

克隆选择算法的优化和品质因数^{*}

舒万能¹, 丁立新²

¹(中南民族大学 计算机科学学院, 湖北 武汉 430074)

²(武汉大学 计算机学院, 湖北 武汉 430072)

通讯作者: 丁立新, E-mail: lxding@whu.edu.cn, http://cs.whu.edu.cn



摘要: 针对传统的克隆选择算法可能存在的早熟收敛现象和缺少交叉操作问题, 提出一种高效的克隆退火优化算法. 该算法结合了模拟退火算法与免疫系统的克隆选择机制, 并保持全局搜索和局部搜索的平衡, 可以有效提高算法的搜索效率, 从而加快算法的收敛速度. 同时, 提出一种品质因数模型来分析该算法的动态性能, 并运用 Markov 链理论对其收敛性进行分析. 最后, 将该算法应用到关联规则数据挖掘中, 取得了较为理想的实验结果.

关键词: 品质因数; 收敛性; 马尔可夫链; 克隆选择算法; 群体多样性

中图法分类号: TP18

中文引用格式: 舒万能, 丁立新. 克隆选择算法的优化和品质因数. 软件学报, 2016, 27(11): 2763-2776. <http://www.jos.org.cn/1000-9825/4911.htm>

英文引用格式: Shu WN, Ding LX. Optimization and quality factor of clonal selection algorithm. Ruan Jian Xue Bao/Journal of Software, 2016, 27(11): 2763-2776 (in Chinese). <http://www.jos.org.cn/1000-9825/4911.htm>

Optimization and Quality Factor of Clonal Selection Algorithm

SHU Wan-Neng¹, DING Li-Xin²

¹(College of Computer Science, South-Central University for Nationalities, Wuhan 430074, China)

²(Computer School, Wuhan University, Wuhan 430072, China)

Abstract: To tackle the problem that traditional clonal selection algorithm may suffer from premature convergence phenomenon and is lack of crossover operator problems, this paper proposes a new efficient clonal annealing optimization algorithm. The proposed algorithm combines simulated annealing algorithm with clonal selection mechanism of immune system, and maintains the balance of global and local search. The algorithm can effectively improve search efficiency, so as to speed up the convergence rate. Meanwhile, a quality factor model is used to analyze the dynamic performance of the algorithm, and an analysis of its convergence is performed using Markov chain theory. Finally, the proposed algorithm is applied to the association rule data mining, achieving satisfactory results.

Key words: quality factor; convergence; Markov chain; clonal selection algorithm; population diversity

克隆选择算法(clonal selection algorithm, 简称 CSA)是基于生物免疫系统的克隆选择原理, 通过构造记忆单元实现全局与局部搜索平衡. 它由一个单个最优个体演变为最优解集, 体现免疫的多样性, 扩大了搜索区域^[1]. 在克隆选择算法中, 抗体多样性只是简单地通过新旧抗体的替换来实现的. 但是如果根据亲和度选出的抗体基因

* 基金项目: 国家自然科学基金(61603420); 湖北省自然科学基金(2014CFB413); 中南民族大学中央高校基本科研业务费专项资金(CZY14007); 科学计算与智能信息处理广西高校重点实验室科学开放基金(GXSCIP201412)

Foundation item: National Natural Science Foundation of China (61603420); National Natural Science Foundation of Hubei Province (2014CFB413); Special Fund for Basic Scientific Research of Central Colleges, South-Central University for Nationalities (CZY14007); Scientific Computing and Intelligent Information Processing of Guangxi University Key Laboratory of Science Open Fund (GXSCIP201412)

收稿时间: 2013-04-09; 修改时间: 2014-07-09; 采用时间: 2015-08-19; jos 在线出版时间: 2015-11-03

CNKI 网络优先出版: 2015-11-04 17:10:03, <http://www.cnki.net/kcms/detail/11.2560.TP.20151104.1710.004.html>

非常接近,那么所产生的后代相对于双亲也必然比较接近.这样,所期望的改善就比较小,基因模式的单一性不仅减慢了收敛速度,而且可能导致进化停滞,使算法过早地收敛于局部极值点.

在利用马尔可夫链分析克隆选择算法的收敛性时,设包含最优解的状态为 j ,当种群 $X(t_1)$ 处于某个状态 $i(i \neq j)$ 时,转移概率 $p_{ij}(t_1) \leq \alpha$ (α 为一个较小的正数),即此时种群 $X(t_1)$ 收敛于状态 j 的概率较小.如果当 $t_2 > t_1$ 时均有 $p_{ij}(t_2) \leq \alpha$,那么, t_1 时刻的种群 $X(t_1)$ 将发生准早熟收敛现象.当 $t_2 > t_1$ 时,种群 $X(t_2)$ 发生了早熟现象^[2].种群收敛现象的外在表现是,种群中的个体所包含的抗体基因型减少,使种群中个体趋于一致.如果种群不是朝着最优解的方向收敛,那么种群将收敛到局部最优解.在传统的克隆选择算法中,由于已经发生了早熟收敛的种群不能通过交叉操作生成新个体,如果仅仅依靠变异的作用产生新的抗体基因,这时将会发生早熟现象.许多研究表明,在某些情况下,交叉能够比变异提供更高水平的基因重组,从而产生多样抗体^[3].

传统的克隆选择算法模型虽然比较粗糙,对免疫克隆选择过程中的内在机理研究还不够深入,然而作为一个智能计算研究的新兴领域,一个受生物免疫系统启发、解决实际问题的智能方法,依然受到许多专家和学者的广泛关注.国内外人工免疫学者在算法构造与设计^[4]、内在克隆机理^[5]、新算子设计^[6]、自适应性研究^[7]、记忆能力^[8]、编码方式^[9]以及与其他智能计算方法相结合^[10]等方面进行了大量研究,并取得了众多研究成果.由于传统的克隆选择算法提出了一个基于信息熵的简单克隆算子,根据亲和度的排序增加克隆规模,没有对抗体克隆的自适应性和动态过程进行深入研究.因此,本文针对克隆选择算法的动态过程提出一种品质因数概念,并根据其可能存在的早熟收敛现象和缺少交叉操作问题,设计一种高效克隆退火优化算法(efficient clonal annealing optimizing algorithm,简称 ECAOA).它结合了模拟退火算法与免疫系统的克隆选择机制,并维持全局搜索和局部搜索的平衡,可以有效提高算法的搜索效率,从而加快算法的收敛速度.

1 ECAOA 算法的操作算子

模拟退火算法(simulated annealing,简称 SA)是源于对热力学中退火过程的模拟,在某一给定初温下,通过缓慢下降温度参数,使算法能够在多项式时间内给出一个近似最优解^[11].SA 算法首次被 Kirkpatrick 等人引入优化问题的求解,它具有很好的局部搜索能力,但是总体搜索能力较差.克隆选择算法虽然搜索速度较快,但是基因模式的单一性容易过早地收敛于局部极值点,导致局部搜索精度不高.如果将模拟退火机制融入到克隆选择原理中,则将实现克隆选择算法和模拟退火算法的优势互补,充分地平衡算法的局部和全局搜索能力.本文对 ECAOA 算法的相关操作算子进行如下定义:

定义 1. 设 T_k 是趋近于 0 的温度控制序列,则抗体 a_i 的退火选择概率为

$$p(a_i) = \frac{e^{-\frac{aff(a_i)}{T_k}}}{\sum_{i=1}^n e^{-\frac{aff(a_i)}{T_k}}} \quad (1)$$

退火选择概率表明抗体 a_i 以概率 $p(a_i)$ 选择进入下一代群体. T_k 表示进化到第 k 代的退火温度, k 表示当前的进化代数, $T_k = T_{k-1} \times \left(1 - \frac{k}{n}\right)$, n 表示最大进化代数.

定义 2. 设抗体 $a_i = \{\varphi_0, \dots, \varphi_{n-1}\}$, $a_j = \{\gamma_0, \dots, \gamma_{n-1}\}$, k 为交叉节点 ($1 < k < n$),则对抗体 a_i 和 a_j 执行的交叉操作 $T_c(a_i, a_j)$ 的具体操作步骤如下:

Step 1. 产生新抗体 a'_i 和 a'_j :

$$T_c(a_i, a_j) = \begin{cases} a'_i = \{\varphi_0, \dots, \varphi_k \in a_i, \varphi_{k+1}, \dots, \varphi_{n-1} \in a_j\} \\ a'_j = \{\gamma_0, \dots, \gamma_k \in a_j, \gamma_{k+1}, \dots, \gamma_{n-1} \in a_i\} \end{cases} \quad (2)$$

Step 2. 产生一个随机数 $r_c = \text{random}(0, 1)$, $r_c \in [0, 1]$.

