Journal of Computer Applications
计算机应用,2013,33(S2):86-90
ISSN 1001-9081
CODEN JYIIDU
2013-12-31
http://www.joca.cn

文章编号:1001-9081(2013)S2-0086-05

基于帝国分裂的帝国竞争算法优化

郭婉青*,叶东毅
(福州大学 数学与计算机科学学院,福州 350108)
(* 通信作者电子邮箱 252698387@qq.com)

摘　要: 帝国竞争算法(ICA)是一种受帝国竞争行为启发的新的群智能优化算法。在ICA的迭代过程中,帝国个数不断减少,导致群体多样性降低,这对于高维多模优化问题的求解是不利的,算法容易陷入局部最优。为了克服这个缺陷,引入一种帝国分裂机制,同时增加扰动策略,使算法性能显著提高,在求解高维优化问题上取得明显的改进效果。对多个标准测试函数进行了实验,结果验证了该算法的优良特性,表明适当的分裂策略和扰动策略对于提高ICA的性能是有效的。

关键词: 优化算法; 帝国竞争算法; 帝国分裂; 扰动策略
中图分类号: TP18　　**文献标志码:** A

Optimization of imperialist competitive algorithm based on empire splitting

GUO Wanqing*, YE Dongyi
(*College of Mathematics and Computer Science, Fuzhou University, Fuzhou Fujiang* 350108, *China*)

Abstract: Imperialist Competitive Algorithm (ICA) is a new algorithm for optimization which is inspired by imperialistic competition. With the number of empires reducing progressively in the iteration process, ICA's population diversity declines. This phenomenon makes ICA easily get stuck into a local optimum when it is applied to high-dimensional multimodal optimization problems. To overcome this shortcoming, an empire splitting strategy and a perturbation strategy were proposed in this paper. The proposed algorithm obtains significant improvements when dealing with high-dimensional optimization problems. Finally, the experimental results on several benchmark functions validate the good property of the proposed algorithm. It indicates that proper splitting strategy and perturbation strategy are effective for improving the performance of ICA.

Key words: optimization algorithm; Imperialist Competitive Algorithm (ICA); empire splitting; perturbation strategy

0　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已被成功应用于解决实际的优化问题,如调度问题[2-5]、分类问题[6-7]以及机械设计[8-9]等。与其他进化算法一样,ICA也是基于群体的优化方法。其个体称为国家,ICA将个体分为两类:殖民地和帝国主义国家,一个帝国主义国家及其管辖的殖民地组成一个帝国。ICA的本质就是帝国之内的同化机制以及帝国之间的竞争机制。ICA的最大优点是收敛速度快,但其同样存在早熟现象,为此,一些改进的算法被提出。例如,Bahrami等[10]利用混沌映射决定同化机制中殖民地的移动方向;Zhang等[11]在殖民地移动之后,随机选择一部分殖民地进行位置更新;Lin等[12]提出了扰动ICA算法(PICA),使殖民地可以朝远离帝国主义国家的方向移动,提高了ICA的局部搜索能力。在文献[13]中,ICA算法的提出者Atashpaz-Gargari等给出了改进的算法源码,本文将此改进算法记为OICA。

以上这些改进的算法都不可避免地遇到同一个问题,即帝国个数不断的减少,导致群体多样性降低,这对于高维多模优化问题的求解是不利的,算法容易陷入局部最优。针对这个缺陷,本文引入一种帝国分裂机制,当一个帝国内有两个势力相当的国家时进行分裂,同时,为了尽早跳出局部极值,增加了扰动策略。通过一系列优化函数的测试,实验结果表明,本文提出的改进策略使算法性能显著提高,在求解高维优化问题上取得明显的改进效果。

1　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。

1.1　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:

$$country = [p_1, p_2, p_3, \cdots, p_N]$$

国家的势力大小通过代价函数来衡量:

$$cost = f(country) = f(p_1, p_2, p_3, \cdots, p_N)$$

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:

