

文章编号: 1001-0920(2012)02-0259-07

求解最小属性约简的病毒协同进化微粒群算法

陶新民¹, 王妍¹, 徐晶², 童智靖¹

(1. 哈尔滨工程大学信息与通信工程学院, 哈尔滨 150001; 2. 黑龙江省科技学院数力系, 哈尔滨 150027)

摘要: 提出一种基于病毒协同进化微粒群的最小属性约简算法. 在算法中, 进化在宿主与病毒种群之间协同进行, 通过满足约简分辨力不变条件的最优病毒种子复制操作产生病毒库, 病毒通过感染操作在宿主种群完成横向局部搜索, 以提高算法局部精确解搜索能力; 同时通过删减操作完成自我更新, 实现增加局部搜索范围的目的. 最后对 UCI 数据集进行属性约简实验, 结果表明该算法在搜索最小属性约简解方面优于其他进化算法, 同时收敛速度及寻优效率也有较大提高.

关键词: 属性约简; 微粒群算法; 病毒库; 协同进化

中图分类号: TP18

文献标识码: A

Minimum rough set attribute reduction algorithm based on virus-coordinative discrete particle swarm optimization

TAO Xin-min¹, WANG Yan¹, XU Jing², TONG Zhi-jing¹

(1. College of Information and Communication Engineering, Harbin Engineering University, Harbin 150001, China;

2. Department of Mathematics and Mechanics, Heilongjiang Institute of Science and Technology, Harbin 150027, China.

Correspondent: TAO Xin-min, E-mail: taixinmin@hrbeu.edu.cn)

Abstract: A minimum rough-set attribute reduction algorithm based on virus-coordinative discrete particle swarm optimization(VCDPSO) is presented. In the algorithm, evolutions of the virus swarm are performed in coordination with the particle swarm, and virus swarm keeps coordinative relations with the particle swarm by virus infection operations and best virus seed extraction operation in order to improve the ability of local search of discrete particle swarm optimization(DPSO). To enlarge the area of local search, the cut operator is introduced in the virus swarm's self-renewal process. A proper fitness function is defined and theoretical analysis and the experimental results on UCI dataset attribute reduction show that the proposed method has better performance than other evolution attribute reduction algorithms, and the searching efficiency and convergence rate for the global optimum are greatly improved as well.

Key words: attribute reduction; discrete particle swarm optimization; virus; coordinative evolution

1 引言

粗糙集理论(RST)是用来描述不完整性和不确定性问题的数学工具^[1], 近年来已被广泛用于机器学习、模式识别等方面. 在粗糙集的各种应用中, 属性约简是粗糙集理论的核心问题之一. Wong^[2]已经证明决策表的最小属性寻找是一个 NP 难题, 因此要设计一个有效的最小属性约简算法是相当困难的.

传统的粗糙集属性约简算法都是基于启发式搜索算法^[3], 尽管算法易于实现, 计算速度快, 但求出的解往往不一定是最小属性约简. 作为一类有效的

全局优化技术, 进化计算已被很多学者用于求解属性约简问题^[4-5], 并取得了相应的成果. 微粒群优化算法^[6-8]是近年来出现的一种新的群体智能优化算法, 由于其运算简单、易于实现且具有较强的全局收敛能力和鲁棒性等优势, 目前已广泛应用于函数优化、神经网络训练、TSP 问题、类覆盖问题等领域. 近年来, 微粒群算法在最小属性约简方面的应用也初见报道^[9-12], 如文献[9]提出的基于量子粒子群的最小属性约简算法; [10]提出的引入种子粒子自适应保护策略的属性约简算法; [11]提出的免疫粒子群的最小

收稿日期: 2010-09-14; 修回日期: 2010-11-16.

基金项目: 国家自然科学基金项目(61074076); 中国博士后科学基金项目(20090450119); 中国博士点新教师基金项目(20092304120017); 黑龙江省博士后基金项目(LBH-Z08227).

作者简介: 陶新民(1973-), 男, 副教授, 从事智能信号处理、智能计算等研究; 王妍(1986-), 女, 硕士, 从事智能优化算法的研究.

属性约简算法以及 [12] 提出的基于表现型共享多目标粒子群属性约简算法等. 然而, 由于离散微粒群算法自身在最优解处的发散特性导致算法局部搜索能力差, 最终使得上述算法在寻求最小属性约简解效率及收敛速度方面的能力大大降低.

受自然界病毒进化理论^[13]的启发, 本文提出一种基于病毒协同进化微粒群的最小粗糙集属性约简求解算法 (VCDPSO). 在算法进化过程中产生两个种群: 宿主种群及病毒种群, 两个种群之间通过最优病毒种子注射操作以及病毒感染操作协同进化, 利用宿主种群的全局进化和病毒种群的局部进化动态结合即可快速得到最小属性约简解. 将本文算法应用在 UCI 标准数据的最小属性约简求解问题中, 结果表明本文算法同其他算法相比具有更高的寻优效率及更快的收敛速度.