Step 3. 分别计算父抗体 F 和子抗体 S 的亲和度值 $aff(F)$ 和 $aff(S)$.根据模拟退火算法的 Metropolis 接受准则,如果 $\min\{1, \exp(aff(S) - aff(F))/T_k\} > r_c$,则接受新的抗体 S , $F = \{a_i, a_j\}$, $S = \{a'_i, a'_j\}$, T_k 是第 k 次退火温度.

定义 3. 设抗体 $a_i = \{\varphi_0, \dots, \varphi_{n-1}\}$, k 为变异节点 ($1 < k < n$), 则对抗体 a_i 执行变异操作 $T_m(a_i)$ 的具体操作步骤如下:

Step 1. 产生子抗体 $a'_i: T_m(a_i) = a'_i = \{\varphi_0, \dots, \varphi_{k-1}, \varphi'_k, \dots, \varphi_{n-1}\}, \varphi'_k \oplus \varphi_{k-1} = 1$.

Step 2. 产生一个随机数 $r_m = \text{random}(0, 1), r_m \in [0, 1]$.

Step 3. 分别计算父抗体 a_i 和子抗体 a'_i 的亲合度值 $\text{aff}(a_i)$ 和 $\text{aff}(a'_i)$. 根据模拟退火算法的 Metropolis 接受准则, 如果 $\min\{1, \exp(-(\text{aff}(a'_i) - \text{aff}(a_i))/T_k)\} > r_m$, 就接受新的抗体 a'_i , T_k 是第 k 次退火温度.

定义 4. 设抗体 $a_i, a_j \in A(t)$, a_i 和 a_j 之间的增强因子为 $d_{ij} \in \{0, 1\}$, 则抗体 a_i 的增强度为

$$\text{Str}(a_i) = \sum_{j=1}^n d_{ij}, d_{ij} = \begin{cases} 1, & \text{aff}(a_i) \geq \text{aff}(a_j) \\ 0, & \text{aff}(a_i) < \text{aff}(a_j) \\ 0, & i = j \end{cases} \quad (3)$$

定义 5. 设抗体 a_i 的退火选择概率为 $p(a_i)$, 增强度为 $\text{Str}(a_i)$, 种群 $A(t)$ 的规模为 $|A(t)|$, 则选择因子 $\delta(a_i)$ 为

$$\delta(a_i) = p(a_i) \times |A(t)| \quad (4)$$

定义 6. 设抗体 a_i 的增强度表示为 $\text{Str}(a_i)$, 选择因子为 $\delta(a_i)$. 如果满足下列条件:

$$\text{Str}(a_1) + \text{Str}(a_2) + \dots + \text{Str}(a_{i-1}) < \delta(a_i) < \text{Str}(a_1) + \text{Str}(a_2) + \dots + \text{Str}(a_i),$$

则选择操作 $T_s(A(t))$ 为

$$T_s(A(t)) = a_i \quad (5)$$

定义 7. 设 $\overline{\text{aff}^*}$ 表示种群 $A(t)$ 的平均亲合度值, $\text{aff}(a_i)$ 表示抗体 a_i 的亲合度值, PopSize 表示种群规模, 则准成熟抗体群 $A_{\text{best}}(t)$ 为

$$\begin{cases} A_{\text{best}}(t) = \{a_i \mid \text{aff}(a_i) \geq \overline{\text{aff}^*}, a_i \in A(t)\} \\ \overline{\text{aff}^*} = \frac{\sum_{i=1}^{\text{PopSize}} \text{aff}(a_i)}{\text{PopSize}} \end{cases} \quad (6)$$

定义 8. 抗体距离 $H(a_i, a_j)$:

$$\begin{cases} d_{ij} = a_i \oplus a_j \\ H(a_i, a_j) = \sum_{k=1}^L d_{ij}(k) / L \end{cases} \quad (7)$$

上述的抗体距离 $H(a_i, a_j)$ 采取了归一化处理, L 为编码长度, $d_{ij}(k)$ 为 d_{ij} 的第 k 位编码, \oplus 为按位异或运算符. 当 $H(a_i, a_j) = 0$ 时, 抗体 a_i 和抗体 a_j 的所有基因相同; 当 $H(a_i, a_j) = 1$ 时, 抗体 a_i 和抗体 a_j 的所有基因均不相同, 抗体之间差异最大.

定义 9 (δ 领域). 如果抗体 a_i 和抗体 a_j 的距离 $H(a_i, a_j)$ 满足:

$$\lim_{\delta \rightarrow 0} (H(a_i, a_j) - \delta) = 0 \quad (8)$$

则称 δ 为抗体 a_i 的 δ 领域.

抗体 a_i 的 δ 领域 (或称为抗体相似度阈值) 反映的是与抗体 a_i 相似的抗体个数情况. 我们用 $m_i(\delta)$ 表示 a_i 的 δ 领域内包含种群 $A(t)$ 中抗体的个数. 下面通过 δ 领域的相关属性, 定义抗体的浓度一般性公式. 抗体的浓度表示基因相同 (或者近似) 的抗体在整个抗体种群中所占的比例^[12], 抗体的浓度定义如下:

定义 10. 抗体浓度 $C(a_i)$:

$$C(a_i) = \frac{m_i(\delta)}{\text{PopSize}} \quad (9)$$

其中, PopSize 为当前种群的规模. 抗体的浓度用于调节抗体的克隆规模, 当某些抗体的亲合度较大时, 会被克隆扩增, 其浓度会快速上升, 从而导致少数几种类型的抗体充斥整个免疫系统. 因此, 应当采取一定的措施抑制那些浓度较高的抗体^[13]. 抗体的生存度 (或称为激励度) 反映的是抗体应答抗原和被其他抗体激活的综合能力^[14]. 在免疫应答中, 为了维持免疫响应的稳定平衡, 具有高亲合度且低浓度的抗体将受到促进, 而低亲合度且浓度较高的抗体将受到抑制. 通过这种相似细胞之间的抑制作用, 抗体种群的多样性得到保持, 同时增强了算法的全局

搜索能力和多解搜索能力.本文定义抗体 a_i 的生存度函数如下:

定义 11. 抗体生存度 $S(a_i)$:

$$S(a_i) = |\lambda \times aff(a_i) - \beta \times C(a_i)| \quad (10)$$

其中, $aff(a_i)$ 和 $C(a_i)$ 分别表示抗体 a_i 的亲度和浓度, λ 和 β 为控制参数, $S(a_i) > 0, \lambda + \beta = 1$.

定义 12. 设种群 $A(t)$ 经过克隆、变异、交叉和选择操作后得到的准成熟抗体群为 $A_{best}(t)$, 抗体 a_i 的生存度表示为 $S(a_i)$, 则克隆抑制操作 $T_r(A_{best}(t))$ 为

$$T_r(A_{best}(t)) = a, S(a) = \max \{S(a_i), a_i \in A_{best}(t)\} \quad (11)$$

克隆抑制算子能够有效地抑制浓度高的抗体, 促进亲度和高的抗体进入新的抗体种群.

定义 13. 设 ECAOA 算法的种群为 $A(t)$, 准成熟抗体群表示为 $A_{best}(t)$, 非成熟抗体群表示为 $\overline{A}(t) = A(t) - A_{best}(t)$, 随机生成的新抗体为 b , 则种群更新操作 $T_d(\overline{A}(t))$ 为

$$T_d(\overline{A}(t)) = b \quad (12)$$

种群更新算子 T_d 是对种群中亲度较低的抗体进行更新, 从抗体种群中删除这些抗体并以随机生成的新抗体替代, 有利于保持抗体的多样性, 探索新的可行解空间区域.

