协同进化算法在关联规则挖掘中的应用

楼 巍,刘 捷,严利民
(上海大学 机电工程及自动化学院,上海 200072)

摘 要:文中采用了一种协同进化算法,分别利用改进的遗传算法和粒子群算法对两个种群同时进行迭代,并在种群之间引入一种信息交互机制,使两个种群协同进化。文中最后通过实验对该协同进化算法、传统的遗传算法以及粒子群算法应用于关联规则挖掘时的性能进行比较,证明了该协同进化算法在可接受的时间复杂度前提下,不仅继承了传统遗传算法挖掘关联规则时无须产生规模庞大的候选项集和有效减少扫描数据库次数的优点,更弥补了其容易早熟收敛的缺陷,从而能高效地搜索出数据库中高质量的关联规则,这点在其应用于高维数据集时尤为显著。

关键词:关联规则挖掘;协同进化;遗传算法;粒子群算法
中图分类号:TP274 **文献标识码**:A **文章编号**:1673-629X(2012)11-0013-05

Applied Research on Association Rules Mining with Co-evolution Algorithm

LOU Wei, LIU Jie, YAN Li-min
(School of Mechatronics Engineering and Automation, Shanghai University, Shanghai 200072, China)

Abstract: It adopts a co-evolution algorithm, which utilizes improved genetic algorithm and particle swarm optimization algorithm to iterate two populations simultaneously. Meanwhile, the mechanism of information interaction between these two populations is introduced. Finally, experiments and application have been made to prove that on the premise of acceptable time complexity, not only does the co-evolution algorithm inherit the superiority of traditional genetic algorithm such as reducing the number of scanning the database effectively and generating small-scale candidate item sets, but also avoid the phenomenon of premature through comparing the properties of co-evolution algorithm, traditional genetic algorithm and particle swarm optimization algorithm when used in association rules mining. High quality association rules can be found when adopted the co-evolution algorithm, especially in high-dimension database.

Key words: association rules mining; co-evolution; genetic algorithm (GA); particle swarm optimization (PSO) algorithm

0 引 言

关联规则挖掘是数据挖掘研究中的一个重要分支,它能帮助不同领域的决策者找出大型数据库中数据项集之间的某种潜在关系,而成为众多学者争相研究的一种知识发现问题[1]。Apriori 算法作为经典的频繁项集生成算法,在关联规则挖掘研究中具有里程碑的作用。然而随着研究的不断深入,Apriori 算法的两个重大缺陷逐渐显现出来[2]:

1)算法必须耗费大量的时间处理规模巨大的候选项集。

2)需多次重复扫描数据库,对候选项集进行模式匹配。

可以看出 Apriori 算法虽然在理论上保证了结果的高精度,然而当用于处理海量、高维数据时,计算时间是相当可观的,甚至是不可能在有限年内解决的。另外 Apriori 算法由于需要对全部或部分数据库中的数据进行遍历,对内存容量的需求很大,并且随着数据规模的增加,对内存容量的需求呈指数性增长。这些问题都限制着 Apriori 算法的使用,因此怎样进一步提高关联规则挖掘算法效率是一个值得深入研究的问题。

针对 Apriori 算法存在的缺陷,已经提出了许多改进算法来弥补其不足。如 J. Han 等人提出的 FP-growth 算法[3]、A. Savasere 等人提出的 Partition 算法等[4]。相较于 Apriori 算法,这些算法的性能大幅提升,但当面对海量、高维数据时,利用这些算法挖掘规则有时仍是不现实的[1]。为此文中从三方面着手,采用了一种协同进化算法,该算法在标准遗传算法的基础上,对其进行优化改进,并引入具有个体记忆功能的改进粒子群算法,同时结合协同进化的思想,使两个种群之间能共享历史信息和当前状态,从而实现在高维

