基于SOM聚类的软构件分类方法*)

王 卓 刘大昕 冯晓宁
(哈尔滨工程大学计算机科学与技术学院 哈尔滨 150001)

摘 要 软构件刻面分类法是一种被各大软构件库系统广泛采用的分类方法,但是传统的刻面分类法需要人工建立和维护庞大的术语空间,增大了软构件建库和入库的工作量。利用基于SOM神经网络的聚类技术可实现无需建立术语空间的软构件自动分类,同时针对软构件的特点和SOM聚类的需要预先确定拓扑结构和聚类结果与输入样本的次序有关等缺点,对SOM聚类的训练过程进行改进以满足软构件聚类的要求。

关键词 软件复用,软构件,刻面分类,聚类分析,SOM神经网络

Software Component Classification Based on Improved SOM Clustering

WANG Zhuo LIU Da-Xin FENG Xiao-Ning
(College of Computer Science and Technology, Harbin Engineering University, Harbin 150001)

Abstract Faceted classification is a popular software component classification adopted by many software component repository systems. However, it needs artificially establish term space of each facet, which increases the workload of establishing the software component repository and the workload of inserting components into the repository. Based on SOM clustering, an automatic component classification is provided to free faceted classification from term space. SOM clustering has two disadvantages such that their topology construction need be defined in advance and the cluster result is disturbed by order of learning samples. So the training process of SOM clustering is improved to increase the accuracy of software component classification.

Keywords Software reuse, Software component, Faceted classification, Clustering analysis, Self-organizing feature map neural networks

随着对软件复用的研究与深入,软构件库作为软件复用的一项重要基础设施,已得到产业界与学术界越来越多的重视。在构件库系统中,无论是构件的存储、检索,还是构件信息的反馈,每一项活动都依赖于构件系统化的组织方式。构件的分类方式允许构件管理者将构件组织成一种有利于用户检索的结构。同时,分类方式还决定了检索的精确性和返回结果的准确性[1]。在众多分类方法中,软构件的刻面分类法以其它能够较大地提高检索效率,而且有助于复用者理解构件和目标领域的优点得到软构件库的广泛应用。例如,RE-BOOT系统[2]和NATO[3]系统都采用各自定义的刻面对软构件进行分类和描述,北大青鸟的软构件库[4]系统也采用以刻面分类法为主与其它方法相结合的分类策略。

构件库系统的刻面分类检索法是Prieto-Diaz和Freeman在1987年提出的,这种方式通过反映构件本质特性的视角(刻面)对构件进行精确的分类。一个刻面分类模式(faceted scheme)由一组描述构件本质特征的刻面组成,每个刻面从不同的侧面对构件库中的构件进行分类。每个刻面由一组术语(term)构成,称为术语空间。术语是刻面的值,是构件的属性。它表示了构件某一侧面的一种划分,术语空间应涵盖刻面所有的可能方面。术语是借鉴领域专家的建议,选取领域中特定的专业词汇,且充分考虑某一刻面的所有可能,含义相近的构件属性值可作为术语的同义词出现[4]。由于建立术语空间需要专家经验因此传统的刻面分类法都是人工建立和维护术语空间,这就制约了描述构件的刻面值和描述构件检索条件的灵活程度,同时也给用户跨越多个软构件库进行查询造成了障碍。

本文为了解决这个问题将聚类分析技术应用到刻面分类法中,使其摆脱术语空间的束缚,并且在分类过程中将构件之间的关系作为考虑因素,简化构件库建库程序提高检索效率。同时,针对SOM聚类在软构件分类中表现出的不足进行改进,改进后的SOM聚类算法具有拓扑结构可变,聚类结果与样本输入次序无关的特性。

