

# 基于粒子群优化算法的数据流聚类算法

肖裕权,周肆清

(中南大学 信息科学与工程学院,湖南 长沙 410083)

**摘要:**针对当前基于滑动窗口的聚类算法中对原始数据信息的损失问题和提高聚类质量和准确性,在现有基于滑动窗口模型数据流聚类算法的基础上,提出了一种基于群体协作的粒子群优化算法(PSO)的新数据流聚类算法。这种优化的新数据流聚类算法利用改进的时间聚类特征指数直方图作为数据流的概要结构以及应用 PSO 在聚类过程中对聚类质量的局部迭代优化。实验结果表明,此方法有效减少了内存的开销,解决了对原始数据信息损失的问题。与传统的数据流聚类算法相比,基于粒子群优化算法的数据流聚类算法在聚类质量和准确性上明显优于传统的数据流聚类算法。

**关键词:**聚类;数据流;粒子群优化算法;滑动窗口

**中图分类号:** TP311

**文献标识码:** A

**文章编号:** 1673-629X(2011)10-0043-04

## Clustering Evolving Data Streams Based on Particle Swarm Optimization

XIAO Yu-quan, ZHOU Si-qing

(School of Information Science and Engineering, Central South University, Changsha 410083, China)

**Abstract:** In view of the current based on sliding windows clustering algorithm of original data information loss problem and improve the cluster quality and accuracy, in the existing basis for data flow clustering algorithm based on sliding window model, proposed based on group collaboration of particle swarm optimization algorithm (PSO) of new data flow clustering algorithm, the optimization of new data flow clustering algorithm by means of improved time clustering indexes as data flow histogram summary of structure and the cluster quality of local iterative optimization in clustering process using the PSO. The experiment results show that this method is effective to reduce the memory spending, and solved the problem of original data loss. Compared with the traditional data flow clustering algorithms, based on the particle swarm optimization algorithm of data flow clustering algorithm evidently excels the traditional flow of data clustering algorithm in cluster quality and accuracy.

**Key words:** clustering; data streams; particle swarm optimization; sliding window

## 0 引言

作为一种强有力的信息处理方法,聚类分析越来越多地被应用于数据流挖掘环境下。目前,有基于划分、层次、模型等<sup>[1,2]</sup>的多种传统的经典聚类算法,然而在数据流环境下,元组随着时间的变化而变化,导致所形成的“簇”也在不断变化。因此传统聚类算法不再适合处理数据流环境下的聚类分析。现有的数据流聚类算法主要以2003年 Aggarwal C 等<sup>[3]</sup>提出的 CluStream 算法为核心代表,它是一种基于 k-means 的数据流聚类算法,它把数据流看成一个随时间变化的过程,而不是一个整体进行聚类分析。这种处理方式简单且直

接,在数据流随时间变化较大时该算法比其它算法产生更高质量的聚类。但是算法的不足之处在于,子聚类的半径随着数据的流入而不断增大,增加了处理代价;再者由于没有在线淘汰“旧数据”,因此,聚类结果并不能体现当前数据流的真实情况。最近 Aggarwal C<sup>[4]</sup>对高维不确定数据流聚类提出了新的算法,这对 CluStream 算法在不确定数据流聚类中的不足做了很好的补充。文中在 CluStream 算法的基础上,通过引入粒子群优化算法,提出了一种新的聚类算法。该算法使用滑动窗口模型,通过改进时间聚类特征结构和时间聚类特征指数直方图来解决 CluStream 算法存在的不足。算法降低了计算复杂度,从而提高了算法的计算效率及计算质量。

收稿日期:2011-02-27;修回日期:2011-05-09

基金项目:湖南省科技厅软件课题(2009ZK3046)

