E+04**	−1.11E+04
	Std	6.46E+02	2.75E+02	2.69E+02	2.63E+02	3.46E+02
Rosenbrock	Mean	7.28E+01	5.67E+02	1.48E+02	4.30E+02	**2.73E+01**
	Std	3.91E+01	3.34E+02	8.76E+01	1.88E+02	5.12E−01
Sphere	Mean	2.25E−05	2.12E+02	6.47E+00	9.02E+01	**0**
	Std	2.12E−05	8.83E+01	4.11E+00	3.47E+01	0

Table 4 Performance comparison among 5 ICAs(*FEs*=25 000, *n*=50)

表4 5种ICA改进算法性能对比(*FEs*=25 000, *n*=50)

函数名		OICA	PICA	ICAAI	ICADE	ICACE
Ackley	Mean	1.60E+01	1.03E+01	5.33E+00	9.86E+00	**8.88E−16**
	Std	5.40E+00	9.80E−01	7.92E−01	1.02E+00	0
Griewank	Mean	9.32E−02	3.44E+01	3.79E+00	2.98E+01	**0**
	Std	5.86E−02	9.66E+00	1.30E+00	6.48E+00	0
Rastrigin	Mean	5.75E+02	3.61E+02	1.95E+02	2.86E+02	**0**
	Std	6.48E+01	5.07E+01	1.99E+01	4.32E+01	0
Schwefel	Mean	−1.34E+04	−1.70E+04	−1.69E+04	**−1.76E+04**	−1.49E+04
	Std	8.33E+02	5.26E+02	5.06E+02	5.45E+02	7.95E+02
Rosenbrock	Mean	2.88E+02	2.72E+04	7.61E+02	2.51E+04	**4.89E+01**
	Std	1.91E+02	1.21E+04	2.61E+02	1.16E+04	5.27E−02
Sphere	Mean	7.34E−02	3.90E+03	2.50E+02	3.51E+03	**0**
	Std	3.98E−02	1.07E+03	1.07E+02	9.21E+02	0

Table 5 Performance comparison among 5 ICAs(*FEs*=25 000, *n*=100)

表5 5种ICA改进算法性能对比(*FEs*=25 000, *n*=100)

函数名		OICA	PICA	ICAAI	ICADE	ICACE
Ackley	Mean	1.92E+01	1.90E+01	1.04E+01	1.70E+01	**8.88E−16**
	Std	1.85E−01	7.55E−01	8.28E−01	6.65E−01	0
Griewank	Mean	1.85E+00	4.97E+02	5.35E+01	4.10E+02	**0**
	Std	2.78E−01	5.88E+01	1.47E+01	6.46E+01	0
Rastrigin	Mean	1.98E+03	1.54E+03	7.23E+02	1.31E+03	**0**
	Std	2.24E+02	8.06E+01	5.07E+01	1.20E+02	0
Schwefel	Mean	−2.36E+04	−2.75E+04	−2.72E+04	**−2.84E+04**	−2.26E+04
	Std	9.88E+02	9.19E+02	8.83E+02	1.32E+03	1.64E+03
Rosenbrock	Mean	1.53E+03	1.03E+06	2.08E+04	7.44E+05	**9.89E+01**
	Std	4.02E+02	1.83E+05	8.14E+03	2.01E+05	6.57E−02
Sphere	Mean	1.03E+02	5.23E+04	6.13E+03	4.55E+04	**0**
	Std	3.89E+01	7.98E+03	2.18E+03	6.21E+03	0

下面比较ICADE、ICACE与OICA、PICA、ICAAI在基准函数上的收敛性,收敛曲线如图1所示。图中横坐标代表测试函数适应值的计算次数*FEs*,范围为0~25 000,纵坐标表示对应*FEs*的函数最优值。

从收敛曲线可以看出,ICACE的收敛速度明显快于ICA的其他改进算法。除了Schwefel函数,ICACE在*FEs*为2 000时基本上能够达到全局最优位置。

综上所述,对于大多数高维多模函数而言,ICACE基本上能够快速找到全局最优,整体性能明显优于其他几种ICA改进算法。

4.2 常用标准复合优化函数实验比较

本节采用文献[16]中的6个标准复合测试函数CF1~CF6进行实验比较,此类函数有大量的局部极值点,很难找到全局最优值,已成为国际上通用的标准测试函数。本节各ICA改进算法的参数设置同4.1节,在相同条件下分别独立运行20次,每次运行以函数的总计算次数(*FEs*=50 000)为停止准则,实验结果如表6所示。可以看出,ICADE和ICACE在大部分复