1 软构件的表示

传统的刻面分类法在每一刻面下用自然语言的单一词汇来描述软构件,这限制了描述语义的丰富性和准确性[6,7]。另外,传统的方法将软构件看作彼此孤立的个体,而在实际运行系统中软构件是彼此联系,相互协作的实体。因此这里采用多元组将刻面值与构件之间相互关系相结合来表示软构件,以便聚类时将构件之间的关系作为聚类的考虑因素之一,处理聚类边界分割时根据样本构件与各聚类中构件之间关系的密切程度来确定样本构件归属的聚类。在表示构件之前首先需要定义构件之间的 n 种相互关系 $R_1, R_2, \cdots, R_n$,这里的关系是指在实际系统运行时软构件之间的相互关系。例如,

*)基金项目:国家电子信息产业发展基金(基于斯达模式的中小企业管理信息化软件系统);黑龙江省发展信息产业专项资金项目(FX-01-054);哈尔滨市科技攻关计划项目(0111211118)。王 卓 博士研究生,主要研究领域为软构件,工作流,CIMS。刘大昕 教授,博士生导师,主要研究领域为数据库与知识库,软件工程,企业信息化。冯晓宁 硕士研究生,主要研究领域为Petri网理论及应用,软件工程。

可以定义软构件之间有六种关系：数据传输关系，程序调用关系，网络传输关系，业务流转关系，消息传递关系和服务共享关系。下面给出构件的多元组定义。

定义 1 构件在每个刻面下定义为一个多元组：

$component=\{Discription,C_1,C_2,\cdots,C_n\}$

式中：$Discription$ 表示该构件在此刻面下的值，同传统的刻面分类法不同的是这里的刻面值由一段自然语言文本描述而不是单一词汇。

C_1 表示与该构件具有 R_1 关系最多的构件类别的名称，也就是说根据统计量在此刻面下与该构件之间拥有 R_1 关系的软构件大多属于 C_1 类别中，如果不存在于该构件之间具有 R_1 关系的构件，C_1 的值记为“0”。$C_2,C_3,\cdots,C_n$ 的含义以此类推。

作为软构件的表示形式多元组中的各项均不能为空，$C_1,C_2,\cdots,C_n$ 的值可以重复。例如，名为采购单的软构件可表示为采购单={“民用企业信息化领域采购业务中的采购单生成和采购信息管理”，“库存”，“采购”，“生产”，“采购”，“采购”，“生产”}。

2 基于 SOM 聚类的分类方法

2.1 构件分类树及分类策略

根据软构件分类的先验知识，将分类问题归为一些简单的基本类并构成树型结构的分类器，将大类别问题逐级分解为一组小类别问题。这样每一基本类中所包含的类别数目将有所减少，由此控制利用神经网络进行聚类时输出神经元的数量，从而避免当 SOM 网络的输出神经元数目 N 增大时，神经网络的训练时间将以指数增长。在每一刻面下建立构件类别的分层次树型结构，称作构件分类树。

定义 2 构件分类树 $CT=(N,L)$，N 是结点，L 是有向边。其中，分类树的结点 N 对应某个刻面分类方案在同一刻面中构件的类别和子类别，$N=(name,discription,leftchild,rightbrother)$，$name$ 是构件类别名称，$discription$ 是构件类别的描述文本，$leftchild$ 是该结点最左边的子结点，$rightbrother$ 是该结点的右边兄弟结点。分类树的边 $L=\{N_1\rightarrow N_2 \mid N_1,N_2\in N;N_2$ 是 N_1 的子类别$\}$。

分类树的根结点对应整个软构件库这个最大的类别，叶子结点对应软构件的最小子类别。分类树中的每一个分支表示一次在同一刻面下构件的子集划分过程。需要说明的是，这里分类树中每一个节点对应的是在同一刻面下构件的不同层次的类别而不是刻面与子刻面，其区别在于刻面是描述构件的不同的侧面而类别是在同一侧面下构件的不同分类。