2 ECAOA 算法的流程

定义 14. 假设克隆操作表示为 T_k , 变异操作表示为 T_m , 交叉操作表示为 T_c , 选择操作表示为 T_s , 克隆抑制操作表示为 T_r , 种群更新操作表示为 T_d , 则 ECAOA 算法可以描述为六元组 $ECAOA = (T_k, T_m, T_c, T_s, T_r, T_d)$.

ECAOA 算法的克隆操作 T_k 采取 CSA 算法的克隆操作方式.

本文设计的 ECAOA 算法的具体流程如图 1 所示.

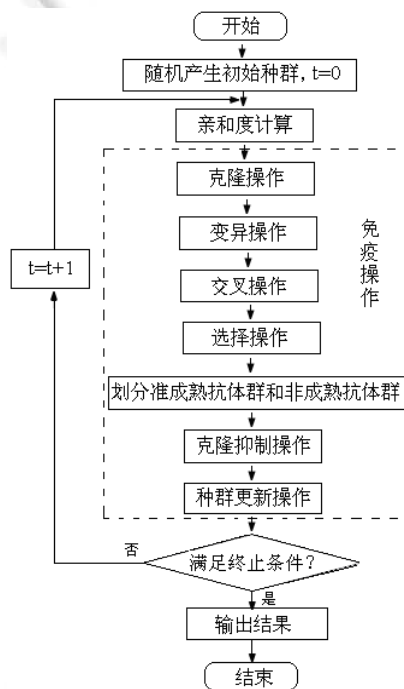


Fig.1 Flow chart of ECAOA algorithm

图 1 ECAOA 算法的流程图

ECAOA 算法的执行步骤如下:

Step 1. 随机产生一个初始种群 $A(t)=\{a_i|i=1,2,\dots,PopSize\}$,种群的规模 $|A(t)|=PopSize$,并对每个抗体 a_i 进行编码, $a_i=\varphi_1,\dots,\varphi_n,|a_i|=n$,初始进化代数 $t=0$,初始退火温度 $T_t=1$.

Step 2. 评估每个抗体的亲和度.

Step 3. 根据 CSA 算法的克隆操作,对种群 $A(t)$ 执行克隆操作 T_k .

Step 4. 执行变异操作 T_m .

Step 5. 执行交叉操作 T_c .

Step 6. 执行选择操作 T_s ,并将选择出来的抗体保留到下一代.

Step 7. 将种群 $A(t)$ 划分为 $A_{best}(t)$ 和 $\overline{A}(t)$.

Step 8. 对准成熟抗体群 $A_{best}(t)$ 执行克隆抑制操作 T_r .

Step 9. 对非成熟抗体群 $\overline{A}(t)$ 执行种群更新操作 T_d .

Step 10. 令 $T_t = T_{t-1} \times \left(1 - \frac{t}{PopSize}\right)$, $t=t+1$. 如果 $T_t \approx 0$, 则算法运行结束; 否则, 跳转到 Step 2.

3 ECAOA 算法性能的评价指标

本文为了更好地对算法的性能进行评估,特对以下一些相关的算法性能评价指标进行定义.

定义 15. 线性性能 $X(s)$:

$$X(s) = \frac{1}{t} \sum_{t=1}^T \overline{aff}(t) \quad (13)$$

$\overline{aff}(t)$ 是在第 t 时刻的平均亲和度值.算法的线性性能指标表示了算法从开始运行一直到当前为止的时间段内性能指标的平均值,它反映了算法的动态性能.

定义 16. 离线性能 $X^*(s)$:

$$X^*(s) = \frac{1}{t} \sum_{t=1}^T aff^*(t) \quad (14)$$

$aff^*(t) = \text{best}\{aff(1), aff(2), \dots, aff(t)\}$ 是在 $[0, t]$ 时间内亲和度值最大的. $X^*(s)$ 表示算法在运行过程中最佳性能值的累积平均,反映了算法的收敛性能.

定义 17. 设 ECAOA 算法的线性性能表示为 $X(s)$, 离线性能表示为 $X^*(s)$, 则算法的品质因数 Q 为

$$Q = \frac{2 \times X(s) \times X^*(s)}{X(s) + X^*(s)} \quad (15)$$

品质因数 Q 结合了算法的线性性能和离线性能. Q 值越高, 表明算法的综合性能越优良.

定义 18. 误差性能 $E(s)$:

$$E(s) = \frac{1}{t} \sum_{t=1}^T \frac{S(t) - S^*}{S^*} \quad (16)$$

其中, $S(t)$ 是算法在第 t 时刻运行所得的实际优化值, S^* 为问题的期望最优值. 算法的误差性能 $E(s)$ 指标反映了算法从开始运行一直到当前为止的时间段内对问题的最佳优化程度, 其值越小, 意味着算法的优化性能越好.

定义 19. 设种群的规模为 $|A(t)|$, 50 次算法独立运行到指定的迭代数时不同的抗体数之和为 Sum , 则群体多样性 M_u 为

$$M_u = \frac{Sum}{50 \times |A(t)|} \quad (17)$$

4 ECAOA 算法的收敛性分析

为了从理论上更好地分析 ECAOA 算法的收敛性能, 本文首先引入波雷尔-坎特利(Borel-Cantelli)引理和 Markov 链的相关基本理论.

引理 1(波雷尔-坎特里(Borel-Cantelli)引理)^[15]. 设 $A_1, A_2, \dots, A_k, \dots, A_n$ 是概率空间上的一事件序列, 令 $P_k = P\{A_k\}$. 若 $\sum_{k=1}^{\infty} P_k < \infty$, 则

$$P\left\{\bigcap_{n=1}^{\infty} \bigcup_{k \geq n} A_k\right\} = 0 \tag{18}$$

若 $\sum_{k=1}^{\infty} P_k = \infty$, 且各 A_k 相互独立, 则

$$P\left\{\bigcap_{n=1}^{\infty} \bigcup_{k \geq n} A_k\right\} = 1 \tag{19}$$

算法的收敛是指当算法迭代足够多的次数以后, 群体中包含全局最佳个体的概率近于 1. 根据波雷尔-坎特里引理和文献[16], 本文引入对算法收敛性的具体定义:

定义 20^[16]. 设 $\{X_n | n \geq 0\}$ 概率空间上定义的随机序列, 若存在随机变数 S^* , 使

$$P\{\lim_{n \rightarrow \infty} (X_n = S^*)\} = 1 \tag{20}$$

或者对于 $\forall \varepsilon > 0$, 有:

$$P\left\{\bigcap_{n=1}^{\infty} \bigcup_{k \geq n} [|X_n - S^*| \geq \varepsilon]\right\} = 0 \tag{21}$$

则称随机序列 $\{X_n | n \geq 0\}$ 以概率为 1 收敛于随机变数 S^* .

在 ECAOA 算法中, 初始种群的产生过程是一个随机过程, 每一次新的种群产生只与其父代种群相关, 并且种群所处的状态为离散状态, ECAOA 算法的这些特点基本符合 Markov 链的特征. 因此, 下面我们将用 Markov 链对 ECAOA 算法的收敛性进行分析.

定义 21(有限 Markov 链)^[16]. 随机过程可能取到的值(状态)组成的集合记为 S , 称为随机过程的状态空间. 设随机变量序列 $\{X_n | n \geq 0\}$ 取值于可数状态空间 $S = N^+ = \{1, 2, \dots, n, \dots\}$, 若对于任意的状态 $i_0, i_1, \dots, i_{n+1} \in S$, 都有

$$P\{X_{n+1} = i_{n+1} | X_n = i_n, \dots, X_1 = i_1, X_0 = i_0\} = P\{X_{n+1} = i_{n+1} | X_n = i_n\} \tag{22}$$

成立, 则称 $\{X_n | n \geq 0\}$ 为可数状态 Markov 链. 如果状态集 S 中状态是有限的, 则称此 Markov 链 $\{X_n | n \geq 0\}$ 为有限 Markov 链.

定义 22^[16]. 如果对任意的 $i, j \in S$, Markov 链 $\{X_n | n \geq 0\}$ 的一步转移概率 $p_{ij}(n)$ 与初始时刻 n 无关, 则称该 Markov 链是齐次的 Markov 链.

定理 1. ECAOA 算法的种群序列 $\{X_n | n \geq 0\}$ 是有限的齐次 Markov 链.