2 粗糙集最小属性约简问题描述

粗糙集理论是由 Pawlak 等人提出的用于描述不完整性和不确定性的数学理论, 该理论从新视角对知识的概念进行了定义, 并将知识看作是对论域的一种划分. 为叙述方便, 本文对粗糙集中的一些基本概念做以下简单介绍.

一个决策表信息系统 L 可表示为 $L = (U, R, V, f)$, 其中 $U = \{x_1, x_2, \dots, x_n\}$ 为有限非空对象集合, 称为论域; $R = C \cup U$ 为属性集合, 其中, C 和 U 分别为条件属性和决策属性集合; $V = \bigcup_{r \in R} V_r$ 为属性值集合; $f: U \times R \rightarrow V$ 为一个映射, 它指的是 U 中每个对象的属性值.

设 $B \subseteq C$, 既可表示为一个属性子集, 又可表示为一个不可分辨关系 $\text{Ind}(B)$. 不失一般性, 设决策属性 D 仅为 1 个, 其取值范围为 $1, 2, \dots, m$, 记 $U/\text{Ind}(D) = \{Y_1, Y_2, \dots, Y_m\}$, 其中 $Y_i = \{x \in U, f(x, D) = i\}$, $i = 1, 2, \dots, m$.

定义 1 设 $X \subseteq U$ 为论域一个子集, $B \subseteq C$ 为属性子集, X 关于 B 的一个下近似集为

$$\underline{B}X = \{[x]_B \subseteq X, x \in U\}, \quad (1)$$

其中 $[x]_B$ 表示 U 中所有与 x 在 $\text{Ind}(B)$ 关系下是等价的元素组成的集合.

定义 2 设 $B \subseteq C$, 对划分 $U/\text{Ind}(D)$ 的 B -近似精度为

$$\gamma_B = \sum_{i=1}^m \text{Card}(\underline{B}Y_i) / \text{Card}(U), \quad (2)$$

其中 $\text{Card}()$ 表示集合的基数.

定义 3 设 $B \subseteq C$, 若 $\gamma_B = \gamma_C$, 且不存在 $A \subset B$, 使得 $\gamma_A = \gamma_C$, 则称 B 为 C 的一个属性约简 (相对于决策属性 D). 条件 1 要求约简不能降低系统的区分

力, 即约简应该有与系统中全部条件属性相同的分辨力; 条件 2 要求在一个约简中不存在多余的属性. 所有 C 的属性约简的交集称为核, 表示为 $\text{Core}(C)$, 它是描述对象不可缺少的属性.

定义 4 设 $R = \{B_1, B_2, \dots, B_k\}$ 为条件属性 C 的所有属性约简集合, 称 S 为 C 的最小属性约简, 其中 S 满足 $\text{Card}(S) = \min_{i=1}^k \text{Card}(B_i)$.

由上述定义可知, 最小属性约简求解是一个带非线性约束条件的组合优化问题, 根据上述定义, 问题求解可描述如下:

$$\begin{aligned} & \min \text{Card}(B); \\ & \text{s.t. } B \subseteq C, \gamma_B = \gamma_C, \forall q \in B, \gamma_{B/\{q\}} < \gamma_B. \end{aligned} \quad (3)$$

属性约简求解的复杂度随着决策表的增大呈指数增长, 符合条件的属性约简非常多, 其中最小属性约简也通常不唯一, 因此计算最小属性约简成为一个 NP 难题.

3 求解最小属性约简的病毒协同进化微粒群算法

3.1 离散微粒群算法描述

二进制离散微粒群算法是为了解决组合优化问题而提出的^[9]. 设微粒群含 L 个微粒, 每个微粒相当于 N 维离散空间的一个活动点. 微粒 i 在时刻 t 的速度、位置、个体最好位置以及全局最好位置分别用 $v_i(t), x_i(t), x_i^{(p)}(t), x_i^{(g)}(t)$ 表示, 则微粒 i 速度和位置各维分量迭代公式如下^[11]:

$$v_{ij}(t+1) = wv_{ij}(t) + c_1r_{1j}(t)(x_{ij}^{(p)}(t) - x_{ij}(t)) + c_2r_{2j}(t)(x_{ij}^{(g)}(t) - x_{ij}(t)); \quad (4)$$

$$x_{ij}(t+1) = \begin{cases} 0, & \rho \geq \text{Sig}(v_{ij}(t+1)); \\ 1, & \rho < \text{Sig}(v_{ij}(t+1)). \end{cases} \quad (5)$$

其中: $i = 1, 2, \dots, L$; $j \in \{1, 2, \dots, N\}$ 为微粒编码中分量的维数; $r_{1j}(t)$ 和 $r_{2j}(t)$ 为 $(0,1)$ 上均匀分布的随机数; w, c_1 和 c_2 为权重及加速系数; $\rho \sim U[0,1]$ 为 $(0,1)$ 区间上均匀分布的随机变量; $\text{Sig}()$ 为 sigmoid 函数, 本文取 $\text{Sig}(x) = \frac{1}{1 + \exp(-x)}$ ^[9].