在基于 SOM 聚类的软构件分类方法中，刻面的定义方法与传统的刻面分类法相同，不同的是不需要再建立术语空间，只需要在每一刻面下初始化构件类别树，定义树型结构和分支的数目，并给出每一节点对应类别的描述文本。在分类法的训练过程中从根节点的下一层开始在每一层中以同一层的所有节点作为初始输出神经元，以其父亲节点对应的类别中包含的全部构件作为样本应用 SOM 聚类算法完成子集划分过程。同时在分类法的训练过程中改进的 SOM 聚类算法可以根据构件的刻面值描述文本与类别描述文本之间的近似程度判断是否需要动态生长出新的神经元，相应的树中也可以动态插入新的类别节点，新的神经元对应的类别节点同原有的神经元所对应的类别节点在同一层次上，或者说它们具有相同的父节点。例如在“应用领域”刻面下的构件分类中使用树型结构，初始情况下第一层分为民用、军用和其他 3 类，第二层民用这一分支又分为信息管理、图形显示、通用工具、网络通信 4 类，理论上构件的分类数目减少了 2/3。在第二层的训练过程中聚类算法动态生长出一个神经元对应类别地理信息系统(GIS)因此第二层被插入一个新的节点“GIS”同前面提到的 4 类共同作为“民用”的子节点。

2.2 对 SOM 聚类的改进

传统的 SOM 聚类的缺点是当训练样本的数目较少时，算法的结果与训练模式输入的次序有关；预定的网络拓扑结构隐含了对结果映射的限制，通常只有在训练结束之后才能发现不同的网络拓扑结构也许能得到更好的结果，而这种预先确定结构的 SOM 网络带来了诸如神经元欠利用，网络映射欠准确以及边缘效应等问题，这些问题影响了软构件自动分类的准确率。

SOM 网络在训练过程中权值修改公式一般形如 $\Delta w_{ij}(t+1)=\eta(t)\times N_g(t)\times\triangle w_{ij}(t)$，其中时间函数 $\eta(t)$ 随着样本的输入线性递减，以便权值的修改量随样本的输入而缩小从而保证算法收敛，但是这就造成了网络的聚类结果同输入样本的次序有关的问题。另外，阈值函数 $N_g(t)$ 一般情况下也采用随样本输入线性减小的函数，这样就缩小了后续样本的影响范围，同样也造成了对后续样本的不公平。为解决这一问题，在这里定义 $N_g(t)$ 为 0 到 $\pi/2$ 之间的正弦函数，这样 $N_g(t)$ 在 0 到 $N_g(0)$ 之间递增最后达到预定的状态。这时神经元调整的阈值范围随着输入样本逐渐扩大直至预定状态，后续样本虽然对权值的改变量被减小但是可修改的神经元范围扩大了，因此提高了各神经元被后续样本修改的频率，从而提高了后续样本对修改权值的影响程度以弥补权值修改量的强行减小。

绝大多数的情况下在 SOM 开始训练之前，并没有先验知识能让我们事先去选择一个合适的网络规模，所以它严重地影响了 SOM 作为最优矢量量化器的应用。在改进的算法中采用神经元生长算法解决这个问题，首先将输出神经元初始化为较小的量，在训练过程中当输入矢量与获胜神经元的权值矢量之间的距离大于事先定义的阈值时，在原有的网络中增加一个输出神经元，并将用此输入矢量初始化新增的神经元的连接权值。

2.3 改进的 SOM 聚类训练步骤

步骤 1 网络初始化。将构件的 n 元组表示量化作为网络的输入样本，将$\{w_{ji}\}$赋予[0,1]区间的随机值；确定时间函数 $\eta(t)$ 的初值 $\eta(0)(0<\eta(0)<1)$；确定邻域 $N_g(t)$ 的初值 $N_g(0)$及总学习次数 T。确定最小距离的阈值 D。

步骤 2 选取一个输入样本 $U_k=[u_{k1},u_{k2},\cdots,u_{kn}]^T$，并对其进行归一化处理，即令 $\bar{U}_k=[\bar{u}_{k1},\bar{u}_{k2},\cdots,\bar{u}_{kn}]^T=U_k/\|u_k\|$ $\|u_k\|=[(u_{k1})^2+(u_{k2})^2+\cdots+(u_{kn})^2]^{1/2}$。

