最小属性约简问题的一个有效的组合人工蜂群算法

叶东毅,陈昭炯
(福州大学数学与计算科学学院,福建福州 350108)

摘 要: 粗糙集理论中的最小属性约简(MAR)问题是一个 NP-难的非线性约束组合优化问题.本文提出一个新的求解 MAR 问题的组合蜂群算法,其中,引领蜂、跟随蜂和侦察蜂采用基于变异运算的搜索模式,在邻域候选蜜源的生成中引入与属性子集相关的两个度量,并且跟随蜂采用与引领蜂不同的局部搜索策略以提高搜索多样性.此外,在本文算法中,角色分工不同的蜂群以不同的方式利用迄今最好蜜源的信息进行搜索.在若干 UCI 数据集上的实验及其统计检验结果表明,本文算法在求解质量上优于其他的元启发式属性约简算法,因而可有效地应用于最小属性约简问题的求解.

关键词: 组合人工蜂群算法;最小属性约简;粗糙集;元启发式方法;局部搜索模式
中图分类号: TP181 **文献标识码:** A **文章编号:** 0372-2112 (2015)05-1014-07
电子学报 URL: http://www.ejournal.org.cn **DOI:** 10.3969/j.issn.0372-2112.2015.05.027

An Efficient Combinatorial Artificial Bee Colony Algorithm for Solving Minimum Attribute Reduction Problem

YE Dong-yi, CHEN Zhao-jiong
(*College of Mathematics and Computer Science, Fuzhou University, Fuzhou, Fujian* 350108, *China*)

Abstract: Minimum attribute reduction (MAR) problem in the context of rough set theory is an NP-hard nonlinearly constrained combinatorial (binary) optimization problem. In this paper, a new combinatorial artificial bee colony (ABC) algorithm is presented for solving the MAR problem. Mutation operation based search schemes are introduced for employed bees, onlooker bees and scout bees. Two different metrics related to attribute subsets are used to generate candidate neighboring food sources. Different local search strategies between an employed bee and its recruited onlooker bees allow for a more diversified neighboring search around a current food source. Moreover, the information of the so-far best solution is exploited in various ways by employed bees, onlookers and scouts, respectively. Performance comparisons with existing best performing meta-heuristic approaches for the MAR problem were carried out on a number of UCI data sets. In addition, a standard statistical *t*-test is used for evaluation purpose. The experimental results show that our combinatorial ABC approach compares favorably with all the other approaches in terms of solution quality. The proposed combinatorial ABC algorithm is thus efficient and well suited for solving the MAR problem.

Key words: combinatorial artificial bee colony algorithrn; minimum attribute reduction; rough sets; meta-heuristic approach; local search schemes

1 引言

作为特征选择的一个有效方法,粗糙集理论中的属性约简及其算法得到了广泛的研究[1~6].为了获得最精简的特征集,需要计算决策表的最小属性约简.因 z 而,最小属性约简问题(简称 MAR 问题)被证明是一个 NP-难的组合优化问题[7],现有的贪心型属性约简算法一般难以找到最小属性约简.因而,人们寻求基于元启发式方法的最小属性约简算法,例如,应用禁忌算法以及遗传算法、粒子群算法、蚁群算法等群体智能优化算法求解 MAR 问题[8~10].这些元启发式约简算法取得了较好的效果,但在实际计算中获得最小属性约简的概率依旧偏低,求解质量有待进一步提高[11].因此,有必要研究新的并且更有效的元启发式属性约简算法.

Karaboga 等人提出的用于求解连续优化问题的人工蜂群算法(简称 ABC 算法)是近年来涌现的一个新的基于群体搜索的元启发式算法[12].该算法模拟蜂群的蜜源搜索和采蜜行为,其中,蜂群由引领蜂、跟随蜂和侦

收稿日期:2013-10-08;修回日期:2014-03-12;责任编辑:赵克
基金项目:国家自然科学基金(No.71231003)

察蜂 3 种角色分工不同的蜂群构成,利用摇摆舞进行蜂群之间的信息沟通和招募.引领蜂和跟随蜂以采蜜和局部蜜源搜索为主,侦察蜂则主要负责全局搜索,以探测新的蜜源.研究表明[13,14],ABC 算法在求解连续优化问题方面往往可以取得优于遗传算法、粒子群优化、蚁群算法等其他群体智能算法的结果,提高了问题的求解质量.

鉴于连续型 ABC 算法的优点,本文提出一个求解 MAR 问题的组合人工蜂群算法.该算法针对属性约简问题的组合特性,在保持连续型 ABC 算法基本框架的基础上,引入若干适应性修改策略和改进措施,取得了良好的改进效果.实验结果和统计检验表明本文的算法可以比现有的一些元启发式属性约简算法获得更好的求解质量.