3.2 病毒协同进化微粒群模型描述

根据病毒进化理论可知^[13], 病毒是一种特有的具有感染功能的生物, 它能够通过获得一个宿主个体基因并感染给另一个体, 使得该个体的部分基因发生相应变化, 进而改变该个体的进化信息, 这种信息又通过进化传递给下一代, 从而加速了生物的进化速度. 1996 年, Kubota 提出一种基于病毒进化理论的遗传算法 VEGA (virus co-evolution genetic algorithm), 并成功地用于求解旅行商问题、调度问题、弹道规划问题

等, 显示了求解 NP-hard 优化问题的优越性. 另外, 文献 [13] 通过将病毒与粒子群结合提出了一个求解背包 NP 问题的病毒微粒群算法, 试验效果较好. 受上述病毒进化理论成功求解 NP-hard 问题的启发, 以及考虑到属性约简问题的特殊性, 本文提出了一种求解最小属性约简的病毒协同进化离散微粒群算法.

3.3 微粒编码表示以及适应值函数定义

对于粗糙集最小属性约简问题, 它是一个带约束条件的组合优化问题, 因此将粒子的编码变成 0-1 组合优化形式. 设 $\text{Card}(C) = N$, 则粒子的编码表示为

$$\mathbf{X} = \{x_1, x_2, \dots, x_N\} \in \{0, 1\}^N. \quad (6)$$

其中: 1 代表对应属性被选取, 0 表示未被选取.

适应值函数即为优化的目标函数, 好的适应值函数定义可以保证所求的最优解满足属性约简的约束条件, 因此这里定义适应值函数为

$$f(\mathbf{X}) = \begin{cases} 1/(N - m + \gamma_X), & \gamma_X < \gamma_C; \\ 1/(2N - m + \gamma_X), & \gamma_X = \gamma_C; \end{cases} \quad (7)$$

$$m = \sum_{i=1}^N x_i.$$

这样定义的目的是为了提升那些近似精度满足属性约简分辨率不变条件的粒子亲和度, 以防止由于约简维数较低而近似精度不满足属性约简条件的微粒成为最优, 进而增大算法所求的最优解满足属性约简约束条件的概率.

3.4 病毒编码及初始生成操作

根据病毒进化理论, 病毒种子库是根据待求问题的先验知识以及种群进化过程中较优个体总结得到的对最优解的一个估计. 特别地, 对于带条件约束的优化问题而言, 病毒种子代表着可行解个体. 由于以往利用进化算法求解属性约简问题时, 只考虑适应值函数^[4,9-12]的取值大小, 根据式 (3) 可知, 所求得的最优解不一定满足属性约简非线性约束的条件, 即求得的解不一定是属性约简. 因此考虑到属性约简问题的特殊性, 本文设其代表着一个近似精度满足属性约简分辨率不变条件的粒子集合, 以保证算法每一次求得的最优解都满足属性约简分辨能力不变的约束条件.

对于求解最小属性约简问题, 病毒种子定义为

$$V = \{v_1, v_2, \dots, v_N\} \in \{0, 1\}^N, \gamma_V = \gamma_C, \quad (8)$$

即满足属性约简的第 1 个条件: 不降低系统分辨能力的粒子作为病毒种子. 最优病毒种子是指其中满足

$$V^b = \min_{m=1}^{VS} \sum_{i=1}^N v_i^m \quad (9)$$

的病毒. VS 为病毒种子库的大小.

本文算法中病毒的编码采用与粒子等长的编码

方案, 编码中包含有通配符*, 除此之外, 其他的字符统称为有效字符, 也称粒子位. 由病毒体构成的集合称为病毒体集合, 记为 $\text{Pop}V$, 大小为 M . 病毒体生成是由最优病毒种子通过复制概率 P_{copy} 进行复制操作完成的, 如图 1 所示.