步骤 3 计算连接权矢量 $\bar{w}_j=[\bar{w}_{j1},\bar{w}_{j2},\cdots,\bar{w}_{jn}]^T$ 与输入矢量 $\bar{U}_k=[\bar{u}_{k1},\bar{u}_{k2},\cdots,\bar{u}_{kn}]^T$ 之间的加权欧氏距离：$d_j=[\sum_{i=1}^{n}\frac{1}{v_i}\times(\bar{u}_{ki}-\bar{w}_{ji})^2]^{1/2}(j=1,2,\cdots,M)$

其中

$$u_{ki}-w_{ji}=\begin{cases}similarity(Meaning(\bar{u}_{k1}),Meaning(\bar{w}_{j1})) & \text{if } i=1\\ 1 & \text{if } i=2,\cdots,n \text{ and } \bar{u}_{ki}=\bar{w}_{ji}\\ 0 & \text{if } i=2,\cdots,n \text{ and } \bar{u}_{ki}\neq\bar{w}_{ji}\end{cases}$$

$similarity(Meaning(\bar{u}_{k1}),Meaning(\bar{w}_{j1}))$是 $\bar{u}_{k1}$ 和 $\bar{w}_{j1}$ 之间的

语义近似程度 $\bar{U}_k=[\bar{u}_{k1},\bar{u}_{k2},\cdots,\bar{u}_{kn}]^T$ 中的第一项与后几项的含义不同，第一项为描述文本的矢量化而后几项则是类别编号，因此距离计算方法也不同。$v=\{v_1,v_2,\cdots,v_n\}$ 时表示 $\bar{u}_k$ 中各项的权重，是 $\bar{u}_k$ 映射到多维空间中各维的放大倍数，以此来调节样本在多位空间中相互之间的距离从而影响聚类结果，也就是说 V 表示样本中各项在聚类时的主次程度。在构件分类时，构件的刻面描述在分类中起主导作用，而构件之间的相互关系只起到影响分类边界处的构件归属的作用，因此 $V1$ 的值必须远大于其他几项的值。

步骤 4 找出最小距离 $d_g=\min[d_j](j=1,2,\cdots,M)$，确定获胜神经元 g。如果 $d_g>0$ 则生长出一个新的输出神经元 g 即为此构件最终归属的类别，将其初始连接权矢量初始化为 $\bar{u}_k$，返回步骤 2，如果 $d_g\leqslant D$，，则 g 为此构件最终归属的类别，转步骤 5。

步骤 5 进行连接权值的调整。对竞争层邻域 $N_g(t)$ 内所有神经元与输入层神经元之间的连接权进行修正：

$$w_{ij}(t+1)=\begin{cases}\bar{w}_{ij}(t)+\eta(t)(\bar{u}_{kj}-\bar{w}_{ij}(t)) & i\in N_g(t)\\ \bar{w}_{ij}(t) & \text{其它}\end{cases}$$

$(j=1,2,\cdots,n)$

步骤 6 对连接权矢量 $w_j(t+1)$ 进行归一化处理，即令

$$\bar{w}_j(t+1)=\frac{w_j(t+1)}{\|w_j(t+1)\|}$$

其中 $\|w_j(t+1)\|=[w_{j1}^2(t+1)+w_{j2}^2(t+2)\cdots+w_{jn}^2(t+1)]^{1/2}$

步骤 7 将下一个构件的 n 元组表示矢量化作为输入模式提供给网络的输入层，返回步骤 2，直到全部学习模式提供一遍。

步骤 8 更新学习速率 $\eta(t)$ 及邻域 $N_g(t)$

$$\eta(t)=\eta_0(1-\frac{t}{T})\qquad N_g(t)=INT[N_g(0)\times\sin(\frac{\pi}{2}\times\frac{t}{T})]$$