证明: 本文用 m 表示为每个抗体的基因位数, l 为每位基因用二进制表示的位数, $|A(t)|$ 为种群规模. 设 Ω 为所有抗体的空间, 其每个坐标分别是 Ω 中的抗体. 亲和度函数 $aff(x)$ 是集合 $\Omega \subset R^n$ 上大于 0 的连续函数, 并定义若 $x \notin \Omega$, 则 $aff(x) = 0$. 抗体空间 $\Omega = \{0, 1\}^{ml}$ 中共有 2^{ml} 个状态, 所以抗体空间 $\Omega^{A(t)}$ 所在的状态空间大小为 $|A(t)| \times 2^{ml}$. 因此, 抗体种群序列空间中的状态数目是有限的. \square

ECAOA 算法的状态转移情况可以用如下的随机过程来描述:

$$A(t) \xrightarrow{\text{克隆}} B(t) \xrightarrow{\text{变异}} C(t) \xrightarrow{\text{交叉}} D(t) \xrightarrow{\text{选择}} E(t) \xrightarrow{\text{克隆抑制}} F(t) \xrightarrow{\text{种群更新}} A(t+1),$$

其中, 下标 t 表示进化的代数. 令 $T = T_d \circ T_r \circ T_s \circ T_c \circ T_m \circ T_k$, 则 $x_i(n+1) = T[X(n)]_i = T_d \circ T_r \circ T_s \circ T_c \circ T_m \circ T_k[X(n)]_i$.

根据前面对 ECAOA 算法各算子的分析以及前面的定义可知, ECAOA 算法的种群序列状态转移概率如下:

$$\begin{aligned} p_{ij}(n) &= p_n\{X(n+1) = Y | X(n) = X\} \\ &= p_n\{Y | X\} \\ &= \prod_{i=1}^{|A(t)|} \left\{ \sum_i p\{T_k(X(N))_i = x_i(n)\} \times p\{T_m(x_i(n)) = x_i(n)\} \times p\{T_c(x_i(n)) = x_i(n)\} \times \right. \\ &\quad \left. p\{T_s(x_i(n)) = x_i(n)\} \times p\{T_r(x_i(n)) = y_i(n)\} \right\} \times \prod_{i=1}^{|A(t)|} p\{T_d(Y(n)) = x_i(n+1)\}. \end{aligned}$$

根据前面对 ECAOA 算法的克隆、变异、交叉、选择、克隆抑制和种群更新操作的定义,计算出的 ECAOA 算法中抗体群的状态转移概率 $p_{ij}(n)$ 与参数时刻 n 无关.所以,ECAOA 的种群序列 $\{X_n|n \geq 0\}$ 是有限的齐次 Markov 链.

在模拟退火算法中,模拟退火的过程可以描述为从一个状态转移到另一个状态的随机游动.利用 Markov 链的性质,分析模拟退火算法对应 Markov 链的周期性和不可约性,可以证明在一定条件下模拟退火算法的全局收敛性^[16].根据模拟退火的 Metropolis 准则,如果进行以下假设:当 $\Delta f=f(X)-f(X')>0$ 时,设 $\Delta f \geq \tau$, τ 为某个充分小的正数,或比温度 t 的收敛阶小,根据文献[17]中的证明,可以得到如下结论:

定理 2. 设在一定温度 t 下,模拟退火算法产生的点列为 $X^{(k+1)}(t)$,若新解产生满足 $\Delta \sim N(0, \sigma^2)$,则当温度 $t \rightarrow 0$ 时,点列 $\{X^{(k)}(t)\}_{k=1}^{+\infty}$ 以概率为 1 收敛于全局极小点,即

$$P(\lim_{k \rightarrow \infty} \lim_{t \rightarrow 0} f(X^{(k)}(t)) = f^*) = 1 \quad (23)$$

在文献[17]中,黄友锐等人利用齐次有限马尔可夫链的方法证明了基于二进制编码的克隆选择算法的全局收敛性能:

定理 3. 二进制编码的克隆选择算法 CSA 是以概率 1 收敛的.

推论 1. ECAOA 算法是以概率 1 收敛的.

证明:本文提出的高效克隆退火优化算法,是在克隆选择算法的基础上引入了模拟退火的思想.由于本文设计的 ECAOA 算法只是在经典的克隆选择算法的基础上采用模拟退火进制进行种群抗体的改良操作,而模拟退火机制根据定理 2 中的描述是以概率为 1 收敛于全局极小点,ECAOA 算法的执行过程并没有改变克隆选择算法的本质特征.因此,定理 3 对 ECAOA 算法同样适用.我们设 $\{X_n|n \geq 0\}$ 是 ECAOA 算法的一个随机变量序列,那么当算法迭代到足够多的次数后,下面的公式一定成立:

$$P\{\lim_{n \rightarrow \infty} (X_n = S^*)\} = 1 \quad (24)$$

根据定义 15 可知,种群序列 $\{X_n|n \geq 0\}$ 中包含全局最优解的概率接近 1,ECAOA 算法是以概率 1 收敛的. □

5 ECAOA 算法的计算复杂度分析

本节分析 ECAOA 算法的计算复杂度主要考虑时间复杂度,即主要通过算法指令被执行最大次数来估计算法运行所需的计算量.为了更好地描述算法的时间复杂度,特进行以下定义:

定义 23. 给定自然数 n 的两个函数 $F(n)$ 和 $G(n)$,当且仅当存在一个正常数 K 和 n_0 ,使得当 $n \geq n_0$ 时,有 $F(n) \leq KG(n)$,则称函数 $F(n)$ 以函数 $G(n)$ 为界,记作 $F(n) = O(G(n))$.

在上述定义中, O 表示数量级的概念.在分析算法的时间复杂度时,可通过复杂度函数主要项的阶 $O(G(n))$ 进行表示.ECAOA 算法中包含 3 个参数,即,种群规模 $N=|A(t)|$ 、抗体的基因位数 l 、群体的克隆规模 M .算法主要包括以下 7 个操作:(1) 计算抗体的亲和度值;(2) 克隆操作;(3) 变异操作;(4) 交叉操作;(5) 选择操作;(6) 克隆抑制操作;(7) 种群更新操作.在抗体群中,由于要对每个抗体计算亲和度值,所以操作(1)的执行次数是 N .操作(2)对抗体群进行克隆操作所需的执行次数是 M .操作(3)采取变异操作所需的最多次数为 N .操作(4)采取的交叉操作所需的计算量最多为 $\log_2 N$.根据本文对选择操作的定义,操作(5)采取的选择操作由于要计算每个抗体的增强度,导致所需的计算量较大,最多次数为 $(N-1) \times N + 1$.操作(6)和操作(7)所采取的操作需要的执行次数均是 1.由于 $N > M, N > \log_2 N$,于是,第 n 次迭代中 ECAOA 算法执行操作的总次数 S_n 应该符合如下的条件:

$$S_n \leq N + M + N + \log_2 N + (N-1) \times N + 1 + 2 < N^2 + 3N.$$

因此,ECAOA 算法的计算量最高级为 $O(N^2)$,这表明种群规模直接影响到算法的搜索速度.在 CSA 算法中,抗体要依次经过克隆、变异、选择、克隆抑制和种群更新等操作,计算复杂度的最高级别也为 $O(N^2)$ ^[17].但是,ECAOA 算法通过引入交叉操作和模拟退火的策略,每次产生的新抗体都先进行 Metropolis 准则判断,有效保证子抗体比父抗体更加优良,从而跳出局部极值,避免了 CSA 算法可能出现的早熟收敛现象,因而同一进化代数中能够更快、更好地找到满意解,使得满足同样收敛条件的时间要快于 CSA 算法.

6 ECAOA 算法在数据挖掘中的应用

近年来,信息爆炸性的递增导致 Web 用户面对越来越多的海量数据,如何从海量的数据中提取有用的知识已经成为当务之急.数据挖掘(data mining)就是为了顺应这种需要发展起来的一种数据处理技术.现代数据挖掘方法主要依靠模式提取技术,为了改善和提高数据挖掘功能、性能和效率,发展趋势是集成多种技术的综合方法^[18],如,将免疫理论与统计学集成的免疫控制图挖掘方法、基于免疫策略的数据挖掘方法、基于免疫规划的广义规则推理方法、基于免疫多变量子波神经网络的数据挖掘方法等.为了检验本文设计的 ECAOA 算法的有效性,下面将 ECAOA 算法应用到关联规则挖掘中.