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

收稿日期:2013-03-15;修回日期:2013-05-07。　基金项目:福建省自然科学基金资助项目(2012J01262)。
作者简介:郭婉青(1989-),女(回族),福建泉州人,硕士研究生,主要研究方向:计算智能;　叶东毅(1964-),男,福建泉州人,教授,博士生导师,博士,主要研究方向:计算智能、数据挖掘。

$$p_n = \left| C_n / \sum_{i=1}^{N_{imp}} C_i \right| \quad (2)$$

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

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

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

1.2　同化机制

帝国主义国家为了更好地控制其殖民地, 将自己的思想模式及文化风俗推广到殖民地的过程, 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, 即殖民地的势力更大, 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。

1.3　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, 由于帝国主义国家对整个帝国的势力影响较大, 而殖民地的影响非常小, 因此, ICA 采用式(4)计算一个帝国的总代价:

$$T.C._n = f(imp_n) + \xi\left(\sum_{i=1}^{N.C.n} f(col_i)\right) / N.C.n \quad (4)$$

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。

最后, 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。

1.4　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地减少, 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, 理想情况下, 该帝国的帝国主义国家及其所有殖民地都应处于同一个位置, 此时, 算法终止。

2　基于帝国分裂的改进 ICA

ICA 的关键在于帝国之内的同化机制和帝国之间的竞争机制, 同化机制保证了算法局部搜索的能力, 竞争机制增加了群体的多样性。ICA 的早期研究工作大多集中在同化机制的改进, 以提高算法局部搜索能力, 并没有考虑到帝国个数不断减少会导致群体多样性降低, 当求解高维优化问题时, 算法容易陷入局部最优。另外, 现实社会历史中, 当一个帝国内除了最强的帝国主义国家外, 另有一个殖民地的势力与其相当, 这两个国家将不断发生冲突, 导致帝国分裂。美国的独立战争就是一个经典的例子, 18 世纪 80 年代北美人民的艰苦抗战, 最终结束了英国的殖民统治, 实现了国家的独立。本文对 ICA 容易陷入局部最优的不足, 引入帝国分裂机制, 正是受上述社会历史现实的启发。

一个帝国进行分裂需要满足下面两个条件:

1) 帝国内有两个势力相当的国家, 即两个国家的代价函数值相差较小;

2) 两个国家之间的距离不能太近。

设置第二个条件的原因是, 当两个帝国距离很近时, 殖民地移动的范围大致相同, 此时分裂没有意义。

本文提出的基于帝国分裂的 ICA 是在 OICA[13] 的基础上进行的改进, 算法步骤如下:

1) 根据式(1) ~(3) 初始化帝国;

2) 当不满足算法终止条件时, 循环进行 3) ~11);

3) 对于每个帝国 i, 循环进行 4) ~ 7);

4) 对帝国 i 内的所有殖民地进行同化操作;

5) 选择部分殖民地进行改革;

6) 计算帝国 i 内所有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) 计算每个帝国的总代价;

9) 相似帝国的合并操作;

10) 帝国之间的竞争操作;

11) 帝国符合分裂条件时, 进行帝国分裂;

12) 算法终止。

上述算法步骤中第 5) 步的殖民地改革即重新更新殖民地位置, 目的是增加群体的多样性, 提高全局搜索能力, 可以有不同的选择策略, 本文选择较弱的殖民地进行改革。

下面讨论第 11) 步帝国分裂的具体做法。

第 1 步　选择势力最强的帝国, 记为 emp_{best}。

第 2 步　从 emp_{best} 中选择势力最强的殖民地, 记为 col_{best}, 将 emp_{best} 中的帝国主义国家记为 imp_{old}。

第 3 步　计算 col_{best} 和 imp_{old} 之间的代价函数值之差以及距离是否符合设定的阈值, 即满足:

$$\begin{cases} dist(col_{best}, imp_{old}) > t_1 . searchspace \\ | f(col_{best}) - f(imp_{old}) | < t_2 \end{cases}$$

其中: $0 < t_1 < 1$; $searchspace$ 是搜索范围的大小, 根据不同的取值方法可以分为全局和局部两种, 全局采用整个搜索空间的最大距离, 是固定的, 而局部则根据帝国的势力范围动态确定。本文将全局的帝国分裂方法记为 GICA, 局部的方法记为 LICA, 后面实验中将对这两种方法进行比较。

第 4 步　符合上述条件的帝国进行分裂, 即将 col_{best} 作为新帝国的帝国主义国家, 记为 imp_{new}, 根据 imp_{new} 和 imp_{old} 的势力大小随机分配剩下的殖民地。