作者简介:肖裕权(1983-),男,湖南永州人,硕士研究生,研究方向为数据库技术、数据流;周肆清,副教授,研究方向为数据库技术、网络信息处理、中间件技术。

## 1 相关概念和定义

### 1.1 滑动窗口模型

在滑动窗口模型<sup>[5,6]</sup>中,数据流可被看作一个不

断增长的多维元组数据项  $\langle x_1, t_1 \rangle \cdots \langle x_i, t_i \rangle \cdots$  组成的无限集合, 其中,  $\langle x_i, t_i \rangle$  是多维元组项的表示形式,  $x_i$  表示数据流中的样本点,  $t_i$  表示该样本点进入滑动窗口时间。对任意  $i (i \geq 1)$ ,  $x_i = (x_i^1 \cdots x_i^d)$ 。假设滑动窗口的大小为  $N$ , 对于任意时间点  $t_i$ , 在滑动窗口模型中只考虑并处理最近到达的  $N$  个元组, 其它数据全部忽略不计。假设数据流传输方向自左向右, 滑动窗口模型如图 1 所示。

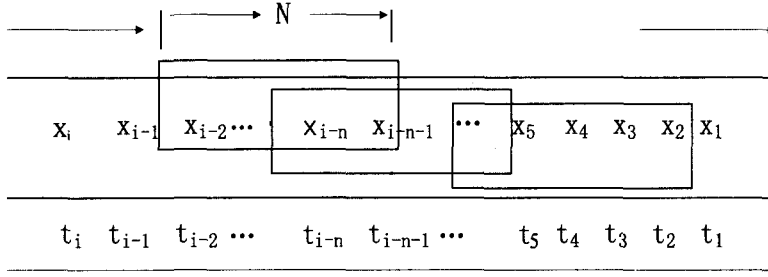


图 1 滑动窗口模型

## 1.2 改进的聚类特征

指数直方图技术是用来生成基于滑动窗口模型的概要数据结构的一种有效方法。聚类特征指数直方图大多都是按照元素的到达次序来构建桶(bucket)。常建龙等人<sup>[6]</sup>提出纳伪和拒真两种聚类特征指数直方图作为概要结构, 较好地保存了数据流当前窗口内的数据分布状况, 从而获取较高质量的聚类结果。文中在现有基础上对聚类特征指数直方图的维护过程加入一种时间淘汰机制, 改进的聚类特征指数直方图根据元组的到达时标将元组划分为若干个桶, 并淘汰掉过期元组, 使概要数据结构更具有时效性和可用性。

定义 1 一个基于滑动窗口模型的  $d$  维  $n$  元组数据流片段  $DS^d = \{ \langle x_n, t_n \rangle, \cdots, \langle x_1, t_1 \rangle \}$  的改进时间聚类特征(improvement temporal cluster feature, TCF) 为  $2d+3$  维的五元组向量:

$$TCF(DS^d) = (CF2^*, CF1^*, t, \bar{t}, n) \quad (1)$$

在式(1)中,  $CF2^*$  为某个 TCF 中各  $d$  维元组的向量平方和, 即  $\sum_{i=1}^n (\vec{x}_i)^2$ ;  $CF1^*$  是各元组的向量线性和, 即  $\sum_{i=1}^n \vec{x}_i$ ;  $t$  为改进时间聚类特征中最近元组的时标  $t_n$ ;  $\bar{t}$  为时间聚类特征中各元组时标  $t_i$  平均值。  $n$  是时间聚类特征中包含的元组个数。

这个改进时间聚类特征是对文献[3]中聚类结构的另一种形式化描述, 它同样具有聚类结构可加性的性质。将一组元组  $C$  的改进时间聚类特征记为  $TCF(C)$ 。那么有性质 1 中所述的可加性。

性质 1  $TCF(C_1 \cup C_2)$  可由  $TCF(C_1)$  和  $TCF(C_2)$  进行组建,  $C_1$  和  $C_2$  为两组元组集合。

证明: 由定义 1 知,  $TCF(C_1 \cup C_2)$  中的  $CF2^*$ ,  $CF1^*$  和  $n$  项为  $TCF(C_1)$  和  $TCF(C_2)$  中的对应项累加和,  $t$  项为  $TCF(C_1)$  和  $TCF(C_2)$  中  $t$  的最大值,  $\bar{t}$  项则由  $TCF(C_1)$  和  $TCF(C_2)$  中  $\bar{t}$  值的平均值获得。