2　决策表最小属性约简问题

一个决策表[1]是一个四元组 $L=\{U,A,V,f\}$,其中 $U=\{x_1,x_2,\cdots,x_n\}$ 是由有限个对象组成的论域,$A=C\cup D$ 是属性集合,其中 C 是条件属性集合,D 是决策属性集合,$C\cap D=\varnothing$,V 是属性的值域,$f:U\times A\to V$ 是对象属性的赋值函数,对于 $P\subseteq A$,$IND(P)$ 表示由 P 导出的不可分辨关系,即

$$IND(P)=\{(x,y)\in U\times U:f(x,a)=f(y,a),\forall a\in P\}$$

不失一般性,假设 D 只包含一个决策属性,该属性取 $k(>1)$ 个不同的值,则商集 $U/IND(D)=\{Y_1,Y_2,\cdots,Y_k\}$,其中每个 Y_i 对应一个决策类.设 $X\subseteq U$,$R\subseteq C$,则称 $\underline{RX}=\{x\in U:[x]_R\subseteq X\}$ 为 X 的 R-下近似,其中 $[x]_R=\{y\in U:(x,y)\in IND(R)\}$;$\gamma_R=\sum_{i=1}^{k}\frac{|\underline{RY_i}|}{|U|}$ 称作 R 关于决策属性 D 的分类精度,其中 $|\cdot|$ 表示集合的基数.

定义 1[1]　给定 $R\subseteq C$.如果 $\gamma_R=\gamma_C$,则称 R 是一个独立属性子集;如果对 $\forall a\in R$,$\gamma_{R\setminus\{a\}}<\gamma_R$,则称 R 为 C 的一个属性约简.

一个决策表可以有多个属性约简,包含属性个数最少的属性约简称为最小属性约简.假设 $C=\{a_1,a_2,\cdots,a_m\}$,$\{0,1\}^m$ 表示 m 维布尔空间.定义映射 $\xi:\{0,1\}^m\to 2^C$ 如下:

$$\forall\boldsymbol{x}=(x_1,x_2,\cdots,x_m)^{\mathrm{T}}\in\{0,1\}^m,$$
$$x_i=1\Leftrightarrow a_i\in\xi(\boldsymbol{x}),a_i\in C,i=1,\cdots,m$$

最小属性约简问题可表示成如下的约束组合优化问题:

$$\max\frac{m-S(\boldsymbol{x})}{m}$$
$$\text{s.t.}\begin{cases}\boldsymbol{x}\in\{0,1\}^m\\ \gamma_{\xi(\boldsymbol{x})}=\gamma_C\\ \forall q\in\xi(\boldsymbol{x}),\gamma_{\xi(\boldsymbol{x})\setminus\{q\}}<\gamma_{\xi(\boldsymbol{x})}\end{cases}\tag{1}$$

其中,$0\leqslant S(\boldsymbol{x})=\sum_{i=1}^{m}x_i\leqslant m$.

由映射 ξ 定义可知,对于 $\boldsymbol{x}\in\{0,1\}^m$,如果 $\boldsymbol{x}$ 是式(1)的一个可行解,则 $\xi(\boldsymbol{x})$ 为 C 的一个属性约简;如果 $\boldsymbol{x}$ 还是式(1)的一个最优解,则 $\xi(\boldsymbol{x})$ 为 C 的一个最小属性约简.

通常将式(1)问题转化为如下形式的无约束组合优化问题进行求解[9]:

$$\max_{\boldsymbol{x}\in\{0,1\}^m}fit(\boldsymbol{x})\tag{2}$$

其中 $fit(\boldsymbol{x})$ 是适应值函数.最常用的一种适应值函数如下所示[9~11]:

$$fit(\boldsymbol{x})=\alpha\frac{m-S(\boldsymbol{x})}{m}+\beta\gamma_{\xi(\boldsymbol{x})}\tag{3}$$

其中 $\alpha>0,\beta>0$ 是权值.

本文也采用上述这种转化为最大化适应值函数的方法,其中适应值函数如式(3)所示,权值分别取为 $\alpha=0.1,\beta=0.9$.

3　求解 MAR 问题的组合 ABC 算法

3.1　标准连续型 ABC 算法及其相关工作

为便于描述本文提出的算法,首先简要介绍标准连续型 ABC 算法的基本原理和步骤,对其详细的分析参见文献[12].

在标准连续型 ABC 算法中,蜂群由 3 组分工不同的蜜蜂构成,分别是引领蜂、跟随蜂和侦察蜂.引领蜂主要负责采蜜并通过摇摆舞招募跟随蜂与其一起进行采蜜工作,招募的跟随蜂数量取决于蜜源花蜜的质量;如果引领蜂完成采蜜工作,则放弃对应的蜜源,转变角色成为一只侦察蜂;侦察蜂通过随机搜索发现新的蜜源,一旦发现新的蜜源,则转换角色成为引领蜂.在标准 ABC 算法中,一次迭代最多放弃一个蜜源.对于一个 D 维的优化问题,蜜源位置对应于优化问题的候选解,蜜源花蜜的质量则用一个与问题目标函数相关的适应值函数来度量.