收稿日期:2012-03-11;**修回日期**:2012-06-13
基金项目:上海市学校德育创新发展专项课题(1028)
作者简介:楼 巍(1963-),男,副教授,从事数据挖掘研究;刘 捷(1987-),男,硕士研究生,研究方向为关联规则数据挖掘。

数据集中搜索出高质量的关联规则。

1 协同进化思想

协同进化的概念最早由 Ehrlich 和 Raven 讨论植物和植食昆虫相互之间的进化影响时提出的。它的核心思想是:相互作用的种群互为不可缺少的生存条件并在长期的进化过程中相互依赖、相互协调,从而提高各自和全局的性能[5]。协同进化思想是文中提出算法的核心内容,通过引入该思想,使种群的进化不再仅与自身相关,而且还受与之相互联系的其他种群影响。目前关于协同进化思想的研究内容非常广泛,包括已经提出的多粒子群协同进化算法、协同进化遗传算法等都应用了这一思想[6]。而文中采用"遗传-粒子群"协同进化,即分别利用改进的遗传算法和粒子群算法的各自特点取长补短,对两个种群同时进行迭代,同时结合协同进化思想,使两个种群协同进化,从而实现从在高维数据集中搜索出高质量的关联规则。为了实现这一思想,需要另外设计一种信息交互机制,即协同操作方法,使信息能够在两个种群之间传递,从而达到协同进化的目的。

2 协同进化算法中的遗传搜索策略

2.1 遗传算法概述

遗传算法是借鉴生物界自然遗传机制和进化过程而形成的一种自适应全局优化概率搜索算法,是由美国密西根大学的 Holland 教授及其学生于 1975 年首先提出的[7]。将遗传算法用于关联规则挖掘时由于无须产生规模庞大的候选项集和多次扫描数据库,为在海量高维数据集中进行规则挖掘提供一种可行的思路。但同时早熟收敛和后期收敛速度慢仍是其不可忽视的现象,所以要想在海量高维数据集中挖掘到价值较高的关联规则,必须对遗传算法进行改进。

2.2 遗传搜索策略

根据遗传算法的定义,染色体编码、个体适应度函数和遗传算子的设计是实现遗传算法的核心内容,下面分别介绍它们在文中的具体应用情况。

●编码规则。

关联规则挖掘对应的解空间是整个事务数据库,因此文中采用实数数组的方法进行编码[8]。实数数组的元素个数与事务数据库中的字段个数相对应,元素值代表了字段的属性值。

●适应度函数设计。

支持度是对关联规则有用程度的衡量,它说明了这条规则在整个数据库中出现的频率;置信度是对关联规则的可靠程度的度量,它反映了所发现规则的确定性。因此文中利用关联规则的支持度和置信度来设计适应度函数,如式 1 所示。

$$F(R_j)=\omega_s\frac{\text{Support}(R_j)}{\text{minsupp}}+\omega_c\frac{\text{Confidence}(R_j)}{\text{minconf}} \tag{1}$$

上式中 Support(R_j)、Confidence(R_j)分别为经过遗传操作所形成的一条新规则的支持度和置信度,minsupp、minconf 分别为用户给定的最小支持度阈值和最小置信度阈值,$\omega_s+\omega_c=1$,$\omega_c\geqslant 0$,$\omega_s\geqslant 0$。且规定当满足 $\frac{\text{Support}(R_j)}{\text{minsupp}}\geqslant 1$,$\frac{\text{Confidence}(R_j)}{\text{minconf}}\geqslant 1$ 时,R_j 为符合要求的规则;否则这条规则将在下一代中被淘汰。

●遗传算子的确定。

1)选择算子。

文中采用赌轮选择方法[7]与最佳个体保存方法相结合的选择算子,即对于每一代种群,首先找出一个最优个体直接进入下一代,对其余个体运用赌轮算法,把选择出的个体保存到交配池中。

2)交叉算子设计。

交叉算子是遗传算法中产生新个体的主要手段,它体现了信息交换的思想,即将交配池中的各个染色体随机搭配成对,以交叉概率 P_c 交换它们之间的部分染色体信息。文中选用单点交叉法[7]。