定义 2 改进聚类特征指数直方图(improvement exponential histogram of cluster feature, EHCF)是改进时间聚类特征 TCF 的集合。集合中的各个改进时间聚类特征分别基于如下各组元组所构建<sup>[6]</sup>。对于上文滑动窗口中的多维元组数据项  $\langle x_i, t_i \rangle$ , 且当  $j < m$  时  $t_j < t_m$ 。这些元组根据到达的先后次序, 可划分为若干组, 记作  $G_1, G_2, \dots$ , 且满足如下约束条件:

当  $i < j$  时, 组  $G_i$  中的所有元组都比组  $G_j$  内的元组先到达。

令  $S(G)$  为组  $G$  的大小, 则  $S(G_i) = 1$ 。对任意  $i (i > 1)$ ,  $S(G_i)$  可取任意正整数  $i$ 。

组  $G_i$  的下标  $i$  为组  $G_i$  的级别, 对  $G_i$  所构建的 TCF 称为  $i$  级 TCF。除了最高级别的组以外, 各级别的元组均小于等于  $\left\lceil \frac{1}{\varepsilon} \right\rceil$ , 其中,  $\varepsilon$  为用户指定的误差参数, 且  $0 < \varepsilon < 1$ 。

$TCF(G_i)$  的  $t$  项为当前有效时标。

定义 3 设有级别为  $i$  的  $n$  个改进时间聚类特征  $TCF(G_{i1}), TCF(G_{i2}), \dots, TCF(G_{in})$ , 其  $\bar{t}$  值分别为  $\bar{t}_1, \bar{t}_2, \dots, \bar{t}_n$ , 现将其合并为  $n-1$  个改进时间聚类特征, 可按如下原则合并: 设  $\bar{t}_k = \bar{t}_k - \bar{t}_j$ ,  $n \geq k > j \geq 1$  且  $k = j + 1$ , 如果有  $\bar{t}_k = \min\{\bar{t}_j\}$ , 则  $TCF(G_p)$  和  $TCF(G_q)$  合并成新的 TCF。

定义 4 当多维元组数据项  $\langle x_i, t_i \rangle$  到达时, 改进时间特征指数直方图采用如下步骤进行增量式维护:

对于新到达的元组数据项  $\langle x_i, t_i \rangle$ , 由定义 1 生成一个新的 0 级  $TCF(C)$ , 其中  $C$  为仅包含  $\langle x_i, t_i \rangle$  的数据集。

将新生成的  $TCF(C)$  加入到改进时间聚类特征指数直方图中。若存在  $\left\lceil \frac{1}{\varepsilon} \right\rceil + 1$  个 0 级 TCF, 则按定义 3 进行两个 TCF 合并, 被淘汰的最老 TCF 变为新的 1 级 TCF。则 0 级 TCF 的个数最多为  $\left\lceil \frac{1}{\varepsilon} \right\rceil$ 。从级别  $i=1$  开始, 若  $i$  级别 TCF 的个数为  $\left\lceil \frac{1}{\varepsilon} \right\rceil + 1$ , 则继续这种合并

过程,直到某级别中 TCF 的个数小于等于  $\left\lceil \frac{1}{\varepsilon} \right\rceil$  为止。

## 2 基于粒子群优化算法的改进聚类分析

粒子群算法,也称粒子群优化算法(Partical Swarm Optimization),缩写为 PSO,是一种进化计算技术,最早由 Kennedy 博士和 Eberhart 博士提出并发现是一种良好的优化工具<sup>[7]</sup>。近年来出现了很多基于粒子群优化算法的改进算法<sup>[8-9]</sup>以及基于粒子群优化算法的应用<sup>[10-11]</sup>。粒子群由许多粒子组成,所有的粒子都有一个由被优化的函数决定的适应值,每个粒子还有一个决定它们飞翔的速度  $v_i(t)$  和位置  $l_i(t)$ 。算法在每一次优化迭代中,粒子通过跟踪两个“极值”来更新自己。一个是粒子本身所找到的最优解,这个解叫做个体极值  $p_i(t)$ ,另一个是粒子群中目前的最优解,称为全局极值  $g(t+1)$ 。在找到这两个最优值时,每个粒子根据如下的公式来更新自己的速度和位置:

$$v_i(t+1) = \varepsilon v_i(t) + c_1 r_1 (p_i(t) - z_i(t)) + c_2 r_2 (g(t) - z_i(t)) \quad (2)$$

$$z_i(t+1) = z_i(t) + v_i(t+1) \quad (3)$$

其中,如果  $(f(z_i(t+1)) < f(p_i(t)))$ ,则  $p_i(t+1) = p_i(t)$ ;反之,则  $p_i(t+1) = z_i(t+1)$ ;对一切  $i$  有,  $g(t+1) = \min\{p_i(t+1)\}$ 。