下面介绍标准连续型 ABC 算法的基本步骤.在初始化阶段,算法随机生成优化问题可行解空间中的 SN 个候选解(即蜜源位置)$\boldsymbol{x}_i(i=1,\cdots,SN)$,其中 SN 为群体的规模.一个蜜源对应一只引领蜂,因而引领蜂的数量为 SN,跟随蜂的数量也为 SN.蜜源 $\boldsymbol{x}_i$ 吸引(即对应的引领蜂招募)跟随蜂数量的比例由下式决定,并依照赌轮原则计算相应的跟随蜂数量:

$$p_i=\frac{fit(\boldsymbol{x}_i)}{\sum_{k=1}^{SN}fit(\boldsymbol{x}_k)}\tag{4}$$

其中 $fit(\cdot)$ 为适应值函数.故蜜源对应的适应值越大,其吸引的跟随蜂数量就越多.

初始化之后,在每一次迭代中,引领蜂在其对应的蜜源 $\boldsymbol{x}_i$ 附近寻找是否存在具有更好质量(更高适应值)的蜜源,这相当于算法在 $\boldsymbol{x}_i$ 附近进行局部搜索,以寻求更好的可行解.候选蜜源 $\boldsymbol{y}_i$ 的生成方式如下:

$$y_{ij} = x_{ij} + \varphi_{ij}(x_{ij} - x_{kj}) \tag{5}$$

其中 $k \in \{1,\cdots,SN\} \setminus \{i\}$ 和 $j \in \{1,\cdots,D\}$ 分别是随机选择的下标,$\varphi_{ij} \in [-1,1]$ 是一个随机数.注意到,$\boldsymbol{y}_i$ 和 $\boldsymbol{x}_i$ 之间只有一个分量发生改变,因此,式(5)是一个单维度的局部搜索模式.跟随蜂也采用与招募它们的引领蜂相同的方式(即式(5))探查 $\boldsymbol{x}_i$ 附近的候选蜜源.如果 $fit(\boldsymbol{y}_i) > fit(\boldsymbol{x}_i)$,则 $\boldsymbol{y}_i$ 取代 $\boldsymbol{x}_i$ 成为新的蜜源;否则继续按照式(5)生成新的候选解 $\boldsymbol{y}_i$.这样的过程最多经过 $limit(>1)$ 次后,如果依旧无法找到新的更好的蜜源,则放弃当前蜜源 $\boldsymbol{x}_i$,对应的引领蜂转变为侦察蜂,并随机生成优化问题可行解空间的一个向量作为新的蜜源,之后该侦察蜂转变为引领蜂,进行新一轮如上所述的采蜜(迭代)过程,直至达到预定的最大迭代次数(记为 MCN)为止,算法输出蜂群中适应值最高的蜜源位置(可行解)作为问题的解.

关于标准连续型 ABC 算法的改进方法已有多篇文献报道[15,16],其中比较典型的一种改进措施是受粒子群算法启发,将群体最好解的信息引入到候选蜜源的生成中,即 $\boldsymbol{y}_i$ 的生成方式(5)修改为:

$$y_{ij} = x_{ij} + \varphi_{ij}(x_{ij} - x_{kj}) + \alpha_{ij}(x_{gj} - x_{kj}) \tag{6}$$

其中 $\boldsymbol{x}_g$ 为迄今蜂群找到的最好蜜源.本文提出的组合 ABC 算法也将借鉴上述这些改进方法的思想.

3.2 本文提出的组合 ABC 算法

MAR 问题是一个组合优化问题,如引言中所述,连续型 ABC 算法不能直接应用于它的求解;现有的求解特定组合优化问题的离散型 ABC 算法(如文献[17,18]等),由于其设计与求解问题密切相关,也不适合用于求解 MAR 问题.因此需要针对 MAR 问题的特点设计新的组合 ABC 算法.

为了便于描述本文提出的组合 ABC 算法(简记为 CABC 算法)以及对照差异,我们主要介绍在标准连续型 ABC 算法的框架下,本文算法做了哪些主要的修改和变化.首先引入一些必要的记号和度量.设 $\boldsymbol{p},\boldsymbol{q} \in \{0,1\}^m$ 为两个布尔向量,$dist_{\mathrm{H}}(\boldsymbol{p},\boldsymbol{q})$ 表示 $\boldsymbol{p}$ 和 $\boldsymbol{q}$ 之间的海明距离,即

$$dist_{\mathrm{H}}(\boldsymbol{p},\boldsymbol{q}) = \sum_{j=1}^{m} |p_j - q_j| \tag{7}$$

