一种求解类覆盖问题的混合算法*

黄艳新+, 周春光, 邹淑雪, 王 岩

(吉林大学 计算机科学与技术学院,吉林 长春 130012)

A Hybrid Algorithm on Class Cover Problems

HUANG Yan-Xin+, ZHOU Chun-Guang, ZOU Shu-Xue, WANG Yan

(College of Computer Science and Technology, Jilin University, Changchun 130012, China)

+ Corresponding author: Phn: +86-431-5166477, E-mail: huangyx@jlu.edu.cn, http://www.jlu.edu.cn

Received 2003-10-18; Accepted 2004-05-08

Huang YX, Zhou CG, Zou SX, Wang Y. A hybrid algorithm on class cover problems. *Journal of Software*, 2005,16(4):513−522. DOI: 10.1360/jos160513

Abstract: An extended class cover problem is presented and then it is reduced to a constrained multi-objective optimization problem. Solving this problem is significantly important to construct a robust classification system. Therefore, through analyzing the parameters of the binary particle swarm optimization, the conclusion that the binary particle swarm optimization can not only explore the search space efficiently, but also utilize the apriori knowledge adequately, is drawn in this paper. Furthermore, a hybrid algorithm combined with the conventional greedy algorithm and binary particle swarm optimization algorithm is proposed to deal with the extended class cover problem. The proposed algorithm can get a better solution in less runtime and the simulated comparative results with other algorithms show its feasibility and validity.

Key words: class cover problem; binary particle swarm optimization; hybrid algorithm

摘 要: 提出一种扩展的类覆盖问题,并将它归纳为一个有约束的多目标优化问题模型,该问题的解决对构建强壮的分类识别系统具有重要的意义.因此,通过对二进制粒子群算法参数特性的深入分析,阐明二进制粒子群算法不仅具有良好的全局搜索特性,而且能够充分利用已有的先验知识.进而提出一种贪心算法与二进制粒子群优化算法相结合的混合算法求解扩展的类覆盖问题,该算法在获得更优解的同时,仍具有较快的运算速度.多种算法的比较结果表明了算法的有效性和可行性.

关键词: 类覆盖问题;二进制粒子群优化;混合算法

中图法分类号: TP18 **文献标识码**: A

* Supported by the National Natural Science Foundation of China under Grant Nos.60175024, 60433020 (国家自然科学基金); the Key Laboratory for Symbolic Computation and Knowledge Engineering of Ministry of Education of China (教育部"符号计算与知识工程"重点实验室基金项目)

作者简介: **黄艳新**(1967−),男,吉林白城人,博士,副教授,主要研究领域为计算智能,模式识别,生物信息学;**周春光**(1947−),男,博士,教授,主要研究领域为计算智能,机器味觉,图像处理;**邹淑雪**(1976−),女,讲师,主要研究领域为计算智能,生物信息学;**王岩**(1978−),男,助教,主要研究领域为计算智能;生物信息学.

Adam Cannon 和 Lenore Cowen[1]首次提出类覆盖问题(class cover problem,简称 CCP),并证明了其 NP 难解性,快速而有效地计算类覆盖问题的最优解或准最优解,对于机器学习[2]、模式识别[3]以及数据挖掘[2]等领域的研究都具有重要的意义.目前,求解类覆盖问题的主要方法是采用文献[4]的贪心算法(greedy algorithm)[2,3].本文提出了一种扩展的类覆盖问题,并将它归纳为一个有约束的多目标优化问题模型,同时提出一种求解该扩展问题的混合算法(hybrid algorithm),该算法将贪心算法和二进制粒子群优化(binary particle swarm optimization)算法有机结合使之能以较少的迭代次数求出更优的解.本文最后通过实验对各种算法进行了比较.

1 类覆盖问题的定义

文献[1]给出的类覆盖问题定义如下:

类覆盖问题 I.

设 m 维空间中有两类数据点,不妨设一类是红色点集合,记为 R;一类是蓝色点集合,记为 B,试求 B 的一个最小子集 S,使得以 S 中的点为中心的超球体可以覆盖 B 中全部点集而不覆盖 R 的任一点.显然,类覆盖问题可以转化为如下的单目标优化问题模型:

$$\begin{aligned}&\min K = |S|\\ &s.t. \max_{v\in B}\{d(v,S)\} < \min_{w\in R}\{d(w,S)\}\end{aligned} \quad \text{(模型 1)}$$

其中,$S \subseteq B$;$|S|$表示求集合 S 的基数;$d(v,S) = \min_{s\in S}\{d(v,s)\}$ 定义为点到集合的距离.以 S 中的任一数据点 x 为中心,以 $d(x,R)$ 为半径可以构造一个超球体,以 S 中的全部数据点为中心的超球体是 B 的一个类覆盖.

依据模型 1 设计算法,其主要缺陷在于:当分类样本数据存在噪声干扰时,求得的 K 值会极大地偏离真正的最优解,即算法强壮性差.文献[3]将类覆盖问题进一步描述为求解类覆盖获取有向图(class cover catch digraph,简称 CCCD)的最小支配集问题,其定义如下.

类覆盖问题 II.

针对 m 维空间中的两类数据点集 R 和 B,相对于 R,建立 B 的类覆盖获取有向图 $D=(V,A)$,其中顶点集 $V=B$;A 表示 B 上有序顶点序偶 (x,x') 的集合,(x,x') 表示 x 到 x' 的有向边.$(x,x')\in A$,当且仅当,x' 位于以 x 为中心和 $r(x)$ 为半径的超球体内,$r(x)=d(x,R)=\min_{r\in R}\{d(x,r)\}$.$\forall v\in V$,定义 $N(v)=\{w \mid w\in V,(v,w)\in A\}$,令 $N(S)=\bigcup_{v\in S}N(v)$,那么有向图 $D=(V,A)$ 的一个支配集 $S\subseteq B$ 满足:$N(S)=V$,即 $\forall w\in V$,要么 $w\in S$,要么 $\exists v\in S$,使得有向边 $(v,w)\in A$.试求类覆盖获取有向图 D 的一个最小支配集.