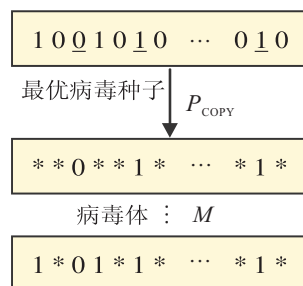


图 1 病毒的生成与复制更新操作

3.5 病毒感染操作

病毒感染操作是指利用病毒体的粒子位替换主粒子体中相应的粒子位信息, 从而产生新的主粒子个体, 如图 2 所示. 病毒体集合中的每个病毒体都以概率 P_{infect} 感染进化种群中的每个主粒子个体, 如果某主粒子个体感染后的适应值变优了, 则用感染后的主个体替换感染前的主个体. 因此, 病毒感染操作的本质在于, 用部分有效的 0-1 组合模式替换属性约简多个解对应的 0-1 组合模式, 如果替换后的解优于替换前的解, 则用替换后的解代替替换前的解.

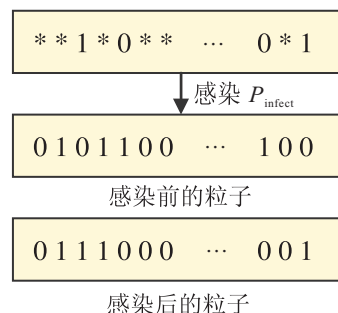


图 2 病毒感染操作

感染概率的值随着进化代数的增大而逐渐减少, 这样有助于进化粒子种群能在初始阶段进行大范围的搜索, 而在进化后期则进行小范围局部精确解的搜索, 具体公式设置如下:

$$P_{\text{infect}}^i = P_{\text{infect}}^{\text{max}} - (P_{\text{infect}}^{\text{max}} - P_{\text{infect}}^{\text{min}}) * i / \text{Iter} \quad (10)$$

其中: $P_{\text{infect}}^{\text{max}}$ 为最大感染概率, 取 0.9; $P_{\text{infect}}^{\text{min}}$ 为最小感染概率, 取 0.1; i 为当前迭代次数, Iter 为总迭代次数.

一个病毒个体可以感染若干个主粒子个体, 每个病毒个体都有一个适应值函数. 病毒个体的适应值是用它所感染的主个体感染前后的适应值的变化来表示, 即代表该病毒的感染效果. 假设病毒 i 所感染的主

粒子个体集合为 U , U 中主粒子个体 j 被感染前后的适应值分别表示为 $\text{fit}p_j$ 和 $\text{fit}p'_j$, 则病毒 i 的适应值为

$$\text{fit}V_i = \sum_{j \in U} (\text{fit}p_j - \text{fit}p'_j). \quad (11)$$

该病毒的适应值体现了病毒本身的粒子位信息对进化种群质量提升的效果. 本文属性约简采用的是最小化函数优化, 因此 $\text{fit}V_i$ 值越大, 说明该病毒的感染效果越好. 病毒的生存时间用生命力来表现, 第 $t+1$ 代的病毒个体 i 的生命力表示如下:

$$\text{life}_{i,t+1} = \gamma * \text{life}_{i,t} + \text{fit}V_i. \quad (12)$$

γ 为生命力递减率, 这里取 $\gamma = 0.3$.

3.6 病毒更新操作

病毒的生成与更新, 包括外部更新及自我更新, 外部更新是指当从进化种群中得到的最优病毒种子发生变化时, 利用复制操作实现病毒库的重新生成与更新; 自我更新是指病毒库本身根据进化的特征进行病毒的自我进化, 当某个病毒 i 的适应值小于零时, 则说明该病毒的感染能力在本代中降低, 因此通过对自身的裁剪操作使其便于完成更大局部空间的开采; 裁剪操作是指对病毒个体中的每个有效基因以概率 P_{cut} 进行通配符替换, 具体操作如图 3 所示. 如此操作可有效地细化该病毒个体的局部解空间的开采能力. 当病毒生命力的值小于零时, 说明该病毒的感染活力已经丧失, 需要利用最优病毒种子重新利用图 1 所示的复制操作实现该病毒体的外部更新.

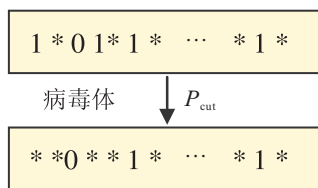


图 3 病毒裁剪操作

3.7 最优病毒种子注射操作

为保证算法每次求得的最优解都尽可能满足最小属性约简式(3)中分辨力不变的条件, 本文采取最优病毒种子注射机制, 即利用最优病毒种子代替进化种群中适应值最低的粒子的方式, 以确保每一次种群进化过程中都包含有最优病毒种子, 最终使得所求得的最优解满足属性约简条件. 对于带约束条件的最小属性约简优化问题, 这样做可以保证每一代进化种群中含有可行解的概率大大增加.

综上所述, 本文提出的算法是通过最优病毒种子对进化种群个体进行注射, 再通过进化种群中得到的满足病毒种子条件的最优个体通过复制操作对病毒体集合进行修正和补充完成自我更新, 使得进化种群与病毒体集合库之间建立联系, 协同进化, 相互作用.