为了验证本文设计的 ECAOA 算法的有效性,分别基于 census 数据集和 chess 数据集平台进行了两类对比实验:

- (1) ECAOA 算法与其他智能优化算法:遗传算法(genetic algorithm,简称 GA)算法、SA 算法和 CSA 算法.
- (2) ECAOA 算法与其他 3 种典型的关联规则挖掘算法:Apriori 算法、DHP 算法和 FP-Growth 算法.Apriori 算法是一种最有影响的挖掘布尔关联规则频繁项集的算法.DHP(direct hashing and pruning)是一种基于散列技术的 Apriori 算法的改良方法.FP-Growth(frequent pattern growth)算法在保留项集关联信息的同时,将提供频繁项集的数据库压缩成一棵频繁模式树(FP-tree),同时依然保留其中的关联信息.

Census 数据集是由美国加利福尼亚大学信息与计算机科学系提供的、专供数据挖掘和机器学习用的一个测试数据库.它的大小为 10.9M,共包含 48 842 条记录,每条记录有 14 个属性.Chess 数据集共包含 3 196 条记录,共含有 75 个项,属于一种稠密的数据集^[19].

第 1 类实验主要从响应时间、执行时间、品质因数、误差性能和群体多样性这 5 个方面进行比较,在 census 数据集上的实验结果如图 2~图 6 所示.

图 2 的实验结果显示:在进化前期(进化代数大约为 70),GA 算法的性能表现最优,CSA 算法的性能表现最差,ECAOA 算法的性能稍微强于 SA 算法;在进化的中期(进化代数大约为 70~160),ECAOA 算法的性能表现最优,GA,CSA 和 SA 三者之间的性能互有优劣;在进化的后期,GA 算法最先收敛,ECAOA 算法的优势表现得最为明显,其响应时间大概只是 GA 算法的 71.42%,SA 算法和 CSA 算法的响应时间几乎相同.

在图 3 和图 4 中,ECAOA 算法在进化中后期的执行时间和品质因数方面都表现出良好的性能,其执行时间比 CSA 算法大约少 9.87%,比 SA 算法大约少 11.23%,比 GA 算法大约少 13.89%,其品质因数比 CSA 算法大约提高 6.85%,比 SA 算法大约提高 17.67%,比 GA 算法大约提高 21.36%.

在图 5 显示的误差性能方面,4 种算法在不同进化代数时测试出的误差性能基本都不相同,ECAOA 算法在大多数情况下优于其他 3 种算法.在图 6 的实验结果中,ECAOA 算法在大约进化到 155 次时获得了最高的群体多样性值 3.62.在进化的前半阶段,随着进化的逐渐深入,4 种算法的群体规模在不断地扩大,从而群体的多样性在显著地增强;在进化的后半阶段,随着算法逐渐找到满意解并缓慢开始收敛,算法的规模在逐步缩小导致群体的多样性显著的降低.ECAOA 算法的平均群体的多样性值为 1.75,CSA 算法的平均群体的多样性值为 1.68,SA 算法的平均群体的多样性值为 1.65,GA 算法的平均群体的多样性值为 1.61.

在 chess 数据集上的实验结果如图 7~图 11 所示.图 7 的实验结果显示:在进化前期(进化代数大约为 140),CSA 算法的性能表现最优,ECAOA 算法的性能和 GA 算法比较接近,但稍微强于 SA 算法;在进化的后期,ECAOA 算法的性能表现最优,其响应时间大概只有 GA 算法的 76.83%,只有 SA 算法的 84.41%,只有 CSA 算法的 82.67%.在图 8 中,ECAOA 算法的执行时间在进化中后期的大多数情况下都表现出良好的性能,但是 4 种算法的整体性能都非常接近.在图 9 中,ECAOA 算法的品质因数在进化期间都表现得非常稳定,其品质因数比 CSA 算法大约提高 1.82%,比 SA 算法大约提高 12.42%,比 GA 算法大约提高 9.84%.

在图 10 显示的误差性能方面,4 种算法在不同进化代数时,ECAOA 算法的平均误差性能大约是 0.32,CSA,GA 和 SA 算法的平均误差性能分别大约是 0.68,0.71,0.82.在图 11 的实验结果中,ECAOA 算法在大约进化到 168 次时,获得了最高的群体多样性值 3.41.群体的多样性在进化的前期都逐渐扩大,在进化的后期都显著地降低.ECAOA,CSA,SA,GA 算法的平均群体的多样性值分别为 1.65,1.61,1.55,1.58.

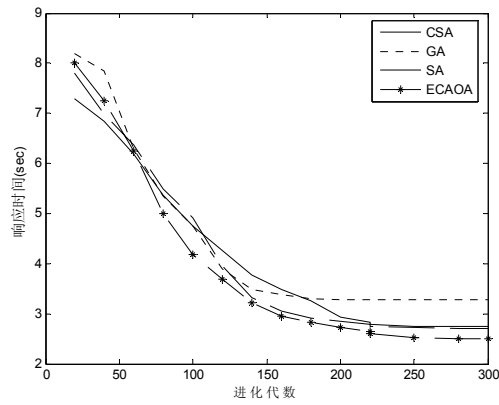


Fig.2 Comparison of the response time of the four algorithms in census data set
图2 Census数据集上4种算法的响应时间比较

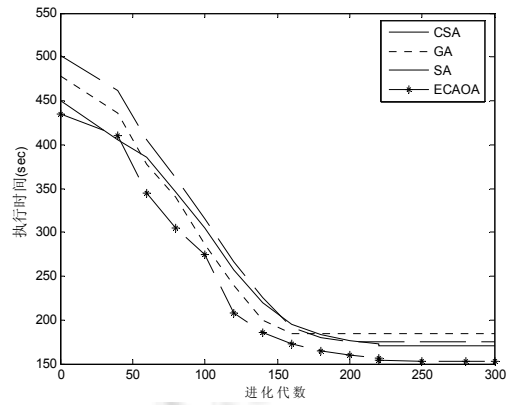


Fig.3 Comparison of the execution time of the four algorithms in census data set
图3 Census数据集上4种算法的执行时间比较

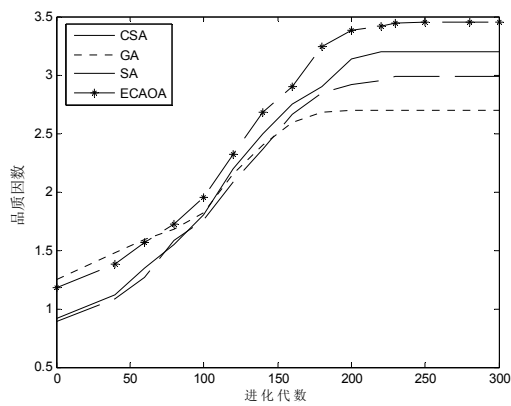


Fig.4 Comparison of the quality factor of the four algorithms in census data set
图4 Census数据集上4种算法的品质因数比较

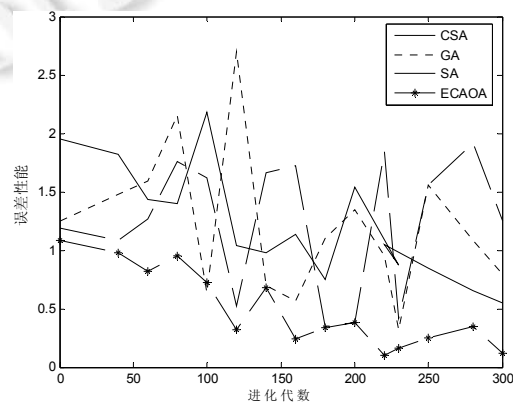


Fig.5 Comparison of the error performance of the four algorithms in census data set
图5 Census数据集上4种算法的误差性能比较