$I_{\mathrm{H}}(\boldsymbol{p},\boldsymbol{q})$ 表示 $\boldsymbol{p}$ 和 $\boldsymbol{q}$ 之间对应分量不相同的下标集合,即

$$I_{\mathrm{H}}(\boldsymbol{p},\boldsymbol{q}) = \{i : p_i \neq q_i\} \tag{8}$$

$dif(\boldsymbol{p},\boldsymbol{q})$ 表示 $\boldsymbol{p}$ 和 $\boldsymbol{q}$ 中各自包含分量为 1 的数量的差别,即

$$dif(\boldsymbol{p},\boldsymbol{q}) = \sum_{j=1}^{m} (p_j - q_j) \tag{9}$$

显然,$dif(\boldsymbol{p},\boldsymbol{q}) \leqslant dist_{\mathrm{H}}(\boldsymbol{p},\boldsymbol{q})$.如果 $dif(\boldsymbol{p},\boldsymbol{q}) = 0$,则对应的属性子集 $\xi(\boldsymbol{p})$ 和 $\xi(\boldsymbol{q})$ 包含相同数量的条件属性;否则,$\xi(\boldsymbol{p})$ 比 $\xi(\boldsymbol{q})$ 包含更多的(当 $dif(\boldsymbol{p},\boldsymbol{q}) > 0$ 时)或更少的(当 $dif(\boldsymbol{p},\boldsymbol{q}) < 0$ 时)条件属性.对于给定的一个正整数 k,记

$$N(\boldsymbol{p},k) = \{\boldsymbol{x} \in \{0,1\}^m : dist_{\mathrm{H}}(\boldsymbol{p},\boldsymbol{x}) \leqslant k\} \tag{10}$$

$N(\boldsymbol{p},k)$ 称作 $\boldsymbol{p}$ 的 k-邻域.

下面分别介绍 CABC 算法中引领蜂、跟随蜂和侦察蜂的搜索方法.显然,类似式(5)那样的单维度局部搜索公式对于组合优化问题是不实用的,因为那样的话,引领蜂及其跟随蜂只能搜索当前蜜源 1-邻域内的候选解,降低了搜索的效率.本文采用多维度更新策略,其中引入了引领蜂和跟随蜂飞行速度的概念.假设 $\boldsymbol{x}_i(t)$ $(1 \leqslant i \leqslant SN)$ 是算法在第 t 次迭代($t < MCN$)时的一个蜜源,定义其对应的引领蜂的飞行速度 $v_i \in R$ 如下:

$$v_i = r_t\varphi_1 dif(\boldsymbol{x}_i(t),\boldsymbol{x}_k(t)) + \varphi_2 dif(\boldsymbol{x}_g,\boldsymbol{x}_i(t)) \tag{11}$$

其中,$r_t = \dfrac{MCN - t}{MCN}$,$\boldsymbol{x}_g$ 是至今蜂群找到的适应值最高(最好)的蜜源,$\varphi_1 \in [-1,1]$ 和 $\varphi_2 \in [0,1]$ 是两个随机数,$k \in \{1,\cdots,SN\} \setminus \{i\}$ 是一个随机选择的下标.随着迭代的深入,v_i 的值主要取决于 $dif(\boldsymbol{x}_g,\boldsymbol{x}_i(t))$ 的大小.速度 v_i 用于确定局部搜索的范围.记

$$D_i = \begin{cases} 1, & v_i < 0 \\ \lfloor m/3 \rfloor + 1, & v_i \geq m/3 \\ \lfloor v_i \rfloor + 1, & \text{其他} \end{cases} \tag{12}$$

其中 $\lfloor\cdot\rfloor$ 是地板函数,即 $\lfloor x \rfloor$ 表示小于 x 的最大整数.令 $h_i = dist_{\mathrm{H}}(\boldsymbol{x}_i(t),\boldsymbol{x}_g)$,则候选蜜源 $\boldsymbol{y}_i$ 的生成方式如下:

(1)如果 $D_i \leqslant h_i$,从 $I_{\mathrm{H}}(\boldsymbol{x}_i(t),\boldsymbol{x}_g)$ 中随机选择 D_i 个下标构成下标集 VI_i,令

$$y_{ij} = \begin{cases} \neg\, x_{ij}(t), & j \in VI_i \\ x_{ij}(t), & \text{其他} \end{cases} \tag{13}$$

其中 $\neg$ 表示二进制取补运算.

(2)如果 $D_i > h_i$,则从 $I_{\mathrm{H}}(\boldsymbol{x}_i(t),\boldsymbol{x}_g)$ 之外随机选择 $(D_i - h_i)$ 个下标构成下标集 CI_i,令

$$y_{ij} = \begin{cases} \neg\, x_{ij}(t), & j \in CI_i \\ x_{ij}(t), & \text{其他} \end{cases} \tag{14}$$

由上可知,候选蜜源 $\boldsymbol{y}_i \in N(\boldsymbol{x}_i(t),D_i)$,其中 $1 \leqslant D_i \leqslant \lfloor m/3 \rfloor + 1$.