式中，$\eta(0)$ 为学习速率初值；t 为学习次数；T 为总学习次数，$INT[x]$ 表示对 x 取整。邻阈函数是在 $[0,N_g(0)]$ 上的以第一向限的正弦曲线递增的函数，这样使后续样本对权值的修改范围扩大，增大后续样本对聚类结果的影响程度。

3 实验与分析

3.1 实验数据介绍

下面通过实验来验证前面所描述的改进的 SOM 聚类算法的性能，并将该算法的结果同传统的 SOM 聚类法和 K 均值聚类法的结果作一比较。选择了两个数据集合来验证本文算法的性能，其中数据集 S_1 为随机生成的数据，而 S_2 为真实构件的数据信息。

(1) S_1 集合由 1000 个二维样本组成，这些样本按相同的先验概率分布于 8 个类别中，每个类别中样本都服从正态分布。其样本分布如图 1 所示。

(2) S_2 集合的数据包含 136 个构件的描述信息，它们来源于容纳了 451 个软构件的软构件库。库中全部软构件由 5 个刻面“功能”、“操作对象”、“应用领域”、“形态”和“性能”来描述。其中在“应用领域”刻面下构件构成一个三层树分类树，第一层分为 3 类：民用、军用和其它。第二层中又将民用分为企业信息化、GIS、通用工具构件、网络通信和其它管理类构件 5 类。第三层中将企业信息化分为采购、库存、电子商务、生产、设备、成本核算、销售和质量管理 8 个基本类别。数据集 S_2 中选取的 136 个构件属于企业信息化这一类别中，构件之间定义六种关系：数据传输关系，程序调用关系，网络传输关系，业务流转关系，消息传递关系和服务共享关系。每个构件在“应用领域”刻面下由一个七元组表示，将构件的刻面值描述文本进行分词和量化后就形成了 136 个七维的输入样本。这些样本在二维空间中的分布如图 2 所示。