交叉概率是交叉算子设计中另一项重要内容,可以根据进化不同阶段的适应度值动态调整交叉概率的值,文中采用的交叉概率调整策略如公式 2 所示,适应度高的个体其交叉概率应该较小,适应度低的个体其交叉概率应该较大。

$$P_c=\begin{cases}\dfrac{P_{c\max}-P_{c\min}}{1+\exp\left(\dfrac{2(f'-\bar{f})}{f_{\max}-\bar{f}}\right)}+P_{c\min}, & f'\geqslant\bar{f}\\ P_{c\max}, & f'<\bar{f}\end{cases} \tag{2}$$

式中 $P_{c\max}$ 和 $P_{c\min}$ 分别表示交叉概率 P_c 的上下限,文中取值分别为 0.9 和 0.3,$f_{\max}$ 表示当前种群中个体的最大适应度值,$\bar{f}$ 表示当前种群的平均适应度值,f' 是两个交叉个体的较大适应度值。

3)变异算子设计。

文中采用的变异算子是基于均匀变异的改进方法[7],即以变异概率 P_m 选择变异个体,选中后将其每一位都依次进行变异,并保证变异后的每一位都在其允许的取值范围内取值[9]。

变异概率是变异算子设计中另一项重要内容,它控制着变异操作被使用的频率。在实际应用中发现:在种群进化初期,个体的差异较大,选择和交叉算子的作用较明显,进化速度较快,变异率可较小;随着进化进行,个体都向高适应度个体靠近,致使种群中个体结

构渐渐单一,如果连续多代未发生进化,则依靠当前的种群可能无法找到最优解,此时可提高变异概率扩大搜索范围;进化末期,种群中的个体都具有好适应度,这时希望变异概率变小,以免破坏优良模式。为此文中采用与迭代次数相关的先增后减的变异概率,其公式如下:

$$P_m = \begin{cases} P_{m\min} + \dfrac{t}{T_{\max}}, & 0 \leqslant \dfrac{t}{T_{\max}} \leqslant (P_{m\max} - P_{m\min}) \\ (2P_{m\max} - P_{m\min}) - \dfrac{t}{T_{\max}}, & (P_{m\max} - P_{m\min}) \leqslant \dfrac{t}{T_{\max}} \leqslant 1 \end{cases} \tag{3}$$

式中,$P_{m\max}$ 和 $P_{m\min}$ 分别是 P_m 的上下限,文中取值分别为 0.1 和 0.001,$T_{\max}$ 为最大迭代次数,t 为当前迭代次数。

3　协同进化算法中的粒子群搜索策略

由于遗传算法的个体不具有记忆功能,每个个体只能反映当前状态,而与历史状态无关,并且它的进化幅度和方向没有摆脱随机性和盲目性,致使算法后期收敛速度慢且容易早熟收敛,挖掘效果都不甚理想,为此文中引入了具有个体记忆功能的粒子群算法。

3.1　粒子群算法概述

粒子群优化(PSO)算法是 1995 年由美国学者 J. Kennedy 和 R. C. Eberhart 受鸟群觅食行为的启发而提出的一种群体智能算法[10]。它的基本思想是在 D 维空间中随机初始化 M 个无体积无质量的粒子,每个粒子代表 D 维搜索空间中的一个可行解,粒子 i 在 t 时刻的位置变量和速度变量分别为 $\boldsymbol{x}_i^t = (x_{i1}^t, x_{i2}^t, \cdots, x_{id}^t)^{\mathrm{T}}$ 和 $\boldsymbol{v}_i^t = (v_{i1}^t, v_{i2}^t, \cdots, v_{id}^t)^{\mathrm{T}}$,粒子的优劣由一个事先设定的适应度函数来评价。每个粒子将在可行解空间中运动,并由速度变量 v 决定其方向和距离。在每一代中粒子将跟踪两个极值,并经过逐代搜索得到最优解。这两个极值分别是:粒子本身迄今为止找到的最优解 $\boldsymbol{P}_i^t = (p_{i1}^t, p_{i2}^t, p_{i3}^t, \cdots, p_{id}^t)^{\mathrm{T}}$ 以及整个群体迄今为止找到的最优解 $\boldsymbol{P}_g^t = (p_{g1}^t, p_{g2}^t, p_{g3}^t, \cdots, p_{gd}^t)^{\mathrm{T}}$。