在标准连续型 ABC 算法中,跟随蜂采用和其追随的引领蜂一样的邻域局部搜索策略.本文采用不同的策略,其不同之处在于跟随蜂的飞行速度由下式给出:

$$v_i = r_t\varphi_1 dif(\boldsymbol{x}_i(t),\boldsymbol{x}_k(t)) + \varphi_2 dif(\boldsymbol{x}_g,\boldsymbol{x}_k(t)) \tag{15}$$

与式(11)相比,第二项 $dif(\boldsymbol{x}_g, \boldsymbol{x}_i(t))$换成了随机性更强的 $dif(\boldsymbol{x}_g, \boldsymbol{x}_k(t))$.这种改变有助于提高当前蜜源 $\boldsymbol{x}_i(t)$附近局部搜索的多样性.除了飞行速度按照式(15)计算外,后续的候选蜜源生成方式和其追随的引领蜂一样(参见式(12)~(14)).

侦察蜂的搜索策略与标准连续型 ABC 算法随机生成一个新蜜源的做法不同,它采用对放弃的蜜源位置进行适当变异的方式确定一个新的蜜源,其中也考虑了当前最好蜜源的信息.具体地说,假设当前蜜源 $\boldsymbol{x}_i(t)$被放弃,则对应的引领蜂转变角色成为侦察蜂,它发现的新蜜源位置 $\boldsymbol{Nx}_i \in \{0,1\}^m$ 如下所示:

$$Nx_{ij} = \begin{cases} x_{ij}(t), & j \in J_1 \\ x_{gj}, & j \in J_2 \\ x_{ij}(t), & \text{其他} \end{cases} \tag{16}$$

其中 $J_1, J_2 \subset I = \{1,2,\cdots,m\}$是两个随机选取的不相交的下标集,$|J_1| = K_1$,$|J_2| = K_2 \leqslant K_1$,而且 $K_1 + K_2 \geq \lfloor m/3 \rfloor + 1$.令 $\boldsymbol{x}_i(t+1) = \boldsymbol{Nx}_i$,这只侦察峰又转换为引领蜂,并在下一次迭代时在新的蜜源 $\boldsymbol{x}_i(t+1)$采蜜和搜索.

在 CABC 算法中,采用如下的精英保存策略:如果被放弃的蜜源恰好对应的是当前最好的蜜源,则除非替换它的新蜜源(如式(16)所定义)有更好的适应值,否则该遗弃的蜜源依然作为至今最好的解 $\boldsymbol{x}_g$ 加以保存.

以上是 CABC 算法的基本步骤和搜索策略.算法参数值的设定将在下一节中给出.

4　实验与结果分析

为了测试 CABC 算法的求解效果,将 CABC 算法与 4 个具有代表性的元启发式属性约简算法进行比较,它们分别是基于禁忌搜索的 TSAR 算法[8],基于遗传算法的 GAAR 算法[9],基于粒子群优化的 PSOAR 算法[10]和基于蚁群算法的 AntAR 算法[9].

算法在一台 3.3GHz CPU 和 2GB 内存的 PC 机上运行.从 UCI 机器学习数据库中选择 14 个属性约简算法常用的离散型数据集用于算法测试,数据集的基本信息如表 1 所示,其中 D-ID 为数据集的编号,n 和 m 分别是数据集对象的个数和条件属性的个数,*Ncls* 是决策类的个数.

表 2 给出 5 个算法的参数设置情况,其中,*SN* 表示群体的规模,*MCN* 是算法最大迭代次数,*V*max 是 PSOAR 算法中粒子速度的上限,*limit* 的含义如第 3.1 节中所述,表示 CABC 算法中放弃一个蜜源之前局部搜索的次数上限,p_m 和 p_c 分别是 GAAR 算法中变异和交叉概率,c_1 和 c_2 分别是 PSOAR 算法中的学习系数,|*TL*|表示 TSAR 算法中禁忌表 *TL* 的长度,"–"表示相应的算法没有设置该项参数.需要说明的是,对于 GAAR、PSOAR 和 AntAR 等 3 个群体智能算法,*SN* 一般取为 20,但在处理 256 维的高维数据集 Semeion 时则扩大了群体规模,将 *SN* 设定为 60,并在表中用 * 标记,以示区别.AntAR 算法群体规模的设置按照文献[9]的方法,设置为数据集的条件属性个数 m.因此,对于大部分的数据集,AntAR 算法的群体规模 > 20.另外,CABC 算法中侦察蜂搜索(参见式(16))用到的两个特定参数 K_1 和 K_2 分别取做 $K_1 = \max\{3, \lfloor m/3 \rfloor + 1\}$和 $K_2 = 2$.因为对于所有的测试数据集,$m \geq 6$,故 $K_1 + K_2 < m$.