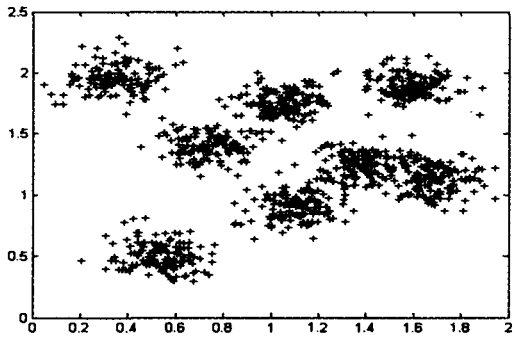

图 1 数据集 S_1 数据分布图

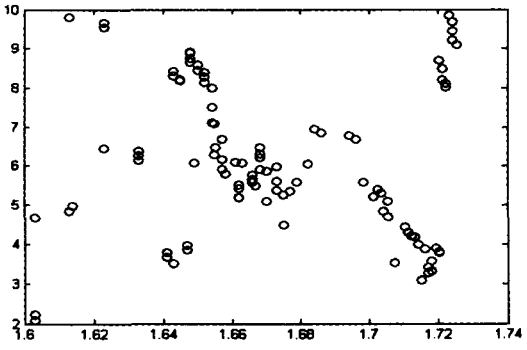

图 2 数据集 S_2 数据分布图

3.2 实验与分析

在实验时，对数据集 S_1 以其中的前 500 个样本作为训练数据，剩下的数据为测试样本，对数据集 S_2 以其中的前 100 个样本作为训练数据，剩下的数据为测试样本。对于数据集 S_1，K 均值聚类将类数设置为 8，从输入样本中随机选取 8 个样本作为初始凝聚点，SOM 聚类输入层 2 个结点，竞争层 8 个结点，学习效率 η 为 0.60，最大循环次数 $N=5000$，改进的 SOM 聚类权值 $V=\{1,1\}$，竞争层初始为 6 个结点其它参数与 SOM 聚类相同。对于数据集 S_2，K 均值聚类将类数设置为 5，从输入样本中随机选取 5 个样本作为初始凝聚点，SOM 聚类输入层 7 个结点，竞争层 5 个结点，学习效率 η 为 0.60，最大循环次数 $N=15000$，改进的 SOM 聚类权值 $V=\{1,0.1,0.1,0.05,0.1,0.05,0.05\}$，竞争层初始为 3 个结点其它参数与 SOM 聚类相同。对于同每个数据集 K 均值聚类和 SOM 聚类分别进行 100 次实验对实验结果取平均值进行性能比较，而对于改进的 SOM 聚类由于其具有聚类结果与输入样本次序无关的优点，同时在实验过程中前 50 的实验数据也表明该算法各次实验的结果差别不大，因此相对减少了实验次数，实验数据见表 1 和表 2 所示。为了证明聚类结果与样本的输入次序无关将数据集 S_1 和数据集 S_2 中的训练样本全部倒序分别用 SOM 聚类和改进的 SOM 聚类进行实验，表 3 给出了将两种算法的两种顺序的样本变化率（样本变化率＝所属类别发生变化的样本数目/总训练样本数目）。

表 1 数据集 S_1 的实验结果

算法名称	平均聚类准确率(%)	平均迭代次数	实验次数	输出神经元数目
K 均值聚类	57.6	21	100	8
SOM 聚类	78.4	34	100	8
改进的 SOM 聚类	82.7	52	50	8

表 2 数据集 S_2 的实验结果

算法名称	平均聚类准确率(%)	平均迭代次数	实验次数	输出神经元数目
K 均值聚类	46.3	14	100	5
SOM 聚类	60.2	53	100	5
改进的 SOM 聚类	84.5	84	30	6

表 3 样本变化率实验结果

	S_1 集合样本变化率(%)	S_2 集合样本变化率(%)
SOM 聚类	23.6	36.7
改进的 SOM 聚类	5.23	10.01

从表 1 和表 2 的平均迭代次数可以看出 K 均值聚类收敛速度最快,可处理的数据集合最大。但是,所得到的结果通常只是局部最优,或者说聚类准确率较低。K 均值方法比较适合于对实时性要求较高,对精度要求不高的信息分类。在构件库管理系统中构件的入库分类属于预处理过程对于实时性没有较高的要求,因此不宜采用 K 均值聚类。改进的 SOM 聚类准确率大大优于传统的 SOM 聚类和 K 均值聚类,但是改进的 SOM 聚类方法较传统的 SOM 聚类法收敛速度略慢。这主要是由于将阈值函数由线性递减改为正弦递增势必影响其收敛速度造成的,而且训练过程还包含了神经元生长过程也会增加训练时的迭代次数。由表 3 的样本变化率可以看出在样本输入次序发生变化之后改进的 SOM 聚类的聚类结果的变化远远小于传统的 SOM 聚类,所以在实验过程中改进的 SOM 聚类的多次试验的结果基本相同,因此可以适当减少实验次数,这样在时间上可以弥补其收敛速度较慢的缺点。

结论 为了克服软构件刻面分类法需要人工建立和维护术语空间的缺点,本文提出了一种基于 SOM 聚类的构件自动分类方法。该方法将聚类分析技术应用于软构件的分类中,定义了构件类别树,实现了摆脱术语空间的软构件自动分类,而且在分类时充分考虑了实际系统部署时构件之间的相互关系,使分类的结果更有利于构件的检索和基于构件系统的搭建。针对 SOM 聚类的聚类结果与样本输入次序有关和需要预先确定拓扑结构的缺点,文中提出了一种改进的 SOM 聚类算法并在实验中将其性能与 K 均值聚类和传统的 SOM 聚类算法加以比较,实验数据表明改进的 SOM 聚类算法聚类的准确率大大提高,而且可以通过在训练过程中生长神经元是找到最优的拓扑结构。