式中,  $\varepsilon \geq 0$  是惯性权重。  $r_1, r_2$  是介于  $(0,1)$  之间的随机数。  $c_1, c_2$  是学习因子,通常  $c_1 = c_2 = 2$ 。在每一维粒子的速度都会被限制在一个最大速度  $v_{\max}$ ,如果某一维更新后的速度超过用户设定的  $v_{\max}$ ,那么这一维的速度就被限定为  $v_{\max}$ 。

在基于粒子群算法的聚类分析中,如果要把元组数据聚类成  $K$  个类,那么每个粒子就代表这  $K$  个类的中心。设  $z_i = (c_{i1}, c_{i2}, \dots, c_{ik})$ ,其中  $c_{ij}$  代表第  $i$  个粒子的第  $j$  类中心点坐标向量。文中采用如下适应度函数来对分类方案优劣进行评价:

$$f(z_i) = \frac{\varepsilon_1 d_1(z_i)}{1 + \varepsilon_2 d_2(z_i)} \quad (4)$$

$$\text{类间距为: } d_1 = \frac{2 \sum_{i=0}^k \sum_{j=i+1}^k d(c_i, c_j)}{k(k-1)} \quad (5)$$

$$\text{类内距为: } d_2 = \frac{1}{k} \sum_{i=0}^k \frac{\sum_{j=0}^{m_i} d(c_{ij}, c_i)}{m_i} \quad (6)$$

其中,  $c_i, c_{ij}$  和  $m_i$  分别是第  $i$  类中的聚类中心、第  $j$  个元组和元素个数。  $\varepsilon_1$  和  $\varepsilon_2$  是用户指定的类间距与类内距的权重,可以用来调节函数  $f(z_i)$  值的范围,从而使分类方案同时满足类内距和类间距对分类不同影响的需求。

文中的聚类算法是对聚类算法 CluStream 进行的

改进,整个过程大致分为维护 EHCF 结构、使用基于粒子群优化算法的数据流聚类算法进行初始聚类、生成聚类结果。在算法中,由用户提供一个迭代次数  $T_{\text{psa}} > 0$ ,每当有数据流入时,使用粒子群优化算法对 EHCF 进行初始分类,这样可以解决滑动窗口模型中对历史数据无记忆的效果,同时又减少了内存开销,当数据量少时,该算法也获得了相当好的聚类质量。算法简称为 CluPSO,其中包含如下参数:  $DS^d$  为待处理的数据流;  $\varepsilon (0 < \varepsilon < 1)$  为误差参数;  $k$  为中心聚个数;  $M$  为 EHCF 的最大个数。  $W_{\text{psa}}$  为基于粒子群优化算法的相关参数。

算法描述如下:

CluPSO( $DS^d, \varepsilon, k, M, W_{\text{psa}}$ )

Begin

(1) 设定一个开始时间,让数据流入滑动窗口;初始化  $\varepsilon$  等相关参数;

(2) 如果  $T_{\text{psa}} > 0$ ,则随机初始化每个粒子的位置  $z_i(t)$  和速度  $v_i(t)$  向量,其值为由  $k$  个聚类中心构成的多维向量;

(3) 从数据流  $DS^d$  中读取元组数据  $x_i$ ;

(4) 使用定义 1 生成新的 TFC,并计算 TFC 中的各项值,使用定义 4 对 EHCF 结构进行增量式维护;

(5) 更新滑动窗口内的数据流;

(6) 如果 EHCF 中有元组过期且某个桶中所有的 EHCF 均过期,则删除该桶;