前者让病毒体集合帮助进化种群更好地进行局部解空间的开采; 后者希望进化种群通过最优病毒种子的复制操作对其进行必要的改良. 同时进化种群与病毒库又根据自身的规则进行自我进化, 实现良性循环, 最终加快本文算法对最小属性约简优化问题的求解.

3.8 病毒协同微粒群求解最小属性约简算法描述

具体算法流程如图 4 所示.

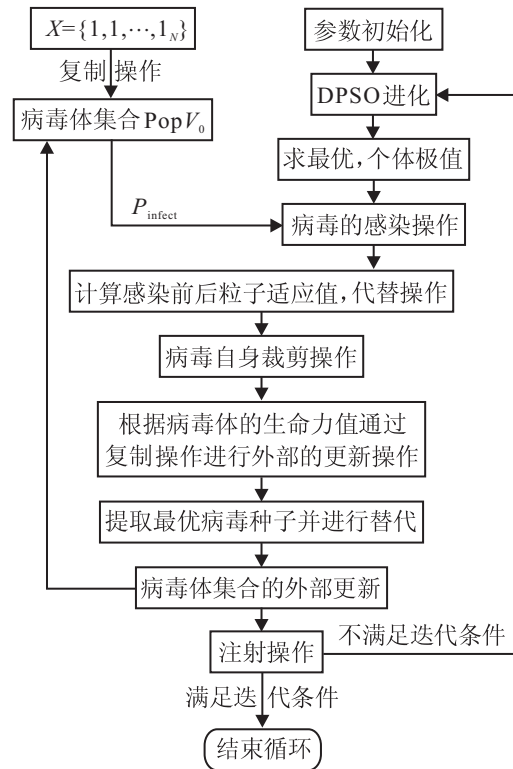


图 4 本文算法流程框架图

1) 算法参数初始化设定: $i = 0, P_{\text{copy}}, P_{\text{cut}}, \gamma, P_{\text{infect}}^{\text{max}}, P_{\text{infect}}^{\text{min}}, \text{Swarm}_i$ 微粒群集合, $\text{Pop}V_i$ 病毒体集合, 将 $X = \{1, 1, \dots, 1_N\}$ 的粒子作为初始最优病毒种子, 利用复制操作完成病毒体集合 $\text{Pop}V_0$ 的初始化.

2) 按照离散微粒群进化公式(4)进化.

3) 根据式(5)进行微粒状态的计算.

4) 计算当前最优解及每个微粒个体极值解.

采用以下操作完成病毒的进化:

1) 执行病毒的感染操作, $\text{Pop}V_i$ 中的每个病毒体以 P_{infect}^i 感染 Swarm_i 中的粒子. 记 Swarm_i 中被感染后的子群个体为 U , 计算感染前后粒子的适应值, 如果感染后的适应值更优, 则用感染后的粒子代替该粒子, 同时依式(11)计算该病毒体的适应值, 如果小于零, 则利用 P_{cut} 进行自身裁剪操作.

2) 根据式(12)计算病毒体的生命力, 如果小于零, 则通过复制操作进行外部的更新操作.

3) 提取最优病毒种子, 如果该最优病毒种子比当

前的最优病毒种子更优, 则进行替代并用新的种子以一定的复制概率 P_{copy} 完成病毒体集合的外部更新; 否则, 进行最优病毒种子的注射操作。

4) 循环直到满足终止条件。

4 对比实验及结果分析

4.1 试验设置

将新算法同基于免疫的数据约简算法(IM)^[4], 基于DPSO的数据约简算法(DPSO)^[9], 基于种子粒子DPSO的数据约简算法(SDPSO)^[10]以及基于多目标的DPSO的数据约简算法(MDPSO)^[12]进行比较, 实验数据取自UCI的实际数据, 其中前4个数据集规模较小, 后4个数据集规模较大, 具体设置如表1所示。本文算法(VCDPSO)的参数设置如下: 迭代次数为200, 粒子种群规模为30, $C_1 = C_2 = 1$, $V_{max} = 5$, 病毒体集合个数为6, 初始感染概率为0.7, 复制概率为0.75, 裁剪概率为0.7; 其他算法参数设置如下: 加速系数及惯性权重分别为 $C_1 = C_2 = 1$, $V_{max} = 4$, 种群规模为30, 算法终止条件为迭代次数200, IM算法的变异概率为0.1, 所有实验均在Matlab7.0下实现。为了消除随机影响, 取20次实验后的算法得到的最优解统计结果进行比较, 其中最小属性约简概率表示20次运算中获得预定决策表最小属性约简的次数。

表1 实验中所用的数据集及其特征

Dataset	Attribute	Class	Size
Tic-tac-toe	9	2	958
Votes	16	2	435
Lymphography	18	4	148
Mushroom	22	2	1 000
Soybean-small	35	4	47
Soybean-Large	35	19	306
Lung	56	3	32
DNA	57	3	1 000