鉴于 ICA 陷入局部极小时, 殖民地与帝国主义国家几乎处于同一位置, 故本文提出在 OICA 上增加扰动因子的策略: 如果迄今为止搜索到的全局最优值连续 u 步迭代没有更新, 则重置殖民地位置。当算法陷入局部极小时, 扰动策略可以强迫跳出局部极小点, 从而引发新一轮的搜索。在 GICA 和 LICA 中引入扰动策略, 可以进一步提高算法性能。

3　算法实验与分析

对于群智能优化算法来说, 算法性能的比较依赖具体的问题, 而更一般的是采用标准测试函数来代替不同类型的实际问题。本文也是通过几个经典的标准测试函数, 采用准确性及收敛性这两个评判指标来度量改进算法的性能。

3.1　测试函数

为了测试算法针对不同类型函数的性能, 本文采用 5 个多模函数(表 1) 和 4 个单模函数(表 2) 作为测试的标准函数。

表1 测试函数(多模)

函数名	定义	范围	最小值
Ackley	$f(x)=20+e-20\exp\left(-0.2\sqrt{\frac{1}{n}\sum_{i=1}^{n}x_i^2}\right)-\exp\left(\frac{1}{n}\sum_{i=1}^{n}\cos(2\pi x_i)\right)$	[-32,32]	$f(0)=0$
Griewank	$f(x)=\sum_{i=1}^{n}\frac{x_i^2}{4000}-\prod_{i=1}^{n}\cos(x_i/\sqrt{i})+1$	[-600,600]	$f(0)=0$
Michalewicz	$f(x)=-\sum_{i=1}^{n}\sin(x_i)(\sin(ix_i^2/\pi))^{20}$	[0,π]	$f>-n$
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]	$f(420.97)=-418.9829n$

表2 测试函数(单模)

函数名	定义	范围	最小值
Rosenbrock	$f(x)=\sum_{i=1}^{n-1}\left(100(x_i^2-x_{i+1})^2+(x_i-1)^2\right)$	[-10,10]	$f(1)=0$
Sphere	$f(x)=\sum_{i=1}^{n}x_i^2$	[-100,100]	$f(0)=0$
Sum Squares	$f(x)=\sum_{i=1}^{n}ix_i^2$	[-10,10]	$f(0)=0$
Zakharov	$f(x)=\sum_{i=1}^{n}x_i^2+\left(\sum_{i=1}^{n}0.5ix_i\right)^2+\left(0.5ix_i\right)^4$	[-5,10]	$f(0)=0$

3.2 实验一:帝国分裂性能测试

该实验将GICA和LICA与PICA[12]、OICA、PSO进行比较。标准测试函数的维度n取30,迭代次数设为1 500。ICA的相关参数设置如表3,其他参数的设定同OICA[13]。PSO的相关参数设置如表4,根据测试函数的求解难度的差异,本文设置不同的收敛阈值,如表5。

表3 ICA参数设置

参数含义	参数变量名	值	参数含义	参数变量名	值
国家个数	N_{pop}	88	分裂阈值	t_1	0.05
帝国个数	N_{imp}	8	分裂阈值	t_2	0.01
同化系数	β	2	扰动因子	u	20
竞争概率	ρ	1			

表4 PSO参数设置

参数	值	参数	值
粒子个数	88	学习因子c_1	2
惯性权重ω	0.7线性递减至0.3	学习因子c_2	2

表5 测试函数收敛阈值设置

测试函数	最优值	收敛阈值
Ackley	0	1
Griewank	0	0.1
Michalewicz	>-30	-27
Rastrigin	0	10
Schwefel	-12 569.487	-11 500
Rosenbrock	0	50
Sphere	0	1E-05
Sum Squares	0	1E-05
Zakharov	0	1

OICA、PICA、GICA、LICA每次运行的初始国家(PSO的初始粒子)是相同的,分别运行40次的实验结果如表6,其中成功率是指,40次中最优值小于收敛阈值的次数占总次数的百分比。

表6 运行40次的平均最优值(成功率)