显然,类覆盖问题 I 与问题 II 是等价的[3].为抵制噪声干扰,文献[3]引入参数 $\alpha,\beta\in[0,1]$,α 用于控制全部超球体至少要覆盖 B 中 $\alpha|B|$ 个数据点;β 决定以 B 上的顶点为中心的超球体最多可以包含 $\beta|R|$ 个 R 中的数据点,那么以 S 中数据点 x 为中心的超球体半径可以定义为 $r_\beta(x)=d_\beta(x,R)=\min_{r\in R}\{d_\beta(x,r)\}$,表示 x 到 R 中数据点的由小到大排序的距离中第 $\beta|R|$ 小的距离测度.

求解图的最小支配集也是 NP-Hard 问题[3],实用中只能求解图的准最小支配集.目前最常用的算法是文献[4]给出的贪心算法.对类覆盖获取有向图直接应用文献[4]的贪心算法获得的超球体覆盖,尤其是针对有聚类特征的分类数据,一般是有一些半径较大的超球体,其中心位于数据点内部区域,同时有大量的半径很小的超球体位于数据点的边界处.在模式识别应用中,这不利于产生强壮的分类规则,尽管引入参数 α 和 β,使上述情况有所改善,但仍然没有很好地解决这个问题[3].

2 扩展的类覆盖问题和改进的贪心算法

本文进一步提出一种类覆盖问题的扩展定义.

扩展的类覆盖问题.

设 m 维空间中有两类数据点集 R 和 $B,\alpha,\beta\in[0,1]$.试求 B 的一个最小子集 S,使得以 S 中的点为中心的超球体至少覆盖 B 中 $\alpha|B|$ 个数据点,每个超球体至多包含 R 中 $\beta|R|$ 个数据点,同时要求超球体的半径尽量均匀.

显然,扩展的类覆盖问题可以归纳为如下多目标优化问题模型:

$$\min K = |S|, Var = \sqrt{\frac{1}{|S|-1}\sum_{v\in S}(r_\beta(v)-\bar{r})^2}$$
$$s.t.\begin{cases} r_\beta(v) \le \gamma \mathrm{Max}D_\beta, \forall v \in S \\ \left|\{w \mid w \in B, \text{and}, \exists v \in S, d(v,w) \le r_\beta(v)\}\right| \ge \alpha |B| \\ \alpha, \beta, \gamma \in [0,1] \end{cases} \quad \text{(模型 2)}$$

其中, $S \subseteq B$ 是关于 B 的类覆盖获取有向图的一个准最小支配集; $\bar{r} = \frac{1}{|S|}\sum_{v\in S} r_\beta(v)$, $r_\beta(v)$ 的定义如上节,距离函数可以采用欧氏(Euclid)距离.针对模型 2,本文提出一种对文献[4]贪心算法的改进算法.

定义 3.1. 设图 $G=(B,E)$, D 是图 G 的准最小支配集, $\forall j \in B$,称 j 被覆盖,当且仅当, $j \in D$,或 $\exists v \in D$,使得边 $(v,j) \in E$.

改进的贪心算法.

令初始准最小支配集 $D=\varnothing$,未被覆盖的顶点集合 $C=B$,做:

① 生成 B 相对于 R 的类覆盖获取有向图,具体方法是:

a) $\forall x \in B$,计算 $d_\beta(x,R)$,令 $\mathrm{Max}D_\beta = \max_{x\in B}\{d_\beta(x,R)\}$;将 B 中各数据点的半径做如下变换:若 $d_\beta(x,R) \le \gamma \mathrm{Max}D_\beta$,则 $d_\beta(x,R)$ 值不变;否则取 $d_\beta(x,R) = \gamma \mathrm{Max}D_\beta$,其中 $0 \le \gamma \le 1$,用于控制覆盖超球体的半径(本文取 $\gamma = 0.4 \sim 0.7$,见实验部分);

b) 生成有向图 $G=(B,E)$,方法是:对每一个 $x \in B$,对所有的 $y \in B$,计算 $d(x,y)$,当 $d(x,y) \le d_\beta(x,R)$时,产生有向边 $(x,y) \in E$;

② 若 $|C| \le (1-\alpha)|B|$,输出准最小支配集 D,算法结束;否则, $\forall x \in C$,计算其覆盖度 $cover(x)|\{y|y\in C, (x,y)\in E\}|$;

③ 取 $z \in C$,满足 $cover(z) = \max_{x\in C}\{cover(x)\}$, $D = D \cup \{z\}$, $C = C - \{x \mid x \in C, (z,x) \in E\}$,转②.

设 $|B|=N$, $|R|=M$,产生的准最小支配集基数 $|D| = \rho(D)$,不考虑数据的维数,则算法的时间复杂度为 $O(NM + N(N-1) + N\rho(D))$,其中第 1 项为 B 中各点最大覆盖球体半径的距离计算次数;第 2 项为生成 B 相对 R 的有向图的距离计算次数;第 3 项为生成最小支配集 D 的距离计算次数.若设 $N \approx M$, $\rho(D) << N$,则算法的时间复杂度约为 $O(N^2)$.