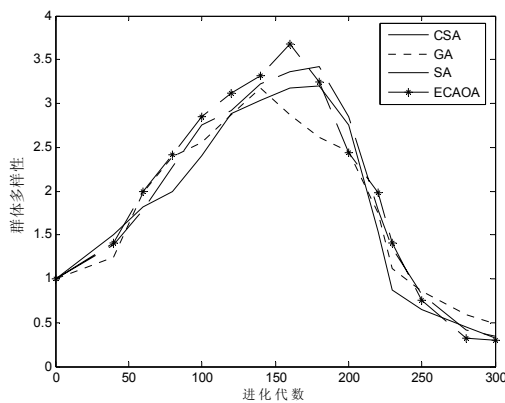


Fig.6 Comparison of the population diversity of the four algorithms in census data set
图6 Census数据集上4种算法的群体多样性比较

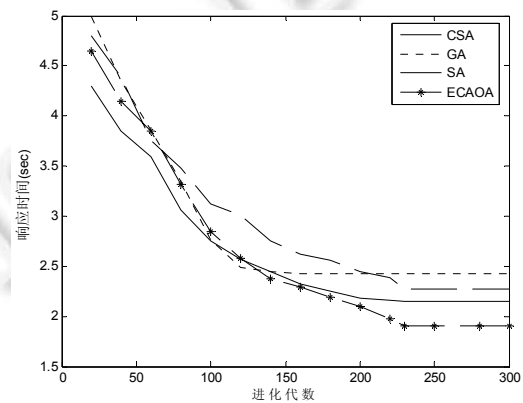


Fig.7 Comparison of the response time of the four algorithms in chess data set
图7 Chess数据集上4种算法的响应时间比较

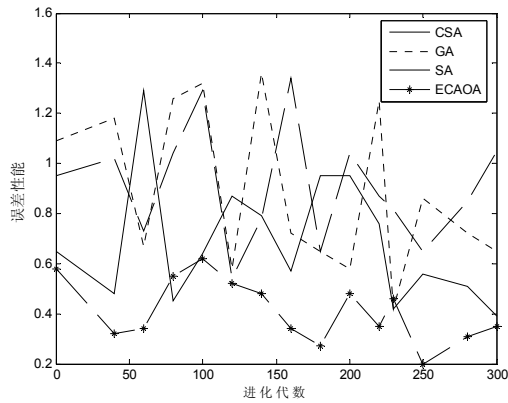


Fig.8 Comparison of the execution time of the four algorithms in chess data set

图8 Chess 数据集上 4 种算法的执行时间比较

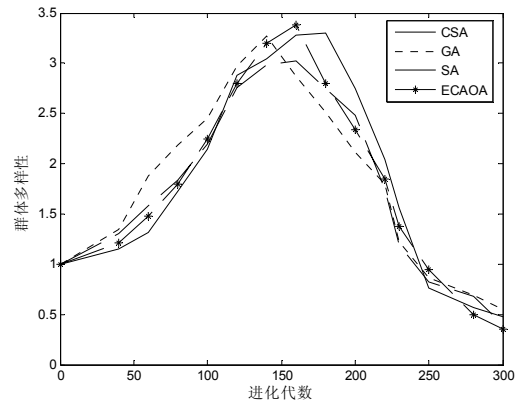


Fig.9 Comparison of the quality factor of the four algorithms in chess data set

图9 Chess 数据集上 4 种算法的品质因数比较

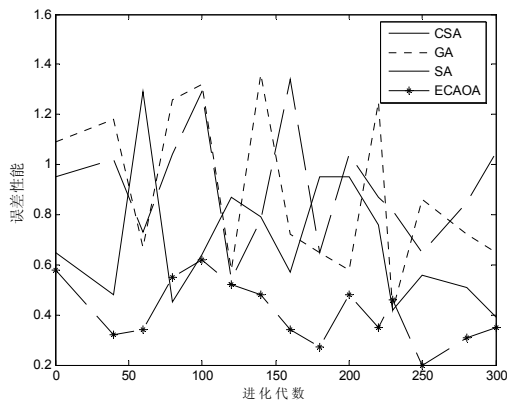


Fig.10 Comparison of the error performance of the four algorithms in chess data set

图10 Chess 数据集上 4 种算法的误差性能比较

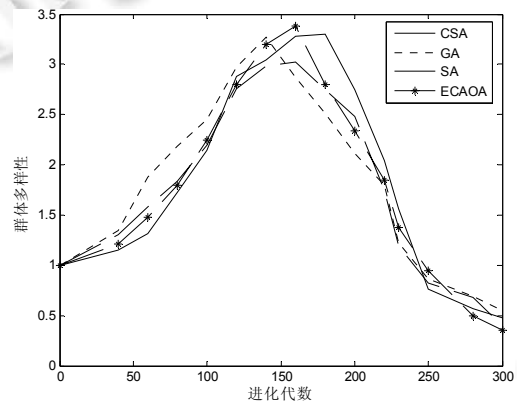


Fig.11 Comparison of the population diversity of the four algorithms in chess data set

图11 Chess 数据集上 4 种算法的群体多样性比较

通过第 1 类实验,我们不难分析出,ECAOA 算法在响应时间、执行时间、品质因数、误差性能和群体多样性这 5 方面都表现出良好的性能.这主要是因为 ECAOA 算法在变异和交叉等操作产生新的抗体后,首先通过模拟退火的 Metropolis 准则来决定新抗体是否保留,从而有效保证新的抗体具有更好的性能.ECAOA 算法将 CSA 算法和 SA 算法的优点进行结合,有效避免了陷入局部极值,能够有效改良算法的品质因数、误差性能和群体多样性等性能指标参数.在进化前期,ECAOA 算法的性能与其他 3 种算法相比并没有表现出明显的优势,相反,在局部情况下,其性能还比较落后.导致这种情况的原因是,ECAOA 算法在进行首次选择操作时要计算每个抗体的增强度,从而增大了所需的计算量.CSA 算法在绝大多数情况下的性能要优于 SA 算法和 GA 算法,主要是克隆操作不仅扩大了解的空间范围,而且提高了优良抗体的比例,这样将积极影响到群体多样性和误差性能等性能指标参数.GA 算法在大多数情况下的性能表现不如其他 3 种算法,主要是因为该算法很容易出现早熟收敛现象,导致其群体多样性较差,误差性能较大.

为了更好地评估 ECAOA 算法的性能,本节结合算法的执行时间、挖掘的关联规则数目和关联规则准确率对算法的挖掘质量进行如下定义:

定义 24. 如果算法挖掘的关联规则数目用 $total$ 表示,算法的执行时间用 t_e 表示,关联规则的准确率用 θ 表

示,则算法的挖掘质量 ϕ 描述如下:

$$\phi = \frac{total \times \theta}{t_e} \quad (25)$$

从公式(25)中不难分析出:执行时间 t_e 越少,关联规则数目 $total$ 越大,准确率 θ 越高,则算法的挖掘质量 ϕ 越大,算法的整体性能越优良.

第 2 类实验主要从运行时间、关联规则数目、准确率和挖掘质量这 4 个方面进行比较,在 census 数据集上的实验结果如图 12~图 15 所示.

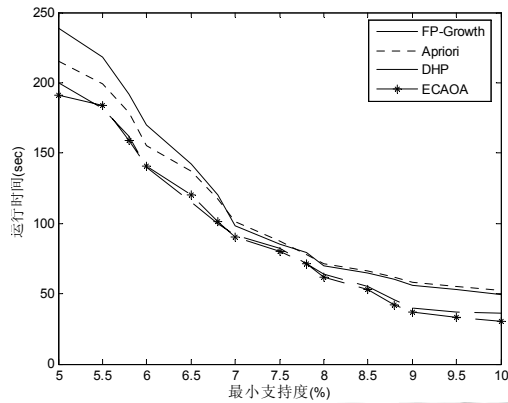


Fig.12 Comparison of the execution time of the four algorithms in census data set

图 12 Census 数据集上 4 种算法的运行时间比较

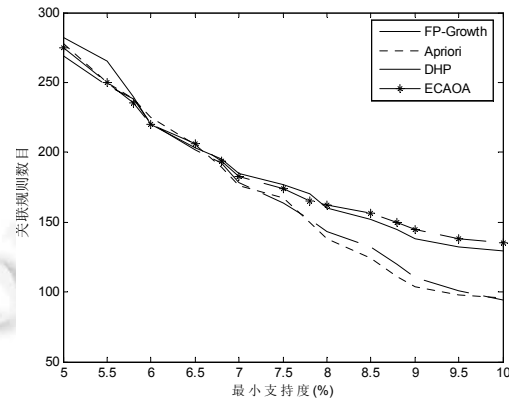


Fig.13 Comparison of number of association rules of the four algorithms in census data set