表 1　测试数据表的基本信息

D-ID	数据集	*n*	*m*	*Ncls*
1	Bupa	345	6	2
2	Breast-cancer	191	9	2
3	Corral	32	6	2
4	Soybean-small	47	35	4
5	Lymphography	148	18	4
6	Soybean-large	307	35	19
7	Splice	2126	60	3
8	Mushroom	8124	22	2
9	Led24	200	24	10
10	Sponge	76	45	12
11	DNA-nominal	2000	60	3
12	Vote	300	16	2
13	Audiology	200	69	24
14	Semeion	1593	256	10

表 2　算法参数的设置

算法	*SN*	*MCN*	*V*max	*limit*	p_m	p_c	c_1	c_2	|*TL*|
TSAR	1	500	–	–	–	–	–	–	7
GAAR	20(60 *)	500	–	–	0.4	0.6	–	–	–
PSOAR	20(60 *)	500	7	–	–	–	2.0	2.0	–
AntAR	*m*	500	–	–	–	–	–	–	–
CABC	20(60 *)	500	–	5	–	–	–	–	–

对于每个参与比较的算法而言,一次迭代的计算量主要取决于适应值函数中分类精度 $\gamma_{\xi(\boldsymbol{x})}$的计算次数.在最坏情况下,计算 $\gamma_{\xi(\boldsymbol{x})}$的复杂性为 $O(mn\log n)$,即算法一次迭代的计算量为 $O(SN \times MCN \times mn\log n)$.因此,在同样的迭代次数情况下,相同群体规模的群体智能算法的计算量大致相同.

为了避免结果的偶然性,在比较实验中,每个算法都独立运行 100 次.如果输出结果(经映射 ξ)构成一个属性约简,则称算法该次运行是正常的;如果输出结果

是一个最小属性约简,则称该次运行是成功的;如果输出结果不构成一个属性约简,则称该次运行是失败的,此时可在其输出的属性子集基础上通过增加或减少属性求得一个属性约简,该结果称为后处理属性约简,并作为该次运行的最终输出结果.对于每个算法,分别用 Min$L$ 和 AvL 表示 100 次算法运行输出属性约简的最小长度和平均长度.

本文采用双样本 t-检验方法对两个不同算法 100 次运行输出的属性约简长度的均值进行差异性检验,判断均值的差异是否是统计显著的(显著性水平 $\alpha = 0.05$).具体地说,给定一个数据集,假设 T1 和 T2 分别是 CABC 算法和另一个参与比较的算法 100 次运行得到的属性约简的长度构成的样本集,$\bar{X}_1, \bar{X}_2$和 S_1, S_2 分别是这样两个样本集的均值和标准差,计算 t-检验值 t-$value$:

$$t\text{-}value = \frac{\bar{X}_1 - \bar{X}_2}{\sqrt{\frac{S_1^2}{n} + \frac{S_2^2}{n}}}$$

其中 $n = 100$. t-临界值为 $t_{\alpha/2}(198) = 1.98$.

算法的计算结果如表 3 所示.表 4 以 $a/b/c$ 的形式分别给出算法运行成功、正常和失败的比例.因为运行如果是成功的,则同时也是正常的,因此 $b + c = 1$.在表 3 和表 4 中,对于每个数据集,获得的最小的平均长度值(AvL)和运行成功比例最大的值加粗表示.双样本 t-检验的结果如表 5 所示.对于一个数据集,如果 t-检验值满足 $|t\text{-}value| > 1.98$ (在表中用 * 号标注该 t-检验值),则表明对应的两个样本集的均值差异是统计显著的(显著性水平 5%).

表 3 算法的计算结果

D-ID	TSAR		GAAR		PSOAR		AntAR		CABC	
	Min$L$	AvL	Min$L$	AvL	Min$L$	AvL	Min$L$	AvL	Min$L$	AvL
1	3	3	3	3	3	3	3	3	3	3
2	8	**8**	8	8.04	8	**8**	8	**8**	8	**8**
3	4	4.15	4	4.08	4	4.02	4	**4**	4	**4**
4	2	2.84	2	2.8	2	2.32	2	2.12	2	**2.06**
5	5	5.9	5	5.9	5	5.82	5	5.76	5	**5.69**
6	9	10.21	9	10.32	9	10.08	9	9.45	9	**9.21**
7	9	9.99	9	9.96	9	10	9	**9.24**	9	9.28
8	4	5.38	4	4.7	4	4.28	4	4.20	4	**4.06**
9	12	13.94	11	11.98	11	12.08	11	**11.74**	11	11.78
10	8	9.22	8	8.6	8	8.42	8	8.35	8	**8.19**
11	9	9.64	9	9.54	9	9.5	9	9.26	9	**9.22**
12	8	**8**	8	8.6	8	8.24	8	**8**	8	**8**
13	13	15.63	12	15.03	12	15.01	12	13.5	12	**13.43**
14	22	27.32	20	26.84	20	26.92	20	24.16	20	**24.04**