函数	PSO	OICA	PICA	GICA	LICA
Ackley	5.134 (0%)	3.827 (0%)	0.600 (80%)	0.517 (90%)	0.560 (85%)
Griewank	1.288 (0%)	0.013 (100%)	0.104 (65%)	0.156 (50%)	0.093 (60%)
Michalewicz	-20.85 (0%)	-19.686 (0%)	-27.224 (65%)	-27.613 (90%)	-27.608 (90%)
Rastrigin	97.24 (0%)	189.313 (0%)	9.921 (45%)	10.002 (55%)	9.822 (50%)
Schwefel	-7.558E+03 (0%)	-8.46E+03 (0%)	-1.19E+04 (95%)	-1.20E+04(100%)	-1.19E+04 (95%)
Rosenbrock	48.099 (80%)	69.241 (40%)	94.303 (20%)	71.905 (30%)	63.809 (40%)
Sphere	3.736 (0%)	3.56E-08 (100%)	2.754 (0%)	0.762 (5%)	1.677 (0%)
Sum Squares	6.043 (0%)	7.31E-09 (100%)	0.268 (0%)	0.379 (10%)	0.179 (5%)
Zakharov	0.964 (70%)	3.462 (10%)	1.147 (45%)	0.839 (70%)	0.655 (75%)

分析表6可知,基于帝国分裂的改进ICA算法在大多数多模函数上比OICA及PSO的性能更加优越,最优解的精度有显著提高,与最新改进算法PICA相比,性能相对较好,但不明显;而对于单模函数Rosenbrock、Sphere和Sum Squares,性能较差,但是通过参数调整,性能可以显著提高,此部分将在实验二进行测试。

因 GICA 和 LICA 的性能没有太大差别，下面仅比较 GICA、PICA、OICA 和 PSO 在多模函数上的收敛性，收敛曲线如图 1 ~ 5。

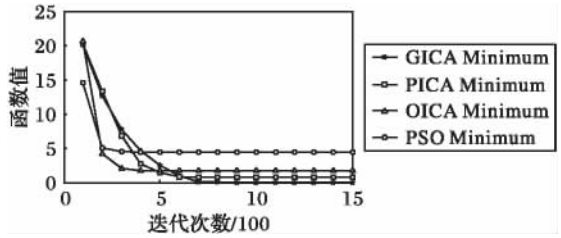

图 1　Ackley 函数上的收敛曲线

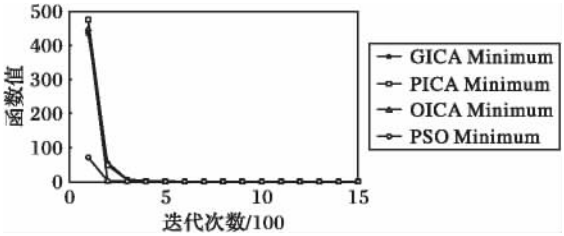

图 2　Griewank 函数上的收敛曲线

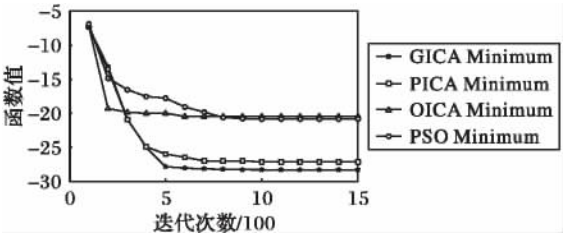

图 3　Michalewicz 函数上的收敛曲线

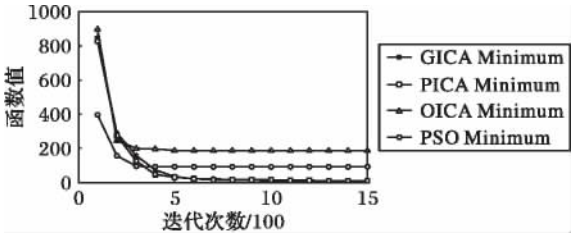

图 4　Rastrigin 函数上的收敛曲线

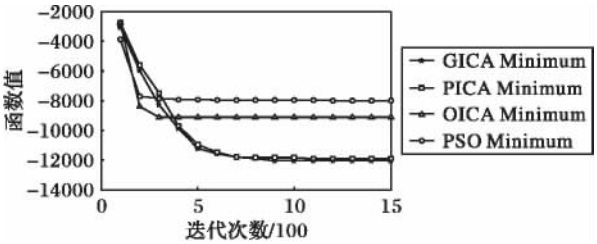

图 5　Schwefel 函数上的收敛曲线

分析图 1 ~ 5，从收敛代数来看，OICA 和 PSO 的收敛速度比较快; 而收敛精度最好的是 GICA，可见 OICA 虽然收敛速度快，但是不容易跳出局部极值，而 GICA 虽然牺牲了一部分时间，却有效提高了算法全局寻优能力。