3 求解扩展的类覆盖问题的混合算法

实值(real-valued)粒子群优化算法是 Kennedy 和 Eberhart 在 1995 年提出的一种基于群体智能(swarm intelligence)原理的优化模型,其灵感来源于自然界鸟类和蚂蚁等的群体觅食过程[5].粒子群优化模型与进化计算方法(如 GA,GP,EC 等)有着密切的联系,它们都基于群体的随机搜索机制,但粒子群优化模型强调群体内个体间的互相协作、共生共存,而非"物竞天择、适者生存".其主要的优势体现在粒子的收敛性保证、算法易于实现、不需要目标函数的梯度(gradient)信息等方面.目前,实值粒子群优化算法已经在许多困难的单峰(unimodal)和多峰(multimodal)优化问题中表现出了良好的性能,并吸引了众多学者对它进行研究[5].二进制粒子群优化算法是 Kennedy 和 Eberhart 在 1997 年提出的对实值粒子群优化算法的简单修正,目的是使它能够处理离散的优化问题[6].

使用改进的贪心算法求解扩展的类覆盖问题不容易得到较优的解,而二进制粒子群优化算法属于全局搜索算法,直接使用它求解扩展的类覆盖问题一般需要很大的迭代次数才能得到较优的解.本文提出了一种改进的贪心算法和二进制粒子群优化算法的混合算法,该算法可以在较少的迭代次数下得到较优的解,且算法经过适当修改对原始类覆盖问题同样适用.

3.1 数据的存储

首先将 B 和 R 中数据存入大小为 N 和 M 的一维数组,特别地,将 B 中数据点存入大小为 N 的一维数组

Array_B,使得 *B* 中欧氏距离较近的数据点在 *Array_B* 中的位置也较近,具体算法:

① 令 k=0;

② 若 $B=\varnothing$,则算法结束;否则,转③;

③ 令 $k=k+1$,若 k=1,则任取 *B* 中一个数据点 y,令 $Array\_B(k)=y, B=B-\{y\}$,转②;否则,取 *B* 中一个数据点 y,满足 $d(Array\_B(k-1),y)=\min_{x\in B}\{d(Array\_B(k-1),x)\}$,其中,函数 $d()$ 表示求两点间的欧氏距离,令 $Array\_B(k)=y, B=B-\{y\}$,转②.

3.2 生成类覆盖获取有向图对应的关联矩阵

构建 *B* 相对于 *R* 的类覆盖获取有向图并生成该有向图对应的关联矩阵.具体方法是:

① 生成 *B* 相对于 *R* 的类覆盖获取有向图 $G=(B,E)$ (方法同改进的贪心算法),将 *B* 中各数据点的半径存入相应的大小为 *N* 的一维数组;

② 根据有向图 $G=(B,E)$,生成图 *G* 对应的 $N\times N$ 的 0-1 关联矩阵 M_G,其中,$M_G(i,j)=1$(或 0)表示存在(或不存在)结点 i 到 j 的有向边,$i,j\in\{1,2,...,N\}$.

显然,由于图 $G=(B,E)$ 是有向图,其对应的关联矩阵不一定是对称的.

3.3 粒子的二进制编码

群体中的粒子由长度为 *N* 的二进制编码构成,每个粒子对应一个问题的解.设某个粒子 I_k 的二进制编码为 $I_k=b_1^{(k)}b_2^{(k)}\ldots b_N^{(k)}$,其中,$b_i^{(k)}$=1(或 0)表示 *B* 中数据点 i 属于(或不属于)该粒子对应的准最小支配集,$i\in\{1,2,...,N\}$;$k\in\{1,2,...,L\}$,*L* 表示进化群体中的粒子总数(本文取 L=20).

3.4 粒子适应度函数

粒子 I_k 的适应度衡量主要依靠 3 个参数:

(1) 粒子编码中"1"的个数 $C(I_k)=b_1^{(k)}+b_2^{(k)}+\ldots+b_N^{(k)}$,$C(I_k)$ 表示粒子 I_k 对应的准最小支配集的基数;

(2) 准最小支配集所覆盖的 *B* 中数据点数 $R(I_k)$;

(3) 粒子编码中"1"所对应数据点的半径的样本标准差 $Var(I_k)$.

显然,$C(I_k)$ 越小,表示该粒子对应的准最小支配集的基数越小,其适应度应该越大;$R(I_k)$ 越大,表示 *B* 中被覆盖的数据点越多,其适应度也应该越大,考虑到当 $R(I_k)\geq\alpha|B|$ 时,*B* 中被覆盖的数据点达到模型 2 的约束条件,此时应该给该粒子一个适当的奖励;$Var(I_k)$ 越小,其适应度也应该越大.基于以上分析,给出粒子 I_k 的适应度函数定义:

$$F(I_k)=\begin{cases}\lambda_1\times\dfrac{N-C(I_k)}{N}+\lambda_2\times\dfrac{R(I_k)}{N}+\lambda_3\times\dfrac{Q-Var(I_k)}{Q}, & \text{if } R(I_k)<\alpha N\\ \lambda_1\times\dfrac{N-C(I_k)}{N}+\lambda_2\times\dfrac{R(I_k)}{N}+\lambda_3\times\dfrac{Q-Var(I_k)}{Q}+0.2, & \text{if } R(I_k)\geq\alpha N\end{cases}\tag{1}$$

其中,λ_1,λ_2 和 λ_3 是权系数.*Q* 是大于 $Var(I_k)$ 的实数,*Q* 值可根据下述定理确定.

定理 3.1. 设 $a_1,a_2,...,a_n$ 是 *n* 个实数的序列,任取其中 *k* 个实数的子序列 $a_{i_1},a_{i_2},...,a_{i_k}$,$i_1,i_2,...,i_k\in\{1,2,...,n\}$,$2\leq k\leq n$,定义 $Var(a_{i_1},a_{i_2},...,a_{i_k})=\sqrt{\frac{1}{k-1}\sum_{j=1}^{k}(a_{i_j}-\overline{a}_k)^2}$,其中,$\overline{a}_k=\frac{1}{k}\sum_{j}^{1}a_{i_j}$.若令 $a_{\min}=\min(a_1,a_2,\ldots,a_n)$,$a_{\max}=\max(a_1,a_2,\ldots,a_n)$,那么,必有 $Var(a_{i_1},a_{i_2},...,a_{i_k})\leq Var(a_{\min},a_{\max})$.