(7) 如果有  $T_{\text{psa}} > 0$ ,则从  $t_{\text{psa}} = 1$  到最大迭代次数  $T_{\text{psa}}$ ,进行如下(8)~(11)的  $T_{\text{psa}}$  次循环;

(8) 对每个粒子  $z_i(t)$ ,计算待分类 EHCF 集合到粒子对应的  $k$  个中心距离,根据距离将 EHCF 进行初始分类;

(9) 由分类结果根据式(4)计算出粒子的适应度  $f(z_i(t))$ ;

(10) 根据适应度值计算粒子个体最优  $p(z_i(t))$  和群最优  $g(t)$ 。

(11) 通过粒子群优化算法的位置  $z_i(t)$  和速度  $v_i(t)$  更新式(2)和(3)式中的位置和速度。

(12) 如果有聚类需求,采用 k-means 算法对 EHCF 进行聚类。生成聚类结果;

(13) 如果数据流  $DS^d$  中还有数据,重复步骤(2)~(13);

End

## 3 实验分析

所有实验在一台配置为 Intel 酷睿 i3 530,1G 内存,WindowsXP 操作系统的 PC 上进行。算法采用 JAVA、SQL Server2005 实现。为了实验数据可比性和有

效性,采用文中多次提到的 CluStream 算法作为比较算法。因为 CluStream 算法使用 KDD-CUP'98 Charitable Donation 数据<sup>[12]</sup>来进行算法的测试,同时该数据在数据流聚类实验中曾被多次使用过,所以本实验也采用了该数据。该数据记录了 95 412 条慈善捐赠信息,并且把有类似捐助行为的捐助者进行聚类。跟 CluStream 算法一样,只用每条记录中的 56 个信息进行测试,根据数据输入的次序作为流的次序,时间作为流的时标值。

聚类质量采用聚类算法中最常用的距离平方和 SSQ<sup>[13]</sup>来衡量。其判断标准为:SSQ 值越小,则算法聚类质量越好。在实验中,CluPSO 中的 EHCF 最大个数对应于 CluStream 中的最大微簇个数。参数设置如下:误差参数  $\varepsilon = 0.1$ , 滑动窗口大小  $N = 10000$  或 30000,半径相对阈值  $\beta = 2$ ,初始聚类迭代次数  $T_{ps0} = 100$ ,聚类中心个数为 20 或 100,其它参数取默认值。

对于以上数据,在实验中对 EHCF 合并过程中所需维护的 TCF 结构数如图 2 所示,它采用两个相邻 TCF 间的时间差取小来合并 TCF,通过升级最老 TCF 来维护 EHCF 和淘汰旧数据。经分析可以知道本文算法使用的 TCF 个数最大值为  $\frac{1}{\varepsilon} \log[\varepsilon N + 1]$ 。 $N$  为滑动窗口的最大长度,因此本文算法在空间复杂度上要优于现有的滑动窗口聚类算法。

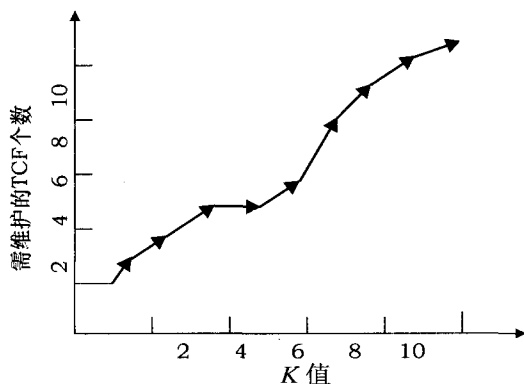


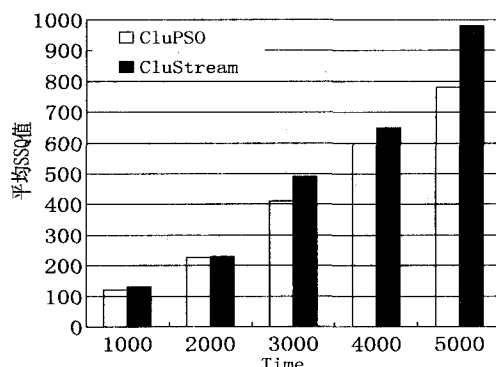
图 2 需维护的时间特征结构数

所有的实验结果表明,CluPSO 算法比 Clustream 算法更稳定并有较好的聚类质量。图 3 给出了随机抽取的两次实验的 10 组实验结果,从图中两种算法的 SSQ 值来看,CluPSO 算法要优于 Clustream 算法。在不同时间的平均 SSQ 值,都比 Clustream 的要小。