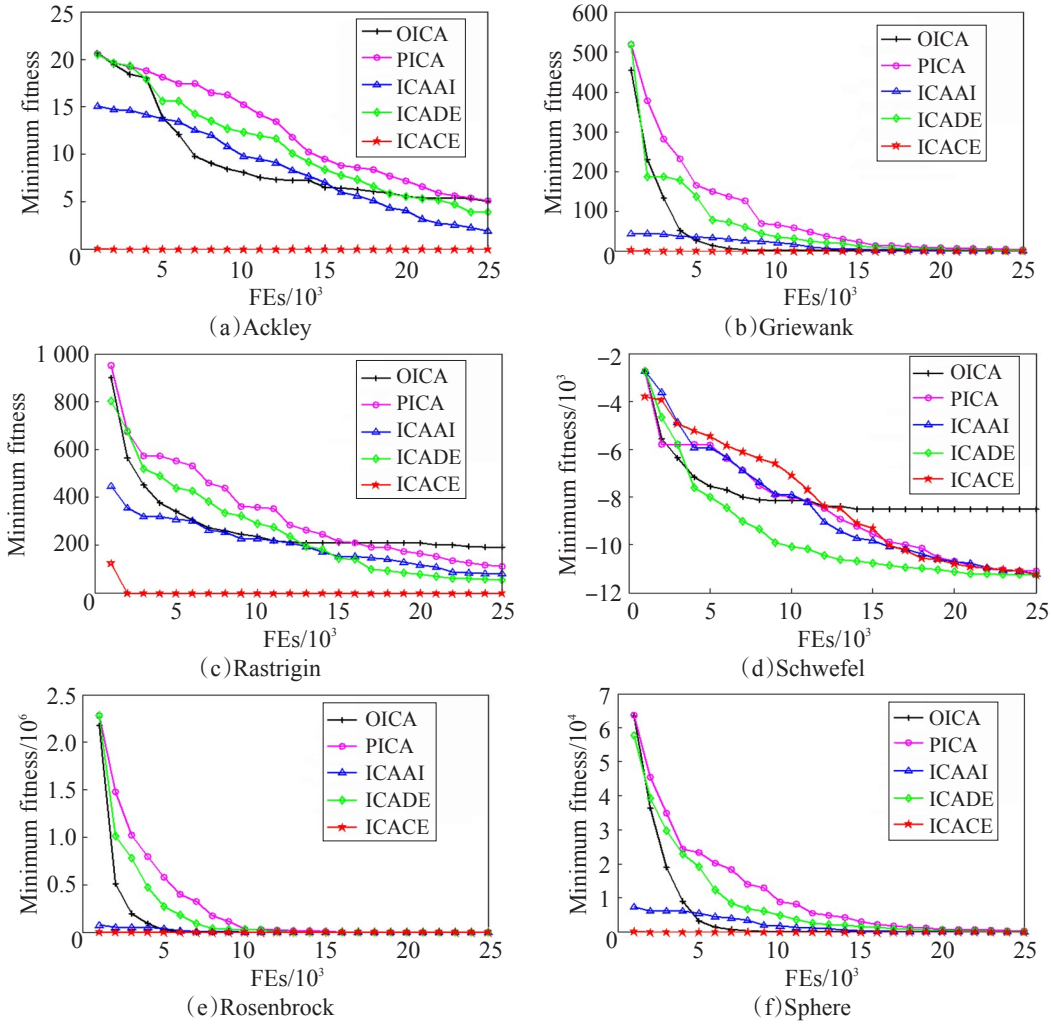

Fig.1 Convergence curves of 5 ICAs on the benchmark functions

图1 5种ICA改进算法在基准函数上的收敛曲线

Table 6 Performance comparison among 5 ICAs on the composition test functions

表6 5种ICA改进算法在复合测试函数上的性能对比

函数		OICA	PICA	ICAAI	ICADE	ICACE
CF1	Mean	1.00E+01	8.27E−19	3.89E−18	**7.56E−19**	3.09E−17
	Std	3.16E+01	9.59E−19	5.49E−18	1.90E−18	6.63E−17
CF2	Mean	3.42E+01	1.55E+01	1.52E+01	2.66E+01	**1.47E+01**
	Std	3.25E+01	8.76E+00	7.40E+00	3.39E+01	6.78E+00
CF3	Mean	2.20E+02	1.55E+02	1.46E+02	**1.33E+02**	1.65E+02
	Std	5.33E+01	5.04E+01	3.81E+01	3.10E+01	4.40E+01