表 4 各个算法运行成功、正常和失败比例情况表

D-ID	TSAR	GAAR	PSOAR	AntAR	CABC
1	1/1/0	1/1/0	1/1/0	1/1/0	1/1/0
2	1/1/0	0.96/1/0	1/1/0	1/1/0	1/1/0
3	0.85/1/0	0.92/1/0	0.98/1/0	**1**/1/0	**1**/1/0
4	0.36/1/0	0.44/1/0	0.72/1/0	0.86/1/0	**0.90**/1/0
5	0.20/0.98/0.02	0.20/0.98/0.02	0.24/1/0	0.30/1/0	**0.34**/1/0
6	0.30/1/0	0.28/0.96/0.04	0.36/1/0	0.70/1/0	**0.79**/1/0
7	0.20/1/0	0.20/0.95/0.05	0.20/1/0	**0.76**/1/0	0.72/1/0
8	0.04/0.88/0.12	0.50/0.96/0.04	0.72/1/0	0.82/1/0	**0.94**/1/0
9	0/0.90/0.10	0.16/0.93/0.07	0.12/1/0	**0.32**/1/0	0.30/1/0
10	0.04/0.89/0.11	0.50/0.98/0.02	0.58/0.99/0.01	0.65/1/0	**0.81**/1/0
11	0.36/0.95/0.05	0.46/1/0	0.5/1/0	0.74/1/0	**0.78**/1/0
12	1/1/0	0.40/1/0	0.76/1/0	**1**/1/0	**1**/1/0
13	0/0.86/0.14	0.01/0.93/0.07	0.01/0.93/0.07	0.03/0.95/0.05	**0.05**/0.97/0.03
14	0/0.83/0.17	0.02/0.90/0.10	0.02/0.90/0.10	0.05/0.90/0.10	**0.06**/0.95/0.05

表 5　t-检验结果

D-ID	CABC vs TSAR	CABC vs GAAR	CABC vs PSOAR	CABC vs AntAR
1	0	0	0	0
2	0	−2.03*	0	0
3	−4.18*	−2.98*	−1.40	0
4	−7.63*	−5.82*	−4.953*	−1.483
5	−2.51*	−2.51*	−1.33	−0.61
6	−8.73*	−9.57*	−7.72*	−3.72*
7	−9.44*	−9.04*	−9.57*	0.70
8	−21.56*	−12.33*	−4.31*	−2.99*
9	−42.85*	−4.25*	−4.96*	0.713
10	−16.56*	−6.51*	−3.63*	−2.57*
11	−6.57*	−5.08*	−4.31*	−0.66
12	0	−8.13*	−5.58*	0
13	−28.79*	−23.52*	−23.61*	−0.86
14	−17.72*	−14.04*	−15.13*	−0.58

从上述 3 个表的数据可以看出,与 TSAR、GAAR 和 PSOAR 这 3 个算法相比,CABC 算法不仅具有更高的算法运行成功的比例(即更高的求得最小属性约简的概率),而且在大多数数据集上能够求得平均长度明显更小的属性约简(统计显著性水平 $\alpha = 0.05$).与 AntAR 算法相比,总体而言,CABC 算法同样具有更好的结果和性能.从算法运行成功的比例和输出结果的平均长度来看,在 4 个数据集(Bupa, Breast-cancer, Corral 和 Vote)上,CABC 算法和 AntAR 算法是相同的,均为 100%;在另外 2 个数据集(Splice 和 Led24)上,AntAR 算法略优于 CABC 算法,而在其他 9 个数据集上 CABC 算法则优于 AntAR 算法,特别是在其中 3 个数据集(Soybean-large, Mushroom 和 Sponge)上,CABC 算法比 AntAR 算法求得平均长度明显更小的属性约简(显著性水平 $\alpha = 0.05$).

综上所述,与其他几个元启发属性约简算法相比,本文提出的基于组合人工蜂群搜索机制的 CABC 算法能够更好地求解 NP-难的最小属性约简问题.

5　结论

本文提出了一个组合人工蜂群算法 CABC,用于求解 NP-难的最小属性约简问题.在保持连续型人工蜂群算法基本搜索机制的基础上,针对离散型问题的特点,在 CABC 算法中引入若干适应性修改策略和改进措施,例如:在局部搜索中考虑两个属性子集大小的差异以控制邻域的大小;充分利用群体最好解的信息;引领蜂及其跟随蜂采用不同的局部搜索策略以提高多样性等.与分别基于禁忌搜索、遗传算法、粒子群优化和蚁群算法等 4 个元启发式搜索方法的属性约简算法相比,CABC 算法通常能够取得更好的求解效果,并提高了获得最小属性约简的可能性.因此,对于最小属性约简问题的求解,本文提出的组合人工蜂群算法是有效可行的.