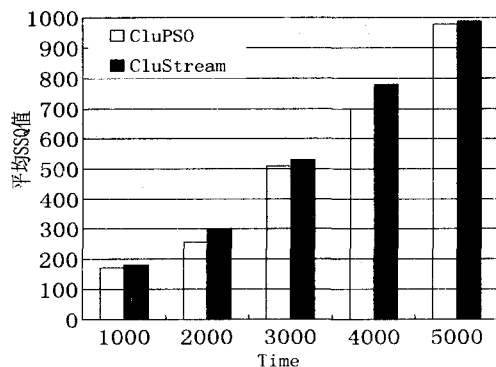
#### 4 结束语

理论分析及实验表明,滑动窗口模型下基于粒子群优化算法的数据流优化聚类算法提高了聚类质量,在聚类质量上优于界标模型下的算法,同时,在计算复杂度上优于现有的一些算法。今后的工作是将算法应

用到网络入侵等更多数据流应用中,根据不同数据流环境改进算法的不足之处。



(a) 聚类大小  $K = 20$ , 窗口大小  $N = 10000$



(b) 聚类大小  $K = 100$ , 窗口大小  $N = 40000$

图 3 聚类质量比较

#### 参考文献:

- [1] McQueen J. Some Methods for Classification and Analysis of Multivariate Observations [C]//In Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability. [s. l.]:[s. n.],1967:281-297.
- [2] Guha S, Rastogi R, Shim K. ROCK: A Roust Clustering-A Filter Solution [C]//In Proceeding of the 2nd IEEE International Conference on Data Mining (ICDM'02). Maebashi City, Japan:[s. n.],2002:15-122.
- [3] Aggarwal C, Hart J, Wang J, et al. A framework for clustering evolving data streams [C]// In: VLDB 2003. [s. l.]:[s. n.],2003:81-92.
- [4] Aggarwal C C. On high dimension projected clustering of uncertain data streams [C]// In: Proc. of the 25th Int'l Conf. on Data Engineering. Shanghai:IEEE,2009:1152-1154.
- [5] 张忠平,王浩,薛伟,等.动态滑动窗口的数据流聚类方法[J].计算机工程与应用,2011,47(7):135-138.
- [6] 常建龙,曹锋,周傲英.基于滑动窗口的进化数据流聚类[J].软件学报,2007,18(4):905-918.
- [7] Mitra S K, Murthy C A, Kundu M K. Technique for fractal image compression using genetic algorithm [J]. IEEE Transactions on Image Processing,1998(4):586-593.
- [8] Jang W S, Kang H I, Lee B H, et al. Optimized fuzzy cluste-

(下转第 50 页)

SQL Server 扩展存储过程 API 编写的,可直接在 SQL Server 实例的地址空间中运行。它可以存取数据库中的数据(通过 ODBC,OL EDB,DAO,ADO 等),可以实现复杂的分析功能(如图像分析、几何运算等),也可以实现硬件通讯等等。

在本例中使用 SQL Server 调用上面生成的 DLL,首先使用系统存储过程 sp\_OACreate 来初始化一个该类的实例,然后利用 sp\_OAMethod 来调用实例的相应方法。

给出代码如下:

```
declare @ hr int,@ src varchar(255),@ desc varchar(255)
declare @ obj int,@ StartResult int
exec @ hr=sp_OACreate 'ClassCanbus.CanOperate', @ obj out
if @ err<>0 goto Err --如果创建失败,则进行错误处理
exec @ hr=sp_OAMethod @ obj, 'StartCan', @ StartResult out,
0,0
if @ err<>0 goto Err --如果调用错误,则进行错误处理
exec sp_OADestroy @ obj
return
Err:
exec sp_OAGetErrorInfo 0,@ src out,@ desc out
select cast(@ err as varbinary(4)) as 错误号,@ src as 错误源,@ desc as 错误描述
```