CF4	Mean	3.83E+02	3.17E+02	3.36E+02	**3.05E+02**	3.60E+02
	Std	5.95E+01	1.48E+01	1.01E+02	1.48E+01	2.88E+01
CF5	Mean	2.49E+01	8.72E+00	9.41E+00	1.01E+01	**7.86E+00**
	Std	9.44E+00	2.95E+00	3.44E+00	4.68E+00	3.16E+00
CF6	Mean	5.04E+02	**4.63E+02**	5.02E+02	7.42E+02	5.02E+02
	Std	5.98E+00	5.00E+01	1.60E+00	2.08E+02	1.06E+00

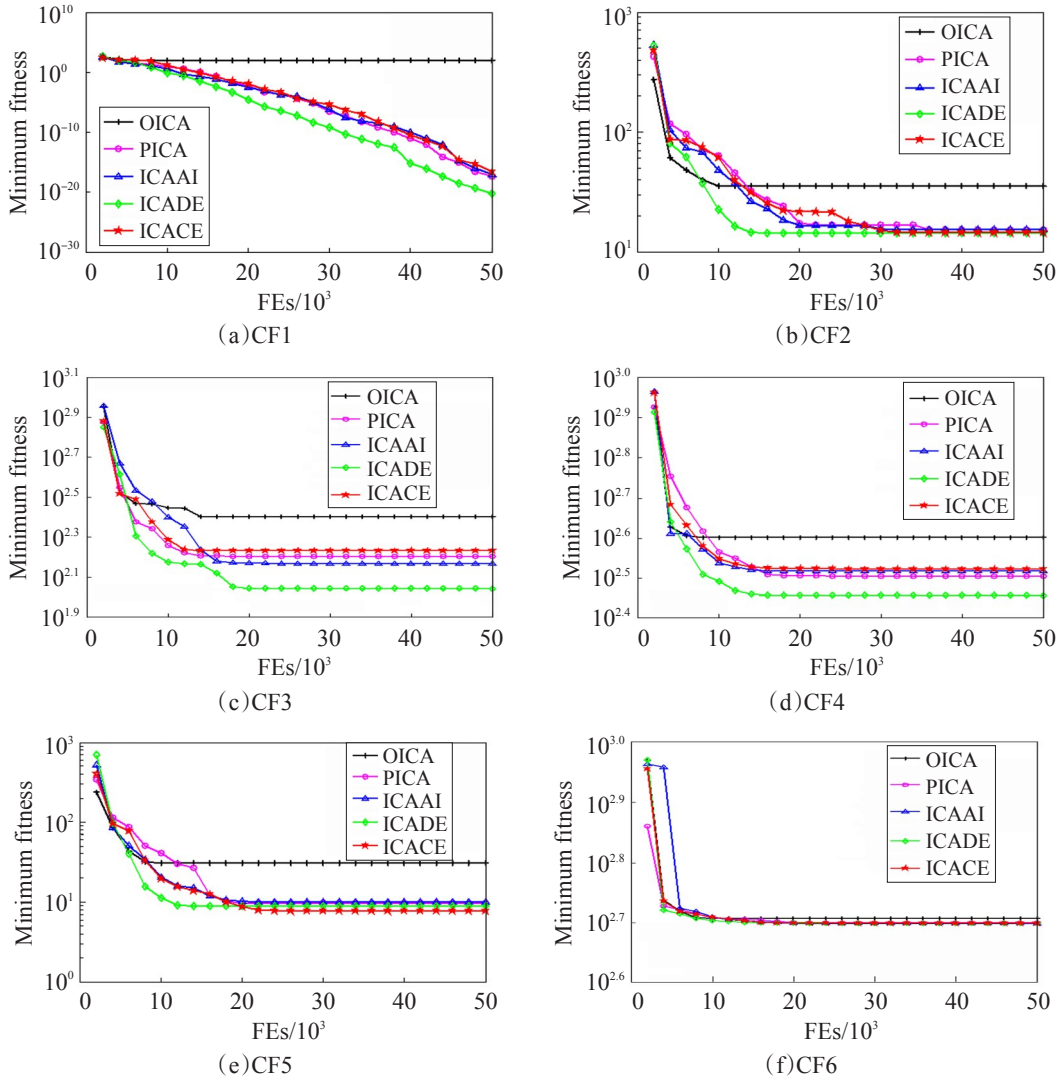

Fig.2 Convergence curves of 5 ICAs on the composition test functions

图2 5种ICA改进算法在复合函数上的收敛曲线

合函数上的求解精度优于OICA，比PICA和ICAAI相对好一点，但不明显。究其原因，ICACE具有较快的收敛速度，而复合函数大量的局部极值点使算法因快速收敛而出现早熟现象，这有待于进一步研究。