证明:令 $f(a_{i_1},a_{i_2},...,a_{i_k})=Var(a_{i_1},a_{i_2},...,a_{i_k})^2=\frac{1}{k-1}\sum_{j=1}^{k}(a_{i_j}-\overline{a}_k)^2$,求函数 f 在闭区域 $a_{\min}\leq a_{i_j}\leq a_{\max}$,$j=1,2,...,k$ 的最大值.取 $\frac{\partial f}{\partial a_{i_j}}=0$,易证函数 f 在闭区域内 $a_{i_1}=a_{i_2}=...=a_{i_k}$ 处取得最小值 0.下面考察函数 f 在边界处的取值,不妨设子序列 $a_{i_1},a_{i_2},...,a_{i_k}$ 中有 *p* 个变量取 $a_{\max}$(与具体哪 *p* 个变量取 $a_{\max}$ 无关),其余 $k-p$ 个变量

取 $a_{\min}$,$1\le p\le k$,计算可得 $f=\frac{1}{k-1}\frac{1}{k^2}\cdot(a_{\max}-a_{\min})^2\cdot((k-p)^2p+p^2(k-p))$,此时把 p 看作自变量.易证,当 $p=\frac{k}{2}$ 时,f 取得最大值,且函数 f 是以自变量 $p=\frac{k}{2}$ 为中心对称的单峰函数.考虑到 $1\le p\le k$,且 p 只能取整数,当 k 是偶数时,有

$$f=\frac{1}{k-1}\frac{1}{k^2}\cdot(a_{\max}-a_{\min})^2\cdot\left(\left(k-\frac{k}{2}\right)^2\frac{k}{2}+\left(\frac{k}{2}\right)^2\left(k-\frac{k}{2}\right)\right)=\frac{1}{4(k-1)}\cdot(a_{\max}-a_{\min})^2,$$

由 $k\ge 2$,得

$$f\le\frac{1}{2}(a_{\max}-a_{\min})^2=Var(a_{\max}-a_{\min})^2;$$

当 k 是奇数时,有

$$f=\frac{1}{k-1}\frac{1}{k^2}.(a_{\max}-a_{\min})^2\cdot\left(\left(k-\frac{k-1}{2}\right)^2\frac{k-1}{2}+\left(\frac{k-1}{2}\right)^2\left(k-\frac{k-1}{2}\right)\right)=\frac{k+1}{4k}\cdot(a_{\max}-a_{\min})^2,$$

由 $k\ge 2$,得

$$f\le\frac{1}{2}\cdot(a_{\max}-a_{\min})^2=Var(a_{\max}-a_{\min})^2.$$

综上所述,得到 $Var(a_{i_1},a_{i_2},...,a_{i_k})\le Var(a_{\min},a_{\max})$. □

根据定理 4.1,可取 $Q=Var(R_{\min},R_{\max})$,其中 $R_{\min}$,$R_{\max}$ 分别表示 B 中数据点的最小半径值和最大半径值,这保证了 $0\le\frac{Q-Var(I_k)}{Q}\le 1$.本文取权系数 $\lambda_1=0.55$,$\lambda_2=0.1$ 和 $\lambda_3=0.15$,这使得适应度函数 $0\le F\le 1$.适应度参数 F 中 $R(I_k)$ 的计算是最费时的.设 $M_G(i,1:N)$ 表示关联矩阵 M_G 的第 i 行向量,$M_G(1:N,j)$ 表示关联矩阵 M_G 的第 j 列向量,$i,j\in\{1,2,...,N\}$,采用如下算法计算 $R(I_k)$:

① 令 p=1,粒子 $I_k=b_1^{(k)}b_2^{(k)}...b_N^{(k)}$;$T$=[0,0,...,0]^T 是大小为 N 的一维 0 向量;

② 若 $b_p^{(k)}=1$,则 $T=T\vee M_G(1:N,p)$,转③;否则,直接转③;

③ 若 $p<N$,令 $p=p+1$,转②;否则,计算 $R(I_k)=\sum_{i=1}^{N}T(i)$,其中 $T(i)$ 表示向量 T 的第 i 个分量,算法结束.

算法中 $T=T\vee M_G(1:N,p)$ 表示两个大小为 N 的一维 0-1 向量的按位或运算,最后向量 T 中分量为"1"的个数即为该粒子所对应的准最小支配集所覆盖的 B 中数据点数.因此,基于图的关联矩阵计算 $R(I_k)$ 至多进行 N 次的向量或运算.

3.5 粒子的迭代公式及参数分析

设粒子群含 L 个粒子,每个粒子相当于 N 维离散空间中的一个活动点.粒子 i 在时刻 t 的速度、位置、个体最好位置和全局最好位置分别用 $v_i(t)$,$x_i(t)$,$x_i^{(p)}(t)$ 和 $x_i^{(g)}(t)$ 表示,那么粒子 i 速度和位置的各维分量迭代公式如下[7]:

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

$$x_{ij}(t+1)=\begin{cases}0, & \text{若}\rho\ge Sig(v_{ij}(t+1))\\ 1, & \text{若}\rho<Sig(v_{ij}(t+1))\end{cases} \tag{3}$$

其中,$i=1,2,...,L$;$j\in\{1,2,...,N\}$,表示粒子编码中各分量的维数;$r_{1j}(t)$ 和 $r_{2j}(t)$ 是(0,1)上均匀分布的随机数,w,c_1 和 c_2 是加速系数;$\rho\sim U[0,1]$,是(0,1)区间上均匀分布的随机变量;$Sig()$ 表示 sigmoid 函数,本文取 $Sig(x)=\frac{1}{1+\exp(-x)}$ [7].