图 13 Census 数据集上 4 种算法的关联规则数目比较

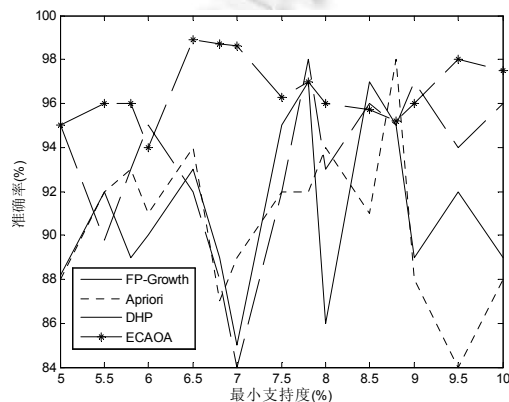


Fig.14 Comparison of the accuracy rate of the four algorithms in census data set

图 14 Census 数据集上 4 种算法的准确率比较

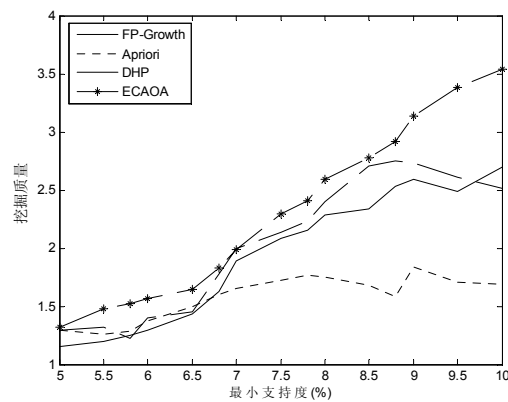


Fig.15 Comparison of the mining quality of the four algorithms in census data set

图 15 Census 数据集上 4 种算法的挖掘质量比较

在图 12 的执行时间对比实验中,当最小支持度小于 7.8%时,ECAOA 算法和 DHP 算法的性能非常接近,Apriori 算法和 FP-Growth 算法的性能表现得相对较差;当最小支持度大于 7.8%时, ECAOA 算法的优势最明显,而 Apriori 算法和 FP-Growth 算法的执行时间几乎相等,其运行时间比 DHP 算法大约要少 10s,比 Apriori 算法和 FP-Growth 算法大约要少 28s.在图 13 中,当最小支持度在 5.75%~7%范围时,4 种挖掘算法得到的关联规则数目非常接近;当最小支持度大于 7%时,ECAOA 算法和 DHP 算法都表现出良好的性能,其挖掘出的关联规则数目比 DHP 算法大约多 5.07%,比 FP-Growth 算法和 Apriori 算法大约多 25.23%.

在图 14 中,ECAOA 算法的准确率平均达到 95.88%,FP-Growth,Apriori 和 DHP 算法的准确率平均分别为 90.39%,94.08%,93.92%。在图 15 显示的挖掘质量对比中,ECAOA 算法的优势表现得非常明显,随着最小支持度的逐渐变大,ECAOA 算法的挖掘质量明显提高。

在 chess 数据集上的实验结果如图 16~图 19 所示。

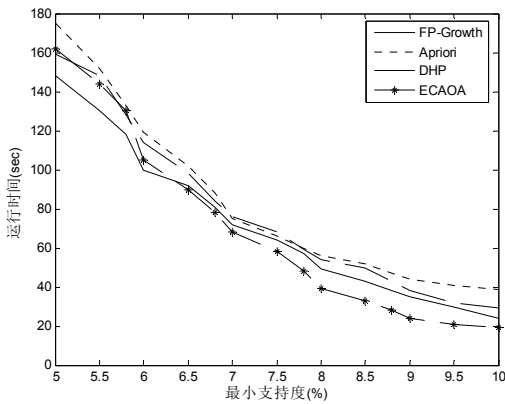


Fig.16 Comparison of the execution time of the four algorithms in chess data set

图 16 Chess 数据集上 4 种算法的运行时间比较

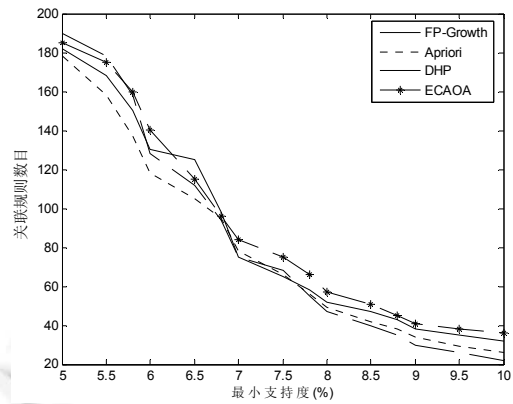


Fig.17 Comparison of number of association rules of the four algorithms in chess data set

图 17 Chess 数据集上 4 种算法的关联规则数目比较

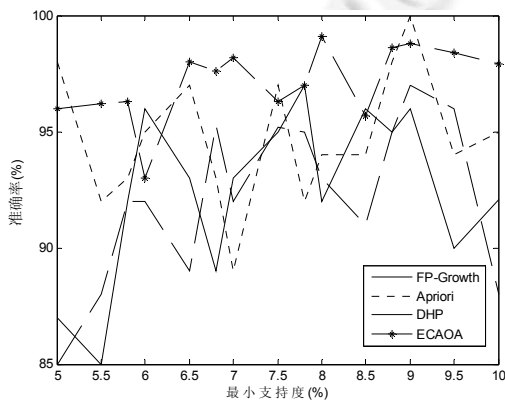


Fig.18 Comparison of the accuracy rate of the four algorithms in chess data set

图 18 Chess 数据集上 4 种算法的准确率比较

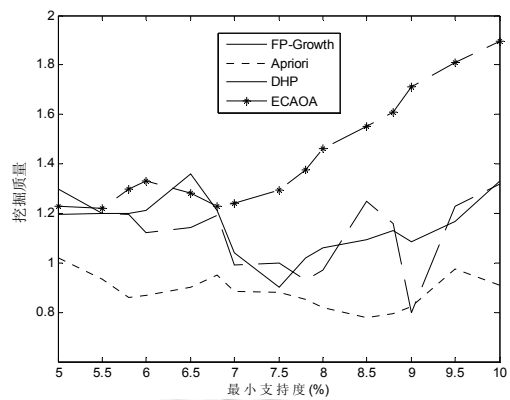


Fig.19 Comparison of the mining quality of the four algorithms in chess data set

图 19 Chess 数据集上 4 种算法的挖掘质量比较

在图 16 的运行时间对比实验中,当最小支持度小于 6.5%时,ECAOA 算法与 Apriori 算法和 DHP 算法三者表现出来的性能互有优劣,但是 FP-Growth 算法的性能表现得相对较好;当最小支持度大于 6.5%时,ECAOA 算法的优势最明显,随着最小支持度逐渐变大,ECAOA 算法的运行缓慢减少。在图 17 中,当最小支持度小于 6.8%时,ECAOA 算法的性能稍微优于 Apriori,DHP,FP-Growth 算法;当最小支持度大于 6.8%时,ECAOA 算法表现出优良的性能,其挖掘出的关联规则数目比 FP-Growth 算法大约多 9.38%,比 Apriori 算法大约多 21.45%,比 DHP 算法大约多 23.68%。

在图 18 中,ECAOA 算法的准确率平均达到 97.05%,FP-Growth,Apriori 和 DHP 算法的准确率平均分别为 93.12%,94.67%,90.32%。在图 19 显示的挖掘质量对比中,随着最小支持度的不断变大,ECAOA 算法的挖掘质量逐渐提高。当最小支持度大于 6.8%时,ECAOA 算法的性能最优良,Apriori 算法的性能表现最差,DHP 算法的性能

最不稳定.