下面比较ICADE、ICACE与OICA、PICA、ICAAI在复合函数上的收敛性，收敛曲线如图2所示。图中纵坐标表示对应*FEs*的函数最优值的对数。

从图2中可以看出，ICADE在CF1、CF3和CF4上的求解结果明显好于其他算法，ICACE在CF2和CF5上的寻优能力相对较好。

文献[17]智能单粒子优化算法（ISPO）中列出了PSO的4种改进算法在CF1~CF6上的测试结果，其*FEs*同样设置为50 000，将ICADE和ICACE的实验结果与其进行比较，如表7所示。可以看出，本文算法相比于PSO的4种改进算法有明显的优势。

5 结束语

本文在ICA群智能优化算法框架上，借鉴微分进化和免疫克隆思想，引入了微分进化算子和克隆进化算子，提出了基于生物进化的改进ICA算法。实验结果表明，本文的改进ICA算法，不仅保证了算法的求解效率，而且在求解高维优化问题上有较明显的

Table 7 Comparison among ICADE, ICACE and 4 optimization PSO algorithms

表7 ICADE、ICACE与PSO的4种改进算法的比较

函数		PSO-*w*	CPSO	CLPSO	ISPO	ICADE	ICACE
CF1	Mean	2.32E+02	1.39E+02	1.41E+02	2.47E+01	**7.56E−19**	3.09E−17
	Std	1.17E+02	2.79E+01	2.13E+02	7.79E+01	1.90E−18	6.63E−17
CF2	Mean	2.29E+02	1.15E+02	1.12E+02	6.54E+01	2.66E+01	**1.47E+01**
	Std	1.61E+02	3.48E+01	1.40E+02	6.03E+01	3.39E+01	6.78E+00
CF3	Mean	3.42E+02	2.27E+02	3.01E+02	2.12E+02	**1.33E+02**	1.65E+02
	Std	1.52E+02	6.45E+01	1.95E+02	6.06E+01	3.10E+01	4.40E+01
CF4	Mean	4.71E+02	3.77E+02	3.32E+02	4.11E+02	**3.05E+02**	3.60E+02
	Std	1.51E+02	4.79E+01	1.26E+02	5.87E+01	1.48E+01	2.88E+01
CF5	Mean	2.69E+02	1.11E+02	1.95E+02	3.37E+01	1.01E+01	**7.86E+00**
	Std	2.41E+02	2.47E+01	2.36E+02	2.88E+01	4.68E+00	3.16E+00
CF6	Mean	9.05E+02	6.71E+02	8.39E+02	6.95E+02	7.42E+02	**5.02E+02**
	Std	3.10E+00	1.58E+02	1.44E+02	2.07E+02	2.08E+02	1.06E+00

优势,对于解决现实中复杂的优化问题有一定的意义。下一步,将在本文的基础上,考虑ICA各参数的自适应性,以进一步提高算法的收敛精度以及收敛速度。

References:

[1] Atashpaz-Gargari E, Lucas C. Imperialist competitive algorithm: an algorithm for optimization inspired by imperialistic competition[C]//Proceedings of the 2007 IEEE Congress on Evolutionary Computation (CEC '07), Singapore, 2007. Piscataway, NJ, USA: IEEE, 2007: 4661-4667.

[2] Behnamian J, Zandieh M. A discrete colonial competitive algorithm for hybrid flowshop scheduling to minimize earliness and quadratic tardiness penalties[J]. Expert Systems with Application, 2011, 38(12): 14490-14498.

[3] Forouharfard S, Zandieh M. An imperialist competitive algorithm to schedule of receiving and shipping trucks in cross-docking systems[J]. The International Journal of Advanced Manufacturing Technology, 2010, 51(9/12): 1179-1193.