4.2 求解最小属性约简性能对比实验

为了测试本文算法在处理数据集最小属性约简的优化性能, 将5种基于进化的属性约简算法应用在上述数据集中, 计算各个算法得到的最小属性约简概率以及约简属性数的最小值、最大值、平均值, 并显示在表2和表3中。

如表2所示, 通过对Tic-Tac-Toe数据集属性的约简可以看出, 基于IM以及传统DPSO的属性约简算法由于采用不适当的适应值函数定义, 使得其所求的最优化解与最小属性约简优化问题的解不等价, 因此无法保证所求的最优解本身是一个属性约简。如IM和DPSO算法所求的长度为7的属性集合, 虽然基数最小但解本身并不满足属性约简的条件, 即不是一个可行解。由于本文采用的适应值函数设计合理, 保证了算法所求的最优解至少满足属性约

表2 各种算法求解小型数据最小属性约简解的性能比较

Dataset	Algorithm	Max	Mean	Rate	Min
Tic-Tac-Toe	IM	7*	7*	0	7*
	DPSO	7*	7*	0	7*
	MDPSO	8	8	100	8
	SDPSO	8	8	100	8
	VCDPSO	8	8	100	8
Votes	IM	8	8	0	8
	DPSO	8	8	0	8
	MDPSO	12	10.43	10	0*
	SDPSO	11	9.35	75	9
	VCDPSO	10	9.2	80	9
Lymphography	IM	7	7	0	7
	DPSO	7	7	0	7
	MDPSO	7	6.94	5.56	6
	SDPSO	7	6.77	22.22	6
	VCDPSO	7	6.50	50	6
Mushroom	IM	3	2.65	5	1
	DPSO	3	2.45	0	2
	MDPSO	3	2.26	10.53	1
	SDPSO	2	1.05	94.74	1
	VCDPSO	1	1	100	1

表3 各种算法求解大型数据最小属性约简的解性能比较

Dataset	Algorithm	Max	Mean	Rate	Min
Soybean-Small	IM	8	7.1	0	7*
	DPSO	8	6.6	0	7*
	MDPSO	7	6.3	27	8
	SDPSO	4	2.3	75	8
	VCDPSO	2	2	100	8
Soybean-Large	IM	14	13.1	0	11
	DPSO	14	12.65	0	8
	MDPSO	14	12.70	13.41	0*
	SDPSO	12	9.65	55	9
	VCDPSO	11	9.5	60	9
Lung	IM	13	11	0	8
	DPSO	13	11.95	0	7
	MDPSO	13	12	10	6
	SDPSO	6	5.30	15	6
	VCDPSO	6	4.8	25	6
DNA	IM	12	11.37	0	10
	DPSO	12	11	0	2
	MDPSO	12	11.25	0	1
	SDPSO	8	8	0	1
	VCDPSO	7	7	100	1

简的条件, 且通过病毒感染操作的限制使算法获得最小属性约简结果的概率大大增加。对于Lymphography和Mushroom两个数据集而言, 由于其结构相对复杂且属性较多, 其他4种算法的属性约简性能有所下降。本文算法由于采用最优病毒种子的注射以及病毒横向感染机制, 保证了种群所求最优解的可行性, 最终使得算法所求的最优解满足最小属性约简条件的可能性大大提高。

由表3可知, 本文算法针对Soybean-Small,

Soybean-Large, Lung 和 DNA 四个大中型数据集属性约简的寻优概率分别为 100%, 60%, 25% 和 100%. 而基于 IM 和 DPSO 的属性约简算法对于这 4 个数据集没能找到最小属性约简结果, MDPSO 和 SDPSO 算法虽然在前 3 个数据集以一定概率找到了最小属性约简, 但对于 DNA 数据集而言没能在有限的迭代次数找到最小属性约简解. 从实验结果可以看出, 对于大中型属性集合数据集最小属性约简求解的性能, 本文算法无论从最大值、平均值以及求解的成功概率都远远大于其他进化算法, 尤其对于 Soybean-Small 和 DNA 数据集的优化, 所求最小属性约简的概率达到了 100%.