3.2　粒子群搜索策略

在使用粒子群算法进行关联规则挖掘时,上节遗传搜索策略中所采用的编码规则及适应度函数设计同样适用,此处不再赘述。PSO 算法中粒子更新公式的改进与控制参数选择是本节的重点讨论对象。

1)粒子更新公式。

由于标准 PSO 算法主要针对连续函数进行搜索运算,而文中采用的 PSO 算法将应用于关联规则挖掘领域,其解空间是整个数据库,属离散域,因此需要对原有粒子更新公式进行离散化的改进。文中提出了 sigmoid 函数:$S(v_{id}^t) = 1/(1 + \exp(-v_{id}^t))$,将其作为粒子位置更新的概率。具体的位置更新公式如式 4、式 5 所示:

$$v_{id}^{t+1} = \omega \times v_{id}^t + c_1 \times r_1 \times (p_{id}^t - x_{id}^t) + c_2 \times r_2 \times (p_{gd}^t - x_{id}^t) \tag{4}$$

$$\begin{cases} x_{id}^{t+1} = x_{id}^t + \mathrm{fix}(v_{id}^{t+1}), & \mathrm{rand}(\,) < S_{id}^{t+1} \\ x_{id}^{t+1} = x_{id}^t, & \mathrm{rand}(\,) \geqslant S_{id}^{t+1} \end{cases} \tag{5}$$

式(5)中,函数 fix( )用来对 v_{id}^{t+1} 向上取整,函数 rand( )用来生成[0,1]的随机数。

2)控制参数。

(1)惯性权重 w。

在 PSO 算法的可调参数中,惯性权重是最重要的参数,通过调节 w 值可以平衡算法的全局搜索和局部搜索能力[11]。一般全局优化算法中,都希望算法在进化初期具有好的全局搜索能力以便找到一片包含最优解的领域,而在进化后期希望其具有较强的局部搜索能力在最优解领域内进行局部搜索,因而 w 的值应该是递减的。故文中采用 w 的非线性调节策略,如式 6 所示。

$$w(t) = w_{\min} + (w_{\max} - w_{\min}) \exp(-3 \times (t/T_{\max})^2) \tag{6}$$

式中,$w_{\max}$ 和 $w_{\min}$ 分别表示惯性权重 w 的上下限,文中分别取 0.9 和 0.4,t 是当前迭代的次数,$T_{\max}$ 为最大迭代次数。

(2)学习因子 C_1、C_2。

学习因子代表将每个粒子推向个体最优和全局最优位置的统计加速项的权值,是调整粒子的自身经验和社会经验在运动中所起作用的因子,体现了粒子的信息交流,设置较大或较小的 C_1、C_2 都不利于粒子的搜索。根据 M. Clerc 推导出的结论[9],文中 C_1、C_2 均取 2.5。

(3)粒子速度。

由于速度本身具有累计的因素,如果速度过大会导致新的位置和原来的位置有很大的偏差,增加了群体的混乱性,削弱了局部搜索的质量[12]。为了得到更好的全局搜索能力而不降低局部搜索得到的解的质量,可将其速度设定在一定的范围 $[-v_{\max}, v_{\max}]$ 内。$v_{\max}$ 决定当前位置与最好位置之间的区域精度。若过大,粒子可能会越过最优解,而过小,则粒子可能会无法探测完整个解空间。因此过大或过小都会影响算法的性能,文中取 $v_{\max} = 3$。