参 考 文 献

1 Poulin J S, Yglesias K P. Experiences with a faceted classification scheme in a large reusable software library(RSL). In: Proc. of Seventeenth Annual International Computer Software and Applications Conference, Phoenix, AZ, 3-5, 1993. 90～99

2 Morel J M, Faget J. The REBOOT environment. In: Proc. of the 2nd Intl. Workshop on Software Reusability Advances in Software. Lucca: IEEE Computer Society Press, 1993. 88～97

3 NEC Software Engineering Laboratory. NATO standard for management of a reusable software component library Volume 2. Tokyo: NATO Communication and Information System Agency, 1991

4 Chang JC, Li KQ, Guo LF, Mei H, Yang FQ. Representing and retrieving reusable software components in JB(Jadebird) system. Electronica Journal, 2000, 28(8): 1～6

5 Han Jiawei, Kamber M. Data mining: concepts and techniques. Jim Gray: Series Editor Morgan Kaufmann Publishers, 2000

6 Kohnen T. The self-organizing map. Neurocomputing. 1998, 21: 1～6

7 Yang Kiduk, Jacob E. A hybrid approach to generating and utilizing faceted vocabulary for knowledge discovery on the web. In: Proc. of the 2004Joint ACM/IEEE Conference on Digital Libraries. New York: ACM Press, 2004. 395～395

8 Yao Haining, Etzkorn L. Towards a semantic-based approach for software reusable component classification and retrieval. In: Proc. of the 42nd Annual Southeast Regional Conference. New York: ACM Press, 2004. 110～115

(上接第 215 页)

之间,比较平稳;折衷周期和最大周期的 R^{∞} 在 n 足够大时趋于一致;R^{∞} 具体变化如图 1 所示,图 1 是在精度为千分之一时,算法的实验结果曲线,一般而言,当 $n>2^k$ 时(k 是整数,k 随精度变化),最小周期算法好于最大和平均周期算法,当 $n<2^k$ 时,平均周期、最大周期和折衷周期算法好于最小周期算法;折衷周期算法始终优于最大周期算法;平均周期算法在 $n<2^j$ 时(j 是整数,j 随精度变化),当 n 足够大时,平均周期法优于折衷周期法。

小结 本文给出了基于时间触发的 CAN 协议构造 SM 的一种优化算法,详细介绍了构造 SM 的过程,简要分析了算法的性能,给出了实验分析结果,指出在绝大多数情况下,折衷周期算法的性能要优于其它三种算法。

本文进一步的工作是:在构造好基于时间触发的 CAN 协议任务调度的 SM 之后,对 SM 中相同性质的任务如何调整,以降低通信过程中的同步信息帧的复杂度。

参 考 文 献

1 Hartwich F ,Müller B,F Führer Th. Timing in the TTCAN network [A]. In: Proc. 8th Intl. CAN Conf. [C], Las Vegas, Nv, 2002

2 Führer Th,Müller B,Dieterle W. Time Triggered Communication on CAN[A]. In: Proc. 7th Intl. CAN Conf. [C]. Amsterdam, Netherland, 2000

3 Fonseca J,Countinho F,Barreriros J. Scheduling for a TTCAN network with a stochastic optimization algorithm [A]. In: Proc. of 4th FET2001,IFAC conf. on Fieldbus System and their Application[C]. Nancy,France,2001

4 Countinho F. Using Genetic algorithms to reduce jitter in control variables transmitted over CAN [c]. In: Proc. of ICC'2000 [C], Amsterdam,Netherland,2000

5 Coffman E G,Courcoubetis C. Perfect packing theorems and the averge-case behavior of optimal and online BIN packing [J]. SIAM Review 44,2002. 95～108

6 Xiaodong GU. Performance Analysis and Improvement for Some Linear On-Line Bin-Packing Algorithms [J]. Journal of Combinatorial Optimization,2002,6:455～471