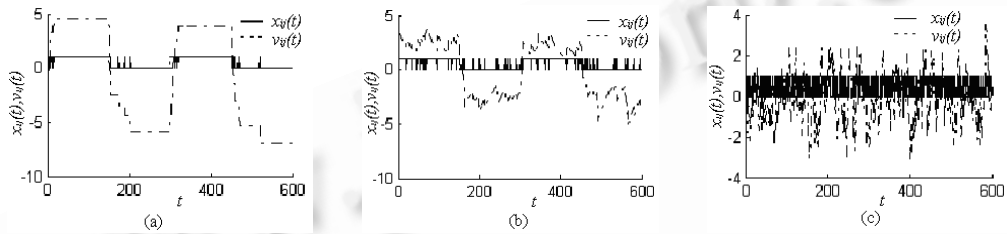
二进制粒子群优化算法的速度迭代公式与实值粒子群优化算法相同,但参数 $v_{\max}$ 以及加速系数 w,c_1 和 c_2 的意义与其在连续粒子群优化算法中的意义完全不同.文献[7]规定粒子的最大速度 $v_{\max}$ 满足 $|v_{\max}|=4$,这使得 $0.018\le Sig(v_{ij}(t+1))\le 0.0982$,类似于遗传算法中的变异操作概率 p_m,它避免算法的早熟收敛.$v_{\max}$ 越大,算法倾向于当前最优解附近的局部搜索,$v_{\max}$ 越小,算法倾向于全局搜索,特别地,当 $v_{\max}=0$ 时,算法变成纯随机搜索算法[7].

目前,对加速系数 w,c_1 和 c_2 的研究成果不多,一般取 $w=c_1=c_2=1$.本文对加速系数 w,c_1,c_2 和 $v_{\max}$ 的意义进行了探讨,并提出了确定这些参数的方案.

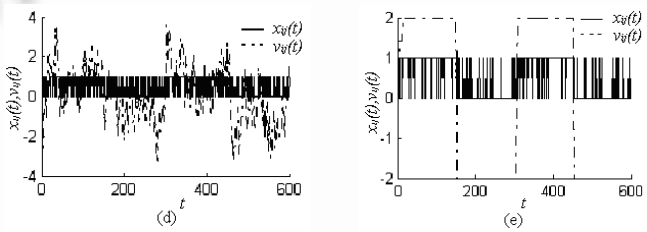
任取粒子群中的一个粒子 i,分以下几种情形讨论当迭代次数 $t\to\infty$ 时粒子的运动轨迹:

A. 当 $x_{ij}^{(p)}(t)=x_j^{(g)}(t)$ 时

由迭代公式(2)和(3),若 $x_{ij}(t)\neq x_{ij}^{(p)}(t)$ 和 $x_j^{(g)}(t)$,随迭代次数 t 增加,当 $x_{ij}^{(p)}(t)=x_j^{(g)}(t)=1$ 时,$v_{ij}(t)\to\infty$,这使得 $x_{ij}(t)\to 1$;当 $x_{ij}^{(p)}(t)=x_j^{(g)}(t)=0$ 时,$v_{ij}(t)\to-\infty$,这使得 $x_{ij}(t)\to 0$.因此,粒子会收敛到全局最优粒子所在位置.加速系数 c_1 和 c_2 越大,粒子收敛的速度越快;如果加速系数 $w>1$,那么 w 越大,粒子收敛的速度越快.如果惯性权值系数 $w<1$,当达到 $x_{ij}(t)=x_{ij}^{(p)}(t)$ 和 $x_j^{(g)}(t)$,加速系数 c_1 和 c_2 不再起作用,随迭代次数 t 增加,w 将使 $v_{ij}(t)\to 0$,即使得 $x_{ij}(t)$ 产生变异的概率趋近于 0.5.因此,$w<1$ 会使 $x_{ij}(t)\to x_{ij}^{(p)}(t)$ 和 $x_j^{(g)}(t)$ 的过程中产生变异,且 w 越小,产生变异的概率就越大.图 1(a)和(b)给出了两组典型参数取值下 $x_{ij}(t)$ 和 $v_{ij}(t)$ 的运动轨迹.



(a) w=1.2, c_1=c_2=1 (b) w=0.97, c_1=c_2=1 (c) w=1, c_1=c_2=1.5



(d) w=1, c_1=0.8, c_2=1.5 (e) w=1.2, c_1=c_2=1,$v_{\max}$=2

Fig.1 The varying curves of the particle position $x_{ij}(t)$ and velocity $v_{ij}(t)$ with respect to time t

图 1 粒子的位置 $x_{ij}(t)$和速度 $v_{ij}(t)$随时间 t 变化曲线图

图 1(a)和(b)中取 $x_{ij}^{(p)}(t)=x_j^{(g)}(t)=\begin{cases}1,\text{当}t=0\sim150,301\sim450\\0,\text{当}t=151\sim300,451\sim600\end{cases}$,对粒子的最大速率 $v_{\max}$ 没有限制,初始速度 $v_{ij}(0)$ 取[−5,+5]之间的随机数,初始位置取 $x_{ij}(0)=0$.由图 1(a)可知,当 $w>1$ 时,只要 $x_{ij}(t)\neq x_{ij}^{(p)}(t)$ 和 $x_j^{(g)}(t)$,随迭代次数 t 的增加,$x_{ij}(t)\to x_{ij}^{(p)}(t)$ 和 $x_j^{(g)}(t)$,在开始阶段,$x_{ij}(t)$ 会存在一些变异,但每次变异,都使 $|v_{ij}(t)|$ 增大,进而使再次变异的概率变小.图 1(b)的实验条件与图 1(a)相同,仅取 $w<1$,由图 1(b)可知,在 $x_{ij}(t)\to x_{ij}^{(p)}(t)$ 和 $x_j^{(g)}(t)$ 的过程中,产生变异的概率有较大的增加.