4　基于协同进化算法的关联规则挖掘步骤描述

协同进化算法的流程图如图 1 所示,其中 Step3

运用了协同进化思想，定义了适用于关联规则挖掘的协同操作方法。这样做有助于避免传统遗传算法应用于关联规则挖掘时，易于早熟收敛的缺陷。具体的步骤描述如下：

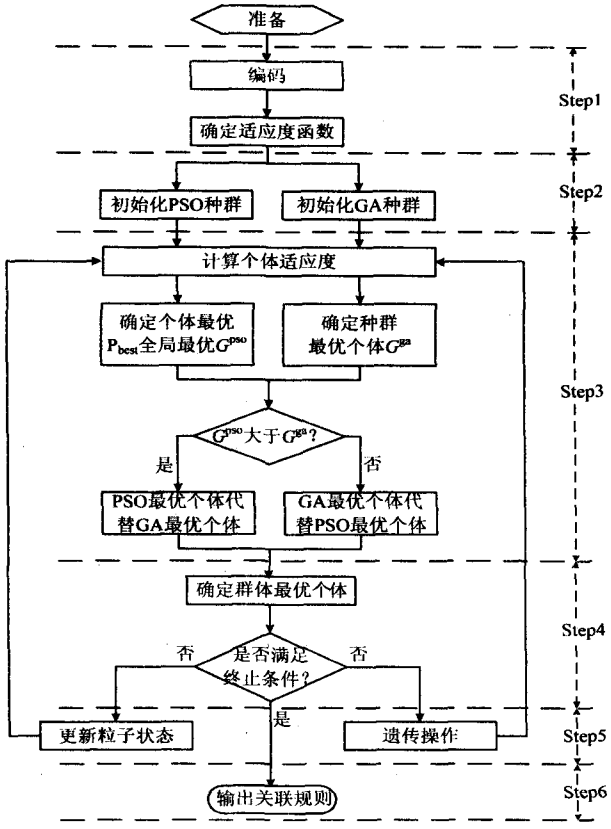

图1 协同进化算法流程图

Step1：根据目标数据库随机产生两个初始化种群，POP_1和POP_2分别采用粒子群搜索策略和遗传搜索策略搜索符合条件的关联规则，两个种群采用相同的编码规则、适应度函数、种群规模和最大进化代数。

Step2：初始化两个种群的各项参数、权重，确定关联规则的最小支持度阈值 minsupp、最小置信度阈值 minconf，以及前后项集约束条件。

Step3：扫描数据库，计算两个种群中所有个体的适应度值，保留符合条件的个体进入各自下一代，淘汰不符合要求的个体。并将POP_1中的全局最优个体G^{pso}与POP_2中的最优个体G^{ga}进行适应度值的比较，用具有较大适应度值的个体替换另一种群的最优个体，作为下一代进化的依据。

Step4：判断此时是否满足终止条件，若迭代次数已经达到最大迭代次数则算法结束，转 Step6；否则继续执行下一步。

Step5：按照公式(4)和(5)分别对POP_1的速度和位置进行更新，产生下一代种群，对POP_2利用遗传操作，得到下一代种群，转到 Step3 继续进行适应度值评价。

Step6：输出关联规则。

当POP_1中的个体陷入局部最优点的时候，个体不再仅根据自身群体的经验去确定下一步位置，同时还会吸取POP_2中最优个体的信息确定下一步的位置。随着POP_2中优秀个体信息的获取，可以引导本已陷入局部最优值的个体偏离原先局部最优点，以较大概率向全局最优点靠近。