本文算法经过适当的修改可以用于邻域粗糙集[19]的最小属性约简的求解.另外,通过有效地借鉴与融合其他群体智能算法的搜索机制,可望进一步提高本文算法的求解质量和性能.我们将在后续的工作中对此进行探索和研究.

参考文献

[1] Pawlak Z, Slowinski R. Rough set approach to multi-attribute decision analysis[J]. European Journal of Operational Research, 1994, 72(3): 443 – 459.

[2] 张腾飞,肖健梅,王锡淮.粗糙集理论中属性相对约简算法[J].电子学报,2005,33(11):2080 – 2083.
Zhang Teng-fei, Xxiao Jian-mei, Wang Xi-huai. Algorithms of attribute relative reduction in rough set theory[J]. Acta Electronica Sinica, 2005, 33(11): 2080 – 2083. (in Chinese)

[3] 苗夺谦,周杰,等.基于代数方程组的属性约简研究[J].电子学报,2010,38(5):1021 – 1027.
Miao Duo-qian, Zhou Jie, et al. Research of attribute reduction based on algebraic equations[J]. Acta Electronica Sinica, 2010, 38(5): 1021 – 1027. (in Chinese)

[4] 胡峰,王国胤.属性序下的快速约简算法[J].计算机学报,2007,30(8):1429-1435.
Hu Feng, Wang Guo-yin. Quick reduction algorithm based on attribute order[J]. Chinese Journal of Computers, 2007, 30(8): 1429 – 1435. (in Chinese)

[5] 杨明.决策表中基于条件信息熵的近似约简[J].电子学报,2007,35(11):2156 – 2160.
Yang Ming. Approximate reduction based on information reduction in decision tables[J]. Acta Electronica Sinica, 2007, 35(11):2156 – 2160.(in Chinese)

[6] 叶东毅.Jelonek 属性约简算法的一个改进[J].电子学报,2000,28(12):81 – 82.
Ye Dong-yi. An improvement to Jelonek's attribute reduction algorithm[J]. Acta Electronica Sinica, 2000, 28(12):81 – 82.(in Chinese)

[7] 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.

[8] HedarA R, Wang J. Tabu search for attribute reduction in rough set theory[J]. Soft Computing, 2008, 12(9):909 – 918.

[9] Jensen R, Shen Q. Semantics preserving dimensionality reduction: Rough and fuzzy rough-based approaches[J]. IEEE Transactions on Knowledge and Data Engineering, 2004, 16(12): 1457 – 1471.

[10] Wang X Y, Yang J. et al. Feature selection based on rough sets and particle swarm optimization[J]. Pattern Recognition Letters, 2007, 28(4):459 – 471.

[11] Ye D Y, Chen, Z J, Ma S L. A novel and better fitness evaluation for rough set based minimum attribute reduction problem[J]. Information Sciences, 2013, 222(2):223 – 233.

[12] Karaboga D, Basturk B. A powerful and efficient algorithm for numerical function optimization: artificial bee colony (ABC) algorithm[J]. Journal of Global Optimization, 2007, 39(3): 459 – 471.

[13] Karaboga D, Basturk B. On the performance of artificial bee colony (ABC) algorithm[J]. Applied Soft Computing, 2008, 8(1):687 – 697.

[14] Karaboga D, Basturk B. A comparative study of artificial bee colony algorithm[J]. Applied Mathematics and Computation, 2009, 214(1):108 – 132.

[15] Gao W F, Liu S F. A modified artificial bee colony algorithm[J]. Computers and Operations Research, 2012, 39(3):687 – 697.

[16] Banharnsakun A, Achalakul T, Sirinaovakul B. The best-so-far selection in artificial bee colony algorithm[J]. Applied Soft Computing, 2010, 11(2):2888 – 2901.

[17] Singh A. An artificial bee colony algorithm for the leaf-constrained minimum spanning tree problem[J]. Applied Soft Computing, 2009, 9(2):625 – 631.

[18] Davidovic T, et al. Bee colony optimization for scheduling independent tasks to identical processors[J]. Journal of heuristics, 2012, 18 (4):549 – 569.

[19] 朱鹏飞,胡清华,于达仁.基于随机化属性选择和邻域覆盖约简的集成学习[J].电子学报,2012,40(2):273 – 279.
Zhu Peng-fei, Hu Qin-hua, Yu Da-ren. Ensemble learning based on randomized attribute selection and neighborhood covering reduction[J]. Acta Electronica Sinica, 2012, 40(2):273 – 279.(in Chinese)

作者简介

叶东毅 男,1964 年生于福建福州.教授,博士.主要研究方向为计算智能和数据挖掘.
E-mail: yiedy@fzu.edu.cn

陈昭炯 女,1964 年生于福建福州.教授.研究方向为人工智能和图像处理.