为更直观地显示基于 SDPSO, MDPSO 以及本文算法求解最小属性约简的收敛情况, 图 5 展示了 3 种 DPSO 算法对 Vote, Mushroom, Soybean-small 及 DNA 数据集求解的收敛效果. 从图 5 中的收敛结果可以看出, 本文建议的基于病毒协同离散微粒群优化的

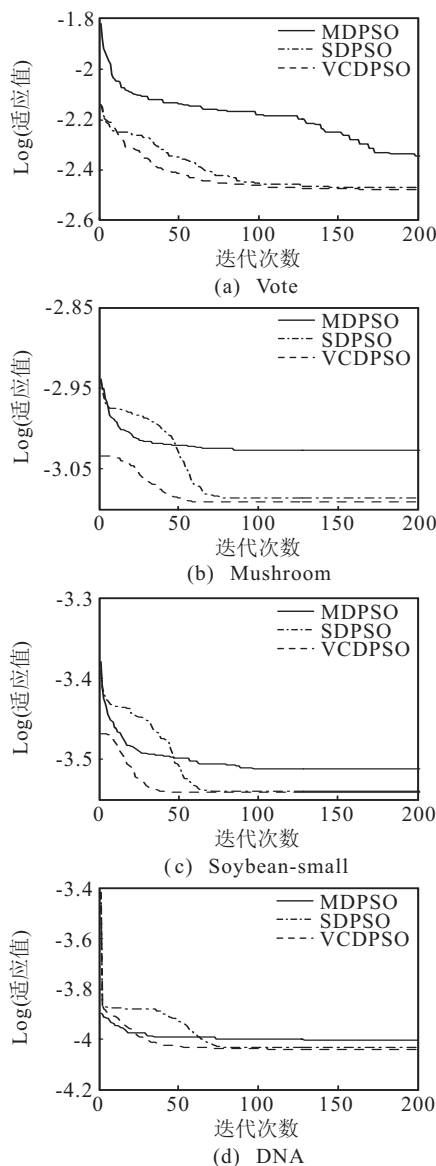


图 5 3 种进化算法求解属性约简收敛情况对比

最小属性约简算法收敛能力要优于其他两种基于 DPSO 的属性约简算法, 这是由于本文算法通过在满足属性约简分辨力不变条件的病毒种子中提取病毒体, 并对进化种群进行感染, 进而提高了 DPSO 算法在最优解附近的局部精确解搜索能力.

4.3 复制概率对算法寻优性能的影响

复制操作对于改善算法的局部精确解搜索能力起着十分重要的作用, 因此为考察本文算法中复制概率对寻求最小属性约简性能的影响, 取 30 个初始种群, 将复制概率设置为 [0.1, 1] 范围, 步长为 0.1, 试验数据为 Lung, Votes 两个数据集, 算法的其他设置同上, 实验结果如图 6 所示. 从实验结果可以看出, 本文算法寻求最优解的概率随着复制概率的增大呈上升趋势, 这是由于复制概率的增加, 病毒库中的病毒汲取最优病毒种子的信息增多, 因此在最优病毒种子附近的局部搜索功能更加精确, 这恰恰弥补了 DPSO 算法在最优解附近局部搜索能力不足的缺陷. 如果最优解附近未能发现更优结果, 则可以利用删减操作进行更大范围的搜索. 通过实验可以看出, 对于本文算法而言, 复制操作应该设置的大一些, 建议在 [0.75, 0.9].

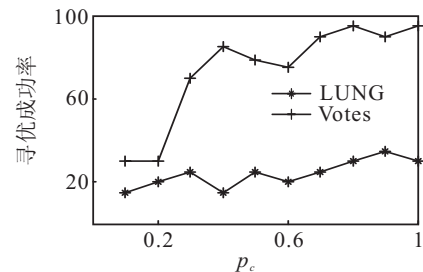


图 6 最优解寻优能力随复制概率变化关系

5 结论

本文提出了一种基于病毒协同微粒群进化的最小属性约简问题求解算法. 该算法通过产生的病毒种群和宿主种群的协同进化来完成对解空间的全局探索和局部开采. 将算法应用在 UCI 数据集的属性约简中, 实验结果表明本文算法能够以较高概率寻求到最小属性约简解, 稳定性强, 并且算法的收敛速度和效率较其他算法也有明显提高. 需要指出的是, 为了简化, 本文算法没有应用约简核, 如果应用约简核, 可将求解后的约简核作为最优病毒种子, 进行相应的最小属性约简求解.

参考文献(References)

- [1] 苗夺谦. 粗糙集理论中概念与运算的信息表示[J]. 软件学报, 1999, 10(2): 113-116.
(Miao D Q. An information representation of the concepts and operations in rough sets theory[J]. J of Software, 1999, 10(2): 113-116.)