5 实验分析与应用

将该协同进化算法在 Windows XP 操作系统下用 MATLAB7.6(R2008a)编程实现，通过跟踪进化过程中种群的平均适应度值和运行时间来比较遗传算法、粒子群算法以及协同进化算法分别应用于关联规则挖掘时的性能优劣。实验数据库来源：①UCI 数据库中的 Connect-4 数据集，该数据集共有 67557 个数据元素，42 个维度。②UCI 数据库中的 Plants 数据集，该数据集共有 22632 个数据，70 个维度。实验环境：CPU Intel 双核 3.0GHz；内存 2GB。实验参数：种群数量 M 为 30，最大迭代次数 T_{max}为 1000，最小支持度 minsupp 为 0.45，最小置信度 minconf 为 0.6，ω_s为 0.6，ω_c为 0.4。此处需要特别说明的是，以上六个实验参数对该协同进化算法的求解结果和求解效率都有一定的影响，但目前尚无合理选择其值的理论依据，在实际应用中，需要经过多次试验后才能确定出这些参数合理的取值大小。

三种算法在 Connect-4 数据集和 Plants 数据集上的进化过程如图 2、图 3 所示。这两个数据集的维度分别为 42 维和 70 维，属高维数据集。由于图 2、图 3 所反映出的信息具有相似性，故以图 2 为例，分析三种算法在高维数据集上运行时的特性。从图 2 中可以看出，在进化初期协同进化算法的个体质量就明显优于遗传算法和粒子群算法。随着迭代次数不断增加，遗传算法在经过 40 次迭代后就已经陷入早熟收敛且无法跳出。粒子群算法相较于遗传算法而言，其个体质量都有所提高，但同样面临无法跳出局部最优解的窘境。而协同进化算法虽然在迭代过程中也同样出现过早熟收敛的现象，但在第 180 次迭代时出现了明显的拐点，表明该算法在此处引导本已陷入局部最优值的个体偏离原先局部最优点，以较大概率向全局最优点靠近。

表1 Co、GA、PSO 应用于 Connect-4 与 Plants 数据集时的运行时间

运行时间(s)	Connect-4 数据集	Plants 数据集
Co	4946.13s	3765.79s
GA	3601.43s	2658.95s
PSO	2153.56s	1769.68s