B. 当 $x_{ij}^{(p)}(t)\neq x_j^{(g)}(t)$ 时

由迭代公式(2)和(3),此时必有 $x_{ij}(t)=x_{ij}^{(p)}(t)$ 或 $x_j^{(g)}(t)$,当 $x_{ij}(t)=x_{ij}^{(p)}(t)$ 时,加速系数 c_1 不起作用,c_2 起作用,随迭代次数 t 的增加,$x_{ij}(t)\to x_j^{(g)}(t)$.而一旦 $x_{ij}(t)=x_j^{(g)}(t)$,加速系数 c_2 不起作用,c_1 起作用,那么随迭代次数 t 的增加,$x_{ij}(t)\to x_{ij}^{(p)}(t)$.因此,粒子的位置坐标会在 $x_{ij}^{(p)}(t)$ 和 $x_j^{(g)}(t)$ 之间振荡.图 1(c)和图 1(d)给出了两组典型参数取值的粒子运动轨迹.

图 1(c)和图 1(d)中取 $x_j^{(g)}(t)=\begin{cases}1,\text{当}t=0\sim150,301\sim450\\0,\text{当}t=151\sim300,451\sim600\end{cases}$,$x_{ij}^{(p)}(t)=1-x_j^{(g)}(t)$,对粒子的最大速率 $v_{\max}$ 没有限制,初始速度 $v_{ij}(0)$ 取[−5,+5]之间的随机数,初始位置取 $x_{ij}(0)=0$.由图 1(c)可知,当取 c_1=c_2(=1.5)时,无论

$x_j^{(g)}(t)$ 和 $x_{ij}^{(p)}(t)$ 的取值如何,$x_{ij}(t)$ 均在[0,1]之间均匀振荡,此时粒子群接近于随机搜索.由图 1(d)可知,当取 c_1=0.8,c_2=1.5(即 $c_2>c_1$)时,由于 c_2 的作用,尽管 $x_{ij}(t)$ 也是在[0,1]之间振荡,但更倾向于 $x_{ij}(t)\to x_j^{(g)}(t)$,即粒子具有局部搜索特性.因此,$c_1$ 和 c_2 应适当选取.

C. 限制 $v_{\max}$ 取值对粒子运动轨迹的影响

若对 $v_{\max}$ 做出约束,则随 $t\to+\infty$,$v_{ij}(t)\to v_{\max}$ 或 $v_{ij}(t)\to -v_{\max}$,可使得在 $x_{ij}(t)\to x_{ij}^{(p)}(t)$或 $x_j^{(g)}(t)$的过程中,存在一定概率的变异.图 1(e)给出了对 $v_{\max}$ 限制的粒子运动轨迹,实验条件与图 1(a)完全相同,仅限定 $v_{\max}$=2,与图 1(a)相比,粒子运动轨迹出现较大的振荡.因此,限制 $v_{\max}$ 与取惯性权值系数 w<1 有相似的效果.

根据上述分析可知,通过精选参数可使二进制粒子群算法在全局搜索的过程中有效利用已有的先验知识,这使贪心算法和二进制粒子群算法的融合成为可能.

3.6 求解扩展的类覆盖问题的混合算法

混合算法的基本思想是将改进的贪心算法求出的问题解初始化为二进制粒子群优化算法的一个粒子,精选算法的参数,使其全局搜索的同时兼具局部搜索特性,即使得算法趋向于在全局最优粒子附近搜索更优的粒子,从而获得比改进的贪心算法更优的解.同时为避免早熟收敛和出现不活动粒子,算法监控每个粒子并激活那些不活动粒子.

使用混合算法必须解决的一个问题是:海明(Haming)距离较近的二进制编码的粒子并不意味着各粒子包含的解集中相应的数据点具有较近的欧氏距离.换言之,由迭代公式(2)和(3),通过精选算法参数可使粒子趋向于全局最优粒子(以海明距离测量),但这并不意味着算法在全局最优粒子包含的解集中各数据点附近(以欧氏距离测量)搜索更优的解.

为解决这一问题,首先,采用第 3.1 节中所述方法存储 B 中数据点,使得 B 中欧氏距离较近的数据点在其存储的一维数组中的位置也较近;其次,对迭代公式(2)进行修正如下:

$$v_{ij}(t+1)=wv_{ij}(t)+c_1r_{1j}(t)(x_{ij}^{(p)}(t)-x_{ij}(t))+c_2r_{2j}(t)(x_j^{(g')}(t)-x_{ij}(t)) \quad (2)'$$

其中,$x^{(g')}(t)$ 是全局最优解 $x^{(g)}(t)$ 附近的一个解,$x_j^{(g')}(t)$ 是其第 j 维分量.通过阶段性地改变 $x^{(g')}(t)$,使算法搜索全局最优解 $x^{(g)}(t)$ 的附近区域.计算 $x^{(g')}(t)$ 采用如下算法:

用大小为 N 的一维数组 B_String 存储 $x^{(g)}(t)$ 的二进制编码,做:

① 令 $k_1=-1,k_2=0,k_3=0,Current\_Pos=0$;

② $k_2=k_2+1$,若 $k_2>N$,令 $k_2=k_2-1$,转⑤;否则,转③;

③ 若 $B\_String(k_2)=0$,则转②;否则,转④;

④ 若 $k_1=-1$,令 $k_1=k_2-(Current\_Pos+1)$,$Current\_Pos=k_2$,转②;否则,转⑤;

⑤ 令 $k_3=k_2-(Current\_Pos+1)$,转⑥;

⑥ 令

$$S=\{Current\_Pos-\lceil k_1/2\rceil,Current\_Pos-\lceil k_1/2\rceil+1,...,Current\_Pos-1,Current\_Pos,Current\_Pos+1,...,Current\_Pos+\lfloor k_3/2\rfloor\},$$