### 3.3 对于上述实现的补充

实际中,为了防止客户端对硬件操控的并发请求造成总线冲突,我们往往还需要采用数据中某个特定字段作为标志位(semaphore),采用锁机制和 ID 号排序的方式来保证对总线的序列访问。并且还要考虑到控制过程可能发生的故障,做好容错处理和收集反馈错误信息。此外,针对网络扩展和网络负载变化引起的信息传输时延,可以考虑使用时分复用技术来改进 CAN 总线的实时性<sup>[11,12]</sup>。

## 4 结束语

文中针对 C/S 结构的现场总线系统中,进行硬件控制系统拓展时面临的复杂度和成本问题,提出了一

种通过 SQL Server 的扩展存储过程来实现硬件控制的方法,并给出了具体的实现过程。

采用文中方法设计的系统已经在实际中应用,并发挥了良好的效果。为使用单位节约了大量的硬件开销,减少了控制总线的设计和部署的困难,取得了良好的经济效益。

### 参考文献:

- [1] Microsoft Corporation. Microsoft SQL SERVER BOOK ONLINE [DB/OL]. [http://msdn.microsoft.com/en-us/library/ms130214\(v=sql.90\).aspx](http://msdn.microsoft.com/en-us/library/ms130214(v=sql.90).aspx).
- [2] 刘 慧. 基于 C/S 与 B/S 的煤矿安全监察信息管理系统研究[D]. 西安:西安科技大学,2005.
- [3] Sunderic D. Microsoft SQL Server 2005 Stored Procedure Programming in T-SQL & .NET[M]. [s. l.]: McGraw-Hill Osborne Media,2006.
- [4] 邬宽明. CAN 总线原理和应用系统设计[M]. 北京:北京航空航天大学出版社,1996.
- [5] 谢子青. 光电隔离抗干扰技术及应用[J]. 现代电子技术,2003(13):33-37.
- [6] 致远电子. CAN 节点设计[DB/OL]. <http://www.zlgmcu.com/download/downs.asp?ID=2022>.
- [7] 致远电子. CAN-BUS 通用测试软件及接口函数库使用手册[DB/OL]. <http://embedcontrol.com/download/downs.asp?ID=776>.
- [8] 江晋剑. 基于 AT89C55 和 CAN 总线的智能节点设计[J]. 计算机技术与发展,2008,18(4):214-216.
- [9] 姜兴刚. 基于 PCI 总线的 CAN 通信卡的设计与实现[J]. 测控技术,2004,23(7):42-43.
- [10] 万 波. SQL SERVER 扩展存储过程实现机制及应用方法初探[J]. 武汉科技大学学报(自然科学版),2001,24(3):294-297.
- [11] 韩进军. CAN 总线实时性研究及改进[J]. 计算机测量与控制,2005,13(11):1247-1249.
- [12] Tindell K W. Analysing Real-Time Communications: Controller Area Network (CAN)[J]. IEEE Real-Time Systems Symposium,1994,23(7):259-263.

(上接第 46 页)

- ring by predator prey particle swarm optimization[C]// IEEE Congress on Evolutionary Computation. [s. l.]: [s. n.], 2007:3232-3238.
- [9] Jarbou I B, Cheikh M, Siarry P, et al. Combinatorial particle swarm optimization (CPSO) for partitional clustering problem [J]. Applied Mathematics and Computation,2007,192(2):337-345.
- [10] 张爱华,江中勤,张 华. 基于粒子群优化算法的分形图像压缩编码[J]. 计算机技术与发展,2010,20(2):21-24.

- [11] 张家柏,王小玲. 基于聚类和二进制 PSO 的特征选择[J]. 计算机技术与发展,2010,20(6):905-918.
- [12] Aggarwal C, Han J, Wang J, et al. A Framework for Clustering Evolving Data Streams[C]// Conference on Very Large Data Bases. Berlin:[s. n.],2003:312-323.
- [13] Aggarwal C, Han J, Wang J, et al. A Framework for Projected Clustering of High Dimensional Data Streams[C]// Proceeding of the 30th very large data bases conference. Toronto:[s. n.],2004:852-863.