- [2] Wong S K M, Ziarko W. On optimal decision rules in decision tables[J]. Bulletin of Polish Academy of Science. 1985, 33(11): 693-696.
- [3] 叶东毅. Jelonek 属性约简算法的一个改进[J]. 电子学报, 2000, 28(12): 81-83.
(Ye D Y. An improvement to Jelonek's attribute reduction algorithm[J]. Acta Electronica Sinica, 2000, 28(12): 81-83.)
- [4] 徐雪松, 章兢, 贺庆. 基于疫苗提取及免疫优化的粗糙集属性约简[J]. 控制与决策, 2008, 23(5): 497-502.
(Xu X S, Zhang J, He Q. Attribute reduction method of rough sets based on bacterin extraction and immune optimum[J]. Control and Decision, 2008, 23(5): 497-502.)
- [5] Poli R, Kennedy J. Particle swarm optimization: An overview[J]. Swarm Intell, 2007, 1(1): 33-57.
- [6] 陶新民, 徐晶. 改进的多种群协同进化微粒群优化算法[J]. 控制与决策, 2009, 24(9): 1406-1411.
(Tao X M, Xu J. Multi-species cooperative particle swarm optimization algorithm[J]. Control and Decision, 2009, 24(9): 1406-1411.)
- [7] 陶新民, 徐晶. 一种协调勘探和开采能力的粒子群算法[J]. 控制理论与应用, 2010, 27(5): 636-640.
(Tao X M, Xu J. Particle swarm algorithm coordinating the exploration and the exploitation[J]. Control Theory & Applications, 2010, 27(5): 636-640.)
- [8] Kennedy J, Eberhart R C. A discrete binary version of the particle swarm algorithm[C]. Proc of the World Multiconference on Systemics, Cybernetics and Informatics. Orlando, 1997: 4104-4109.
- [9] 王加阳, 谢颖. 基于量子粒子群优化的最小属性约简算法[J]. 计算机工程, 2009, 35(12): 48-150.
(Wang J Y, Xie Y. Minimal attribute reduction algorithm based on quantum particle swarm optimization[J]. Computer Engineering, 2009, 35(12): 148-150.)
- [10] 叶东毅. 基于二进制粒子群优化的一个最小属性约简算法[J]. 模式识别与人工智能, 2007, 20(3): 295-230.
(Ye D Y. Minimum attribute reduction algorithm based on binary particle swarm optimization[J]. PR&AI, 2007, 20(3): 295-300.)
- [11] 廖建坤. 基于免疫粒子群优化的最小属性约简算法[J]. 计算机应用, 2007, 27(3): 550-552.
(Liao J K. Minimal attribute reduction algorithm based on particle swarm optimization with immunity[J]. Computer Applications, 2007, 27(3): 550-552.)
- [12] 杨晓燕, 陈国龙. 基于粒子群优化的最小属性约简算法[J]. 福州大学学报: 自然科学版, 2010, 38(2): 193-197.
(Yang X Y, Chen G L. Minimum attribute reduction algorithm based on particle swarm optimization[J]. J of Fuzhou University: Natural Science, 2010, 38(2): 193-197.)
- [13] 高芳, 崔刚. 求解背包问题的病毒协同进化粒子群算法[J]. 哈尔滨工业大学学报, 2009, 41(6): 103-107.
(Gao F, Cui G. Virus evolutionary particle swarm optimization algorithm for knapsack problem[J]. J of Harbin Institute of Technology, 2009, 41(6): 103-107.)

~~~~~

(上接第258页)

- [6] Ge S S, Wang C. Direct adaptive NN control of a class of nonlinear systems[J]. IEEE Trans on Neural Networks, 2002, 13(1): 214-221.
- [7] Li Y H, Sheng Q, Zhuang X Y, et al. Robust and adaptive backstepping control for nonlinear systems using RBF neural networks[J]. IEEE Trans on Neural Networks, 2004, 15(3): 693-701.
- [8] Zhang T, Ge S S, Hang C C. Stable adaptive control for a class of nonlinear systems using a modified Lyapunov function[J]. IEEE Trans on Automatic Control, 2000, 45(1): 129-132.
- [9] Zhang T P, Ge S S, Hang C C. Adaptive neural network control for strict-feedback nonlinear systems using backstepping design[J]. Automatica, 2000, 36(12): 1835-1846.
- [10] Nussbaum R D. Some remarks on the conjecture in parameter adaptive control[J]. System and Control Letters, 1983, 3(3): 243-246.
- [11] Ge S S, Wang J. Robust adaptive neural control for a class of perturbed strict feedback nonlinear systems[J]. IEEE Trans on Neural Networks, 2002, 13(6): 1409-1419.
- [12] Ge S S, Fan H, Lee T H. Adaptive neural control of nonlinear time-delay systems with unknown virtual control coefficients[J]. IEEE Trans on Systems, Man and Cybernetics-Part B, 2004, 34(1): 499-516.
- [13] 胡剑波, 庄开宇. 高级变结构控制理论及应用[M]. 西安: 西北工业大学出版社, 2008: 19-24.  
(Hu J B, Zhuang K Y. Advanced variable structure control theory and applications[M]. Xi'an: Northwestern Polytechnical University Press, 2008: 19-24.)
- [14] Sanner R M, Slotine J E. Gaussian networks for direct adaptive control[J]. IEEE Trans on Neural Networks, 1992, 3(6): 837-863.