其中,$\lceil x\rceil$ 表示取大于或等于 x 的最小整数,$\lfloor x\rfloor$ 表示取小于或等于 x 的最大整数.采用轮盘赌(roulette wheel selection)方法[8]选出一个位置 $i\in S$,使得 $B\_String(i)=1$,且 $\forall j\in S-\{i\}$,令 $B\_String(j)=0$.为使靠近 $Current\_Pos$ 位置的比特位具有较大概率取 1,令 S 中任意一个元素 x 被选中的概率采用如下双边高斯(Gauss)函数计算:

$$f(x)=\begin{cases}\exp\left(-\dfrac{(x-c)^2}{\sigma_1^{\ 2}}\right), & \text{当}x\le c\\ \exp\left(-\dfrac{(x-c)^2}{\sigma_2^{\ 2}}\right), & \text{当}x>c\end{cases}.$$

其中,$\sigma_1=k_1/3$,$\sigma_2=k_3/3$,$c=Current\_Pos$.若 $k_2=N$,将 B_String 转化为 $x^{(g')}(t)$,算法结束;否则,令

$k_1 = k_3$, $Current\_Pos = k_2$,转②.

最后,为使算法具有爬山法的局部搜索特性,对于粒子的迭代公式中的参数确定如下:

(1) 取 $v_{\max} = 5\times(t/T)^{\kappa}$,其中 t 是算法的迭代次数,T 是设定的算法总迭代次数,κ 是曲率参数(本文取 κ=1.2);

(2) 取 w=1;c_1+c_2=2, $c_2 = 2/(1-\exp(-\theta\tau))$,其中,$\tau$ 随算法迭代次数 t 加 1 而加 1,同时每当算法重新计算 $x^{(g')}(t)$ 时,τ 被清零一次;θ 是曲率参数,它控制 c_2 由 1→2 变化的速率(本文取 θ=0.4).

混合算法.

① 初始化.将改进的贪心算法求出的问题解初始化为粒子 1,其他 L–1 个粒子采用随机方法产生.由于应该只有很少的数据点能被选中成为解集中的元素,所以对每一个粒子编码上的每一个位置以 0.2 的概率取 1,以 0.8 的概率取 0;计算粒子的适应度,取个体最优粒子等于初始粒子,取全局最优粒子 $x^{(g)}(t)$为适应度最高的粒子,计算 $x^{(g')}(t)$;取各粒子的初始速度为[–5,+5]之间的随机数;设迭代次数 t=0 和算法总迭代次数 T(本文取 T=300);

② 采用迭代公式(2)′和(3)更新各个粒子,计算粒子的适应度;

③ 若更新的粒子适应度高于前一代粒子适应度,则取各个体最优粒子等于更新的粒子;

④ 若更新的粒子中适应度最高的粒子的适应度高于全局最优粒子适应度,则取全局最优粒子 $x^{(g)}(t)$ 为更新的粒子中适应度最高的粒子,并计算 $x^{(g')}(t)$;

⑤ 若 $x^{(g')}(t)$ 连续 Y 代没有变化,则重新计算 $x^{(g')}(t)$ (本文取 Y=15),同时清零参数 τ;

⑥ 当一个粒子成为不活动粒子(即位置连续 E 代不发生变化的粒子,本文取 E=5)时,对其重新进行初始化;

⑦ 令 t=t+1;若 $t \le T$,转②;否则,输出全局最优粒子对应的准最小支配集,算法结束.

4 实验分析和结论

4.1 实验数据

本文以日本琦玉大学提取的 11 种矿泉水的味觉信号作为实验数据[9],取每种矿泉水的味觉信号各 100 组,原始数据包含 Na^+,K^+,pH,Cl^-,Ca^{++}这 5 个电极的输出信号,对原始数据进行主成分分析(principal component analysis)处理后,得到如图 2 所示的 2 维信号,横坐标 c_1 表示主成分 1,纵坐标 c_2 表示主成分 2.

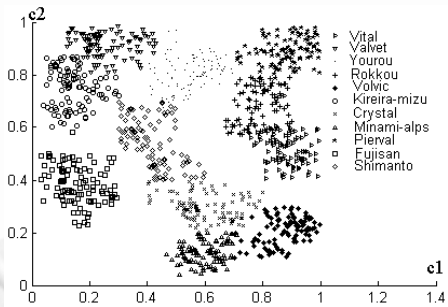


Fig.2 11 kinds of taste signals of mineral waters

图 2 11 种矿泉水的味觉信号

4.2 实验结果

在求解每一类数据的超球体类覆盖时,视该类数据点集合为 B,所有其他类数据点为集合 R.针对 11 种矿泉水的味觉信号使用本文的混合算法(参数集:种群大小 L=20, $v_{max}=5\times(t/T)^{1/2}$; w=1; $c_1+c_2=2$, c_2=2/(1−exp(−0.4τ));λ_1=0.55,λ_2=0.1,λ_3=0.15; $\alpha=0.96$, $\beta=0.01$, $\gamma=0.5$)、改进的贪心算法(参数: $\gamma=0.5$)、文献[7]的 BPSO 算法(参数集:种群大小 L=20,v_{max}=4;w=c_1=c_2=1;λ_1=0.55,λ_2=0.1,λ_3=0.15;α=0.96, $\beta=0.01$, $\gamma=0.5$)和遗传算法(参数集:种群大小 L=20,一致交叉概率 p_c=0.6,变异概率 p_m=0.005,λ_1=0.55,λ_2=0.1,λ_3=0.15; $\alpha=0.96$, $\beta=0.01$, $\gamma=0.5$)求解其超球体类覆盖.以 Shimanto 数据为例,应用 BPSO 算法和混合算法分别迭代 1 000 次和 300 次,适应度函数值变化情况如图 3 和图 4 所示.两种算法求出的 11 种矿泉水味觉信号类覆盖个数

K 和半径的样本标准差 Var 对比数据见表 1.