3.3　实验二: 同化系数 β 的影响

同化系数 β 影响殖民地的移动范围，对算法的局部搜索能力起决定性作用，因此本实验测试同化系数 β 对 OICA 及 GICA 算法的影响，本实验除了 β 取值不同外，其他参数设置同实验一。因 GICA 与 LICA 实验结果相差不大，此处只列举 GICA 的测试结果。表 7、表 8 分别是 OICA 和 GICA 中 β 分别取 2、3、4 的实验结果。

表 7　OICA 运行 40 次的实验结果

函数	平均值(成功率)			标准差		
	β = 2	β = 3	β = 4	β = 2	β = 3	β = 4
Ackley	3.827 (0%)	5.102 (60%)	13.663 (30%)	4.713	7.652	9.277
Griewank	0.013 (100%)	4.557 (85%)	49.724 (40%)	0.017	20.220	46.128
Michalewicz	−19.686 (0%)	−20.935 (0%)	−20.156 (0%)	1.679	1.444	2.060
Rastrigin	189.313 (0%)	120..991 (0%)	115.031 (35%)	63.401	79.175	103.964
Schwefel	−8.46E +03 (0%)	−9.47E +03 (0%)	−7.04E +03 (0%)	580.545	884.693	758.609
Rosenbrock	69.241 (40%)	1622 (60%)	3554 (30%)	27.877	3626.431	4877.951
Sphere	3.56E −08 (100%)	2500 (75%)	7500 (20%)	5.34E −08	4442.617	8506.963
Sum Squares	7.31E −09 (100%)	295 (20%)	1375 (0%)	6.73E −09	291.051	876.521
Zakharov	3.462 (10%)	488.93 (0%)	948.05 (0%)	8.545	192.032	142.901

表 8　GICA 运行 40 次的实验结果

函数	平均值(成功率)			标准差		
	β = 2	β = 3	β = 4	β = 2	β = 3	β = 4
Ackley	0.517 (90%)	1.74E −08 (100%)	2.08E −12 (100%)	0.415	7.79E −08	2.56E −12
Griewank	0.156 (50%)	0.019 (100%)	0.027 (100%)	0.154	0.023	0.027
Michalewicz	−27.613 (90%)	−24.784 (0%)	−27.485 (75%)	0.559	1.107	1.007
Rastrigin	10.002 (55%)	42.634 (0%)	4.328 (95%)	3.545	15.310	2.963
Schwefel	−1.20E +04(100%)	−1.13E +04 (35%)	−1.14E +04 (55%)	207.896	410.497	303.727
Rosenbrock	71.905 (30%)	13.536 (90%)	24.708 (85%)	35.143	21.570	24.480
Sphere	0.762 (5%)	7.35E −56 (100%)	9.96E −19 (100%)	1.567	2.18E −55	4.45E −18
Sum Squares	0.379 (10%)	7.55E −56 (100%)	2.76E −24 (100%)	0.563	3.079E −55	4.485E −24
Zakharov	0.839 (70%)	2.83E −03 (100%)	2.05E −09 (100%)	0.967	0.013	6.47E −09

分析表 7、表 8 可知，对于 Rosenbrock、Sphere、Sum Squares、Zakharov 四个单模函数来说，OICA 在 β 取 2 时效果最好，而 β 取 3 或者 4 时，标准差较大，算法稳定性较差; 相反，GICA 在 β 取 3 或者 4 时，正确率及最优解的精度都有显著提高，标准差较小，算法稳定性较好。

对于多模函数来说，OICA 在 Griwank 函数上也表现出随

着β的增加标准差偏大的情况，对于 Ackley、Michalewicz、Rastrigin 以及 Schwefel 函数则在β取 3 或者 4 时效果相对较好，但不明显；而 GICA 在多模函数上没有表现出一般规律性，但总的看来，β取 2 或 4 时，效果较好，最优解质量具有明显的优势。