通过第 2 类实验,我们不难分析出:ECAOA 算法在绝大多数情况下,在运行时间、关联规则数目、准确率这 3 方面的性能都要稍微超过其他 3 种挖掘算法,在挖掘质量方面则明显优于其他 3 种挖掘算法.这主要是因为 ECAOA 算法充分结合了克隆选择算法和模拟退火算法的优点,在每次挖掘过程中都采取并行的方式进行搜索,同时在算法的执行过程,总是将优良的抗体在群体中的规模不断扩大,变异和交叉操作又不断地丰富了种群的多样性,并且将生存度最高的抗体通过克隆抑制操作直接保留到下一代,同时,种群更新操作将性能较差的抗体直接进行删除.ECAOA 算法在整个数据挖掘期间充分体现出人工免疫算法所具备的并行性和鲁棒性特征.Apriori 算法在大多数情况下的性能表现都不如 DHP 和 FP-Growth 算法,主要是因为其挖掘的规则存在大量冗余,并且算法需要多次扫描数据库,导致所消耗的时间较大,而 DHP 算法通过使用 TID,避免了反复读取事务数据库,FP-Growth 算法将频繁项集的数据库压缩成一棵频繁模式树的方式,能够有效提高算法的执行效率.

通过应用 ECAOA,CSA,GA,SA,Apriori,DHP 和 FP-Growth 算法对 census 和 chess 数据集从不同方面进行测试,实验结果显示了本文设计的 ECAOA 算法具有较快的执行时间和较强的搜索能力,能够有效地避免 CSA 算法可能存在的早熟收敛现象,同时具有良好的品质因数、挖掘质量和群体多样性,能够成功地应用于数据挖掘领域中.

7 结束语

针对传统的克隆选择算法可能存在的早熟收敛现象和缺少交叉操作问题,提出一种高效的克隆优化退火算法 ECAOA.它结合了模拟退火算法与免疫系统的克隆选择机制,并根据退火选择概率进行选择操作,同时,根据模拟退火的 Metropolis 接受准则设计了交叉操作,不仅能够有效保持全局搜索和局部搜索的平衡,而且可以有效提高算法的搜索效率,从而加快算法的收敛速度.在第 1 类数据挖掘实验测试中,ECAOA 算法在响应时间、执行时间、品质因数、误差性能和群体多样性这 5 个方面与 CSA,SA,GA 算法相比都表现出较好的性能.在第 2 类数据挖掘实验测试中,ECAOA 算法在绝大多数情况下,在运行时间、关联规则数目、准确率这 3 个方面的性能都要稍微超过其他 3 种挖掘算法,在挖掘质量方面,则明显优于其他 3 种挖掘算法,并解决了传统的克隆选择算法可能存在的早熟收敛问题.

References:

- [1] Mohammadi M, Raahemi B, Akbari A, Nassersharif B, Moeinzadeh H. Improving linear discriminant analysis with artificial immune system-based evolutionary algorithms. *Information Sciences*, 2012,189(4):219–232. [doi: 10.1016/j.ins.2011.11.032]
- [2] Naderi B, Khalili M, Tavakkoli-Moghaddam R. A hybrid artificial immune algorithm for a realistic variant of job shops to minimize the total completion time. *Computers & Industrial Engineering*, 2009,56(4):1494–1501. [doi: 10.1016/j.cie.2008.09.031]
- [3] Molla-Alizadeh-Zavardehi S, Hajiaghaei-Keshteli M, Tavakkoli-Moghaddam R. Solving a capacitated fixed-charge transportation problem by artificial immune and genetic algorithms with a Prüfer number representation. *Expert Systems with Applications*, 2011, 38(8):10462–10474. [doi: 10.1016/j.eswa.2011.02.093]
- [4] Aragón VS, Esquivel SC, Coello Coello CA. An immune algorithm with power redistribution for solving economic dispatch problems. *Information Sciences*, 2015,295(2):609–632. [doi: 10.1016/j.ins.2014.10.026]
- [5] Ulutas BH, Kulturel-Konak S. An artificial immune system based algorithm to solve unequal area facility layout problem. *Expert Systems with Applications*, 2012,39(5):5384–5395. [doi: 10.1016/j.eswa.2011.11.046]
- [6] Liu RC, Zhu BB, Bian RY, Ma YJ, Jiao LC. Dynamic local search based immune automatic clustering algorithm and its applications. *Applied Soft Computing*, 2015,27(2):250–268. [doi: 10.1016/j.asoc.2014.11.026]
- [7] Aydin I, Karakose M, Akin E. A multi-objective artificial immune algorithm for parameter optimization in support vector machine. *Applied Soft Computing*, 2011,11(1):120–129. [doi: 10.1016/j.asoc.2009.11.003]
- [8] Sakthivel VP, Bhuvaneshwari R, Subramanian S. Artificial immune system for parameter estimation of induction motor. *Expert Systems with Applications*, 2010,37(8):6109–6115. [doi: 10.1016/j.eswa.2010.02.034]

- [9] Turkoglu I, Didem Kaymaz E. A hybrid method based on artificial immune system and k -NN algorithm for better prediction of protein cellular localization sites. *Applied Soft Computing*, 2009,9(2):497–502. [doi: 10.1016/j.asoc.2008.07.003]
- [10] Hussain AJ, Fergus P, Al-Askar H, Al-Jumeily D. Dynamic neural network architecture inspired by the immune algorithm to predict preterm deliveries in pregnant women. *Neurocomputing*, 2015,151(3):963–974. [doi: 10.1016/j.neucom.2014.03.087]
- [11] Yu HM, Yao PJ. Combined genetic algorithm/simulated annealing algorithm for large-scale system energy integration. *Computers and Chemical Engineering*, 2000,8(24):2023–2035. [doi: 10.1016/S0098-1354(00)00601-3]
- [12] Ling J, Cao Y, Ying JH, Xu GX, Huang TX. An algorithm for Generating diverse antibody based on niche technology. *Acta Electronica Sinica*, 2003,31(8):1130–1133 (in Chinese with English abstract). [doi: 10.3321/j.issn:0372-2112.2003.08.003]
- [13] Xi GY, Yue JP, Zhou BX, Tang P. Application of an artificial immune algorithm on a statistical model of dam displacement. *Computers & Mathematics with Applications*, 2011,62(10):3980–3986. [doi: 10.1016/j.camwa.2011.09.057]
- [14] Mohammad AH, Zitar RA. Application of genetic optimized artificial immune system and neural networks in spam detection. *Applied Soft Computing*, 2011,11(4):3827–3845. [doi: 10.1016/j.asoc.2011.02.021]
- [15] Li SZ, Li MY, Pan YX. Genetic annealing algorithm and its convergence analysis. *Control Theory and Applications*, 2001,19(3):376–381.
- [16] Duan HB, Zhang XY, Xu CF. *Bio-Inspired Computing*. Beijing: Science Press, 2010. 98–105 (in Chinese).
- [17] Huang YR. *Intelligent Optimization Algorithm and Its Application*. Beijing: National Defense Industry Press, 2008. 198–210 (in Chinese).
- [18] Farzanyar Z, Kangavari M, Cerccone N. Max-FISM: Mining (recently) maximal frequent itemsets over data streams using the sliding window model. *Computers & Mathematics with Applications*, 2012,64(6):1706–1718. [doi: 10.1016/j.camwa.2012.01.045]
- [19] Yang BR, Li JH, Song W, Li X. KD (D&K): A new knowledge discovery process model for complex systems. *Acta Automatica Sinica*, 2007,33(2):151–155 (in Chinese with English abstract). [doi: 10.1360/aas-007-0151]

附中文参考文献:

- [12] 凌军,曹阳,尹建华,徐国雄,黄天锡.基于小生境技术的多样性抗体生成算法.电子学报,2003,31(8):1130–1133. [doi: 10.3321/j.issn:0372-2112.2003.08.003]
- [16] 段海滨,张祥银,徐春芳.仿生智能计算.北京:科学出版社,2010. 98–105.
- [17] 黄友锐.智能优化算法及其应用.北京:国防工业出版社,2008. 198–210.
- [19] 杨炳儒,李晋宏,宋威,李欣.面向复杂系统的知识发现过程模型 KD(D&K)及其应用.自动化学报,2007,33(2):151–155. [doi: 10.1360/aas-007-0151]



舒万能(1981—),男,湖北武汉人,博士,讲师,主要研究领域为智能计算,云计算.



丁立新(1967—),男,博士,教授,博士生导师,CCF 专业会员,主要研究领域为智能计算,智能信息处理.