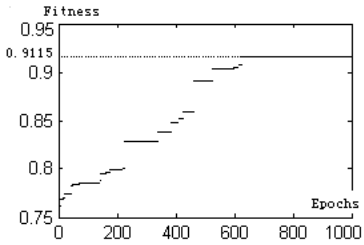


Fig.3 The varying curves of the global optimal particle fitness in BPSO algorithm for the Shimanto signals

图 3 全局最优粒子适应度函数变化曲线图(BPSO 算法,针对 Shimanto 数据)

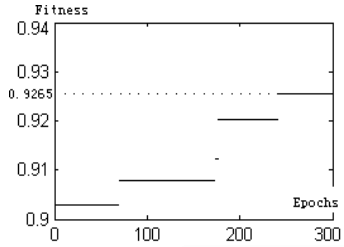


Fig.4 The varying curves of the global optimal particle fitness in Hybrid algorithm for the Shimanto signals

图 4 全局最优粒子适应度函数变化曲线图(混合算法,针对 Shimanto 数据)

Table 1 Comparison of the hyper-sphere class cover results from the compared algorithms for 11 kinds of taste signals of mineral waters

表 1 各种算法求解 11 种矿泉水数据的超球体类覆盖的结果对比关系表

	Hybrid algorithm		Improved greedy algorithm		BPSO algorithm		Genetic algorithm	
	k	*Var*	*k*	*Var*	*k*	*Var*	*k*	*Var*
Vital	10	0.001 947	13	0.000 987 4	10	0.002 677	11	0.002 023
Valvet	5	0.000 309	9	0.000 951 3	5	0.000 470	5	0.001 053
Yourou	9	0.000 051	11	0.000 519 9	8	0.000 319	9	0.000 411
Rokkou	23	0.000 343	20	0.000 375 9	28	0.000 624	25	0.000 846
Volvic	2	0.000 014	5	0.002 495 1	2	0.000 006	2	0.000 006
Kireira-Mizu	9	0.000 073	9	0.000 506 7	7	0.000 609	8	0.001 432
Crystal	10	0.000 786	11	0.000 052 3	16	0.000 490	12	0.001 551
Minmi-Alps	3	0.000 194	5	0.001 132 1	2	0.000 519	4	0.000 938
Pierval	14	0.000 308	10	0.000 443 2	16	0.000 574	15	0.000 883
Fujisan	2	0.000 002	4	0.003 217 1	2	0.000 009	3	0.001 203
Shimanto	11	0.000 243	10	0.000 648 2	11	0.000 458	10	0.000 363
Running time (s) (Epochs)	124.137 (300 epochs)		8.503		347.139 (1 000 epochs)		349.512 (1 000 epochs)	

注:实验环境为 P4CPU,512MRAM 笔记本电脑,Windows XP 系统,Matlab 6.5 软件包

由图 3 和图 4,在求解 Shimanto 数据的类覆盖过程中,经过 1 000 次迭代,BPSO 算法的全局最优粒子的适应度由 0.754 3 变化到 0.911 5,而混合算法仅用 300 次迭代,其全局最优粒子的适应度由 0.902 9(相当于改进的贪心算法求得的解的适应度)变化到 0.926 5.由表 1 可知,改进的贪心算法具有明显的速度优势,但求得的解质量较差;BPSO 算法和遗传算法求得的类覆盖个数 K 和标准差 Var 比较接近,运算时间也相差不大(迭代次数相同,均为 1 000 代);混合算法求得的解质量最好,尤其是标准差 Var 较其他算法有很大的改善,且运算时间为 124.137s (300 次迭代),而 BPSO 算法和遗传算法的运算时间分别为 347.139s 和 349.512s.因此,混合算法以较少的迭代次数求出了比其他算法更优的解.算法参数对实验结果是有影响的,本文仅取一组典型值,这些参数的确定并不困难,而且可以通过实验做进一步的调整.

5 结束语

本文提出了一种扩展的类覆盖问题和求解此扩展问题的混合算法,实验证明了算法的有效性.对于海量数据的快速类覆盖求解算法以及对于二进制粒子群优化算法的收敛性和收敛速率分析等都是进一步研究的课题.

References:

[1] Cannon A, Cowen L. Approximation algorithms for the class cover problem. Annals of Mathematics and Artificial Intelligence, 2004,40(3):215−223.

[2] Marchette DJ, Priebe CE. Characterizing the scale dimension of a high dimensional classification problem. Pattern Recognition, 2003,36(1):45−60.

[3] Priebe CE, Marchette DJ, DeVinney J, Socolinsky D. Classification using class cover catch digraphs. Journal of Classification, 2003,20(1):3−23.

[4] Parekh AK. Analysis of a greedy heuristic for finding small dominating sets in graphs. Information Processing Letters, 1991,39(5): 237−240.

[5] Kennedy J, Eberhart RC. Particle swarm optimization. In: Proc. of the 1995 IEEE Int'l Conf. on Neural Networks. Perth Western: IEEE Press, 1995. 1942−1948.

[6] Kennedy J, Eberhart RC. A discrete binary version of the particle swarm algorithm. In: Proc. of the 1997 Conf. on Systems, Man, and Cybernetics. Piscataway: IEEE Press, 1997. 4104−4109.

[7] van den Bergh F. An analysis of particle swarm optimizers [PH.D. Thesis]. Pretoria: Department of Natural and Agricultural Science, University of Pretoria, 2001.

[8] Wang ZZ, Bo T. Evolutionary Computation. Changsha: National University of Defense Technology Press, 2000. 26−37 (in Chinese)

[9] Huang YX, Zhou CG. Recognizing taste signals using a clustering-based fuzzy neural network. Chinese Journal of Electronics, 2005,14(1):21-25.

附中文参考文献:

[8] 王正志,薄涛.进化计算.长沙:国防科技大学出版社,2000.26−37.