综上所述，GICA 对于同一个同化系数β具有较好的稳定性，而且可以根据不同类型的优化函数调整同化系数β以提高最优解精度，而 OICA 稳定性较差，调整β值对算法性能的提高作用不明显。

4 结语

本文在 ICA 群智能优化算法框架上，提出一种基于帝国分裂的改进 ICA 算法，通过引入适当的分裂机制及扰动策略，使算法尽快跳出局部极值，提高了算法性能。实验结果表明，基于帝国分裂的改进 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为明显的优势，对于求解现实中复杂的优化问题有一定的意义。

参考文献:

[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. Piscataway: IEEE Press, 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(9/10/11/12): 1179 - 1193.

[4] KARIMI N, ZANDIEH M, NAJAFI AA. 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 electromagnetics[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]// UKSim'10: Proceedings of the 2010 12th International Conference on Computer Modelling and Simulation. Washington, DC: IEEE Computer Society, 2010: 98 - 103.

[11] ZHANG Y, WANG Y, PENG C. Improved imperialist competitive algorithm for constrained optimization[C]// IFCSTA'09: Proceedings of the 2009 International Forum on Computer Science Technology and Applications. Washington, DC: IEEE Computer Society, 2009, 1: 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, 284/285/286/287: 3135 - 3139

[13] ATASHPAZ-GARGARI E. Imperialist competitive algorithm (ICA)[CP/OL]. [2012 - 09 - 18]. http://www.mathworks.com/matlabcentral/fileexchange/22046-imperialist-competitive-algorithm-ica.

(上接第 82 页)

表 1 设定精度下两个网络的收敛情况统计结果

p = 0.02		p = 0.05		p = 0.06		p = 0.08		p = 0.10	
N_s	N_t	N_s	N_t	N_s	N_t	N_s	N_t	N_s	N_t
9	616.2	8	592.4	8	890.3	10	655.4	10	426.9
10	179.7	10	131.7	10	182.2	10	166.2	10	161.4

3 结语

在多层前向小世界网络结构的基础上，将复杂动态网络研究中的和谐统一的混合择优构建思想融入其中，建立了多层前向小世界结构自适应网络模型，该模型在已经确定的权值矩阵的指导下有目的地选择权值连接，有效地保护连接权值上的信息。该模型继承了原始多层前向小世界网络的基本特性，并且能在有限的网络连接下达到最优的网络性能。首先通过在设定精度的情况下对不同概率下的网络收敛效果作比较，发现在p = 0.05 附近时，收敛效果最好。其次将该模型网络与 WS 多层前向小世界网络模型相比较，在相同概率p下，经过择优的多层前向小世界结构自适应网络具有更好的收敛速度。

参考文献:

[1] 刘天舒. BP 神经网络的改进研究及应用[D]. 哈尔滨: 东北农业大学, 2011.

[2] 沈学利, 张红岩, 张纪锁. 改进粒子群算法对 BP 神经网络的优化[J]. 计算机系统应用, 2012, 19(2): 57 - 61.

[3] SIMARD D, NADEAU L, KROGER H. Faster learning in small-world neural networks[J]. Physics Letters A, 2005, 336(1): 8 - 15.

[4] 李小虎, 杜海峰, 张进华, 等. 多层前向小世界神经网络及其函数逼近[J]. 控制理论与应用, 2010, 27(7): 836 - 842.

[5] 方锦清, 汪小帆, 郑志刚. 网络科学的理论模型及其应用课题研究的若干进展[J]. 复杂系统与复杂性科学, 2008, 5(4): 1 - 12.

[6] SOMPOLINSKY H. Neural networks with nonlinear synapses and static noise[J]. Physical Review A, 1988, 34(3): 2571 - 2575.

[7] 杨群生, 余英林. 最大 - 乘积模糊联想记忆的神经网络学习算法[J]. 控制理论与应用, 2000, 17(5): 699 - 702.

[8] 方锦清. 网络科学的理论模型探索及其进展[J]. 科技导报, 2006, 24(12): 67 - 72.

[9] LU J, HE J, CAO J, *et al*. To pology influenees performances in the associative memory neural network[J]. Physies Letters A, 2006, 354(5/6): 335 - 343.