[4] Karimi N, Zandieh M, Najafi A A. Group scheduling in flexible flow shops: a hybridised approach of imperialist competitive algorithm and electromagnetic-like mechanism[J]. International Journal of Production Research, 2011, 49(16): 4965-4977.

[5] Shokrollahpour E, Zandieh M, Dorri B. A novel imperialist competitive algorithm for bi-criteria scheduling of the assembly flowshop problem[J]. International Journal of Production Research, 2011, 49(11): 3087-3103.

[6] MousaviRad S J, Akhlaghian Tab F, Mollazade K. Application of imperialist competitive algorithm for feature selection: a case study on bulk rice classification[J]. International Journal of Computer Applications, 2012, 40(16): 41-48.

[7] Karami S, Shokouhi S B. Application of imperialist competitive algorithm for automated classification of remote sensing images[J]. International Journal of Computer Theory and Engineering, 2012, 4(2): 137-143.

[8] Coelho L D S, Afonso L D, Alotto P. A modified imperialist competitive algorithm for optimization in electromagnetic[J]. IEEE Transactions on Magnetics, 2012, 48(2): 579-582.

[9] Lucas C, Nasiri-Gheidari Z, Tootoonchian F. Application of an imperialist competitive algorithm to the design of a linear induction motor[J]. Energy Conversion and Management, 2010, 51(7): 1407-1411.

[10] Bahrami H, Faez K, Abdechiri M. Imperialist competitive algorithm using chaos theory for optimization (CICA)[C]//Proceedings of the 2010 12th International Conference on Computer Modelling and Simulation (UKSim '10), Cambridge, UK, 2010. Wanshington, DC, USA: IEEE Computer Society, 2010: 98-103.

[11] Zhang yang, Wang Yong, Peng Cheng. Improved imperialist competitive algorithm for constrained optimization[C]//Pro-

ceedings of the International Forum on Computer Science-Technology and Applications (IFCSTA '09), Chongqing, China, 2009. Wanshington, DC, USA: IEEE Computer Society, 2009: 204-207.

[12] Lin J-L, Cho C-W, Chuan H-C. Imperialist competitive algorithms with perturbed moves for global optimization[J]. Applied Mechanics and Materials, 2013, 284/287: 3135-3139.

[13] Lin J-L, Tsai Y-H, Yu C-Y, et al. Interaction enhanced imperialist competitive algorithms[J]. Algorithms, 2012, 5(4): 433-448.

[14] Atashpaz-Gargari E. Imperialist competitive algorithm (ICA) [CP/OL]. (2008-11-10)[2012-01-10]. http://www.mathworks.com/ matlabcentral/fileexchange/22046-imperialist-competitive-algorithm-ica.

[15] Zhou Yanping, Gu Xingsheng. Development of differential evolution algorithm[J]. Control and Instruments in Chemical Industry, 2007, 34(3): 1-5.

[16] Liang J J, Suganthan P N, Deb K. Novel composition test functions for numerical global optimization[C]//Proceedings of the IEEE International Swarm Intelligence Symposium (SIS '05), Messina, Italy, 2005. Piscataway, NJ, USA: IEEE, 2005: 68-75.

[17] Ji Zhen, Zhou Jiarui, Liao Huilian, et al. A novel intelligent single particle optimizer[J]. Chinese Journal of Computers, 2010, 33(3): 556-561.

附中文参考文献:

[15] 周艳平, 顾幸生. 差分进化算法研究进展[J]. 化工自动化及仪表, 2007, 34(3): 1-5.

[17] 纪震, 周家锐, 廖惠连, 等. 智能单粒子优化算法[J]. 计算机学报, 2010, 33(3): 556-561.

GUO Wanqing was born in 1989. She is a master candidate at Fuzhou University. Her research interest is computational intelligence.

郭婉青(1989—),女,福建泉州人,福州大学硕士研究生,主要研究领域为计算智能。

YE Dongyi was born in 1964. He received the Ph.D. degree in applied mathematics from University of Toulouse (France) in 1992. Now he is a professor and Ph.D. supervisor at College of Mathematics and Computer Science, Fuzhou University. His research interests include computational intelligence, rough set theory and optimization method, etc. He has published more than 90 papers in domestic and international journals and conferences.

叶东毅(1964—),男,福建泉州人,1992 年于法国图卢兹大学应用数学专业获得博士学位,现为福州大学数学与计算机科学学院教授、博士生导师,主要研究领域为计算智能,粗糙集理论,最优化方法等。发表学术论文90余篇。