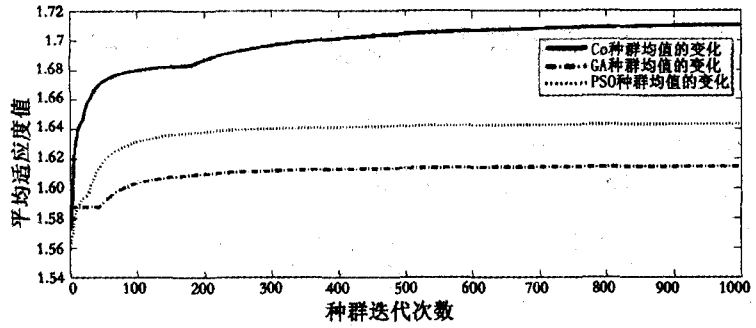

图 2　Co、GA、PSO 应用于 Connect-4 数据集时的对比示意图

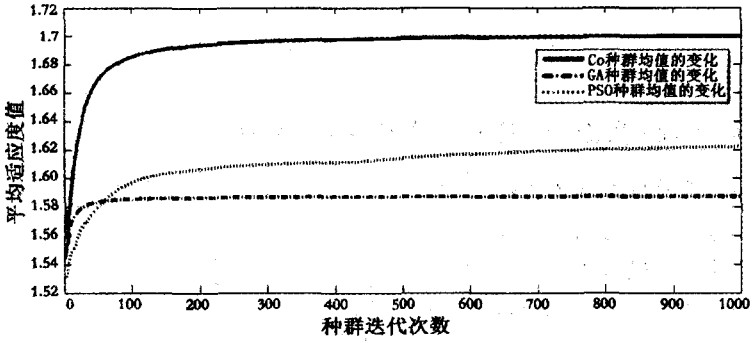

图 3　Co、GA、PSO 应用于 Plants 数据集时的对比示意图

从表 1 可以看出,在高维数据集上,文中介绍的协同进化算法在运行时间上相对于其他两种全局优化算法没有优势,但完全在可以接受的范围之内。

6　结束语

文中介绍了一种协同进化算法,分别利用改进的遗传算法和粒子群算法对两个种群同时进行迭代,并在种群之间引入一种信息交互机制,使两个种群协同进化。通过实验验证、分析,证明了在高维数据集中,尽管协同进化算法的运行时间相较于其他两种全局优化算法稍长,但运行时间完全在可以接受的范围之内,而且由于有效地引入了协同进化思想,相比使用其他两种算法进行关联规则挖掘时,其不仅在挖掘质量上更胜一筹,同时在跳出局部最优解的能力上也优势显著,实现了在高维数据集中搜索出高质量的关联规则。

参考文献:

[1] Han Jiawei, Kamber M. Data Mining: Concepts and Techniques[M]. 2nd ed. Beijing: China Machine Press, 2011: 146-155.

[2] Agrawal R, Imielinski T, Swami A. Mining association rules between sets of items in large databases[C]//Proc of 1993 ACM-SIGMOD Int Conf on Management of Database. Washington, DC: ACM Press, 1993.

[3] Han Jiawei, Pei Jian, Yin Yiwen. Mining frequent patterns without candidate generation [C]//Proceedings of the 2000 ACM-SIGMOD International Conference on Management of Data. Dallas, Texas: ACM Press Publisher, 2000: 1-12.

[4] Savasere A, Omiecinski E, Navathe S. An efficient algorithm for mining association rules in large databases[C]//Proceedings of the 21th International Conference on Very Large Databases (VLDB'95). Zurich, Switzerland: Morgan Kqufmann Publisher, 1995.

[5] Wiegand R P. An analysis of cooperative coevolutionary algorithms[D]. Fairfax: George Mason University, 2003.

[6] 许　珂,刘　栋. 多粒子群协同进化算法[J]. 计算机工程与应用,2009:45(3):51-54.

[7] 雷英杰. MATLAB 遗传算法工具箱及应用[M]. 西安:西安电子科技大学出版社,2004:38-45.

[8] Sharma S K, Irwin G W. Fuzzy coding of genetic algorithms [J]. IEEE Trans on Evolutionary Computation, 2003(7): 344-355.

[9] Thierens D. Adaptive mutation rate control schemes in genetic algorithms[C]//Proceedings of the 2002 Congress on Evolutionary Computation. [s. l.]: [s. n.], 2002: 980-985.

[10] 李　丽. 粒子群优化算法[M]. 北京:冶金工业出版社,2009.

[11] 王　丽,王晓凯. 一种非线性改变惯性权重的粒子群算法[J]. 计算机工程与应用,2007,43(4):47-48.

[12] Shi Y, Eberhart R C. Parameter selection in particle swarm adaptation[M]//Evolutionary Programming VII. Berlin, Germany: Springer-Verlag, 1997: 591-600.

(上接第 12 页)

[9] 邢进良. BP 神经网络及其应用[J]. 沙洋师范高等专科学校学报,2007(5):46-50.

[10] 刘年生. 神经网络混沌加密算法及其在下一代互联网安全通信中的应用研究[D]. 厦门:厦门大学,2003.

[11] 钱海军. 基于 BP 神经网络的图像压缩的 Matlab 实现[J]. 电脑开发与应用,2011(12):77-78.

[12] 朱艳秋,陈贺新,戴逸松. 彩色图像三维矩阵变换压缩编码[J]. 电子学报,1997,25(7):16-21.

[13] 魏政刚,袁杰辉,蔡元龙. 图像质量评价方法的历史、现状和未来[J]. 中国图形学报,1998,3(5):236-239.

[14] 张建宏. 基于混沌神经网络的图像压缩算法[J]. 煤炭技术,2010(5):167-168.

[15] 赵婷婷. 基于神经网络的视频加密与压缩技术的研究[D]. 大连:大连理工大学,2009.