

文章编号:1001-9081(2009)02-0535-04

## 基于支持向量机的计算资源反馈调度

徐 锐<sup>1</sup>, 王万良<sup>1</sup>, 李祖欣<sup>1,2</sup>

(1. 浙江工业大学 信息工程学院, 杭州 310032; 2. 湖州师范学院 信息工程学院, 浙江 湖州 313000)

(xukun907@yahoo.com.cn)

**摘要:** 在实时控制系统(RTCS)中,计算资源一般受限。由于诸多不确定性因素和工作负载的动态变化,系统总是运行在不可预期的开放环境中。为保证系统的稳定运行,提出了一种基于支持向量机(SVM)的计算资源反馈调度方法。它周期性地监测系统计算资源,通过SVM在线预测,得到各个控制回路的下一个采样周期,从而实现系统计算资源的动态分配。通过仿真实验验证了该反馈调度方法的性能,并与理想情况、传统开环调度方法进行了比较,显示了其优越性。

**关键词:** 实时控制系统; 支持向量机; 计算资源; 反馈调度

中图分类号: TP302 文献标志码:A

## Feedback scheduling of computing resources based on support vector machines

XU Kun<sup>1</sup>, WANG Wan-liang<sup>1</sup>, LI Zu-xin<sup>1,2</sup>

(1. College of Information Engineering, Zhejiang University of Technology, Hangzhou Zhejiang 310032, China;

2. School of Information Engineering, Huzhou Teachers College, Huzhou Zhejiang 313000, China)

**Abstract:** The computing resource is generally resources-limited in the real-time control systems (RTCS). Due to the uncertain running conditions and dynamic fluctuation of workload, the system always runs in an unexpected open environment. A feedback scheduling technique of computing resources based on support vector machines (SVM) was designed in order to ensure the system's stable running. The feedback scheduler of the system, which monitored the system's computing resources periodically, predicted a next sampling period for each control task by using SVM mechanism. Consequently, the computing resources of the system can be dynamically allocated online. The simulation of results highlights that the proposed scheduling technique is more effective in comparison with ideal case and traditional open-loop scheduling techniques.

**Key words:** Real-Time Control Systems (RTCS); Support Vector Machines (SVM); computing resource; feedback scheduling

### 0 引言

在现代实时控制系统中,多个控制任务共同分享同一个实时内核的处理器(计算)时间。因此计算时间被认为是一种资源约束,需要一些调度算法来为这些任务分配计算时间。在实时调度中,通常假设实时任务的时间约束是预先精确知道的,而且闭环控制任务常常具有一个已知的最坏执行时间和一个固定的采样周期;同时,这些任务还具有分别等于它们周期的截止时间。基于这些严格的假设,实时控制系统有两种传统的设计领域:控制器的设计和基于结构的调度策略。

一般来说,随着复杂性的增大,多任务会为争夺嵌入式的处理器而相互进行竞争。因此,总体控制性能不仅仅取决于控制算法的设计,而且与共享计算机资源调度效率有关。然而,嵌入式控制系统一般是基于控制与调度的分离而建立的<sup>[1]</sup>。这种分离是基于这样的假设:反馈控制器可以作为定期任务进行建模和实现,此周期任务有一个确定的周期,一个已知的最坏执行时间和一个强制的截止时间。这个假设已经被控制理论界广泛采用并用来发展采样控制理论。然而在实际应用中,这些假设通常是不切实际的。如一个任务通常有一个最坏执行时间,它是未知的且无法观察的。获得任务的

最坏执行时间虽然不是不可能的,但却是非常困难的。从控制的角度来讲,基于结构的调度策略算法都是开环的<sup>[2]</sup>。典型的是,这些资源分配在一个静态模式下,一旦调度基于系统的设置而建立,它将不能根据连续反馈而动态的调整。虽然这些调度算法在资源充足的环境中运行良好,但它们容易造成系统高度利用不足,并且系统在资源不足的情况下其性能将快速的恶化。尤其是在未知环境下,系统的工作量不能被精确建模,系统的性能将会很差。

针对上述问题,近几年人们提出了一种同时考虑实时计算和控制的方法——控制与调度协同设计<sup>[3-4]</sup>。它将控制领域的反馈思想运用到资源分配中,在计算资源约束下,使实时控制系统的整体性能得到保障。能够为系统在负载可变的动态环境下提供所需的控制质量(Quality of Control, QoC)保证,实现了资源的反馈调度。

在这个新兴的计算资源反馈调度领域,将计算资源(CPU利用率)作为反馈变量,各个回路采样周期作为操纵变量,来实现资源优化分配的一些相关工作已经展开<sup>[5-7]</sup>。文献[5]提出了当性能函数采用标准的二次型或线性函数来描述时,在一定的允许范围内和采样时间(采样周期)近似存在着线性关系。因此,在面临工作量发生变化时,调度器可以通过调

收稿日期:2008-08-08;修回日期:2008-09-23。 基金项目:国家自然科学基金资助项目(60573123)。

作者简介:徐锐(1983-),男,浙江临安人,硕士研究生,主要研究方向:网络控制系统; 王万良(1957-),男,江苏高邮人,教授,博士生导师,主要研究方向:计算机控制、智能自动化、计算机网络、计算机集成自动化系统、生产计划与调度; 李祖欣(1972-),男,浙江长兴人,副教授,博士研究生,主要研究方向:网络控制系统、智能控制。

整当前控制任务集的采样周期来实现系统的可调度和整体控制性能的优化。文献[6]讨论了在实时控制回路中为达到控制性能最优化,分配计算资源的两个主要子问题:控制性能如何依赖于采样时间和如何设计最优化的递归资源分配程序。文献[7]采用了基于神经网络的反馈调度器对各个回路的采样周期进行动态调整以适应负载的变化,同时提高了网络利用率。

针对工作负载不可预期的变化及控制系统的运行环境的不确定性,保证系统的稳定运行,并且为了克服数学最优反馈调度方法计算量过大的缺点<sup>[8-9]</sup>和一般机器学习算法的过拟合问题以及提高学习机的泛化能力<sup>[10-11]</sup>,本文设计了一个基于支持向量机(Support Vector Machines,SVM)的动态预测反馈调度器。它周期性地监测计算资源,根据可使用的最大CPU利用率,通过SVM在线预测,得到各个控制回路的下一个采样周期,从而实现系统计算资源的动态分配。

## 1 问题描述

对于我们所考虑的多任务控制系统,各个控制回路的控制器同时共享了同一个CPU的计算资源。反馈调度器的设计为在任务调度(计算资源)受限时,最大化地改善系统总体控制性能起到了关键作用。此时可以将反馈调度问题描述为一个约束优化问题,即在系统任务可调度约束下,通过优化调度参数来实现整体控制代价的最小化。最优反馈调度问题是建立在一定的数学优化方法上<sup>[12-13]</sup>。假定第*i*个控制任务的采样周期和执行时间分别表示为*h<sub>i</sub>*和*c<sub>i</sub>*,系统中的控制回路数为*n*,可将最优反馈问题描述为<sup>[6,14]</sup>:

$$\min J = \sum_{i=1}^n J_i(h_i) \quad (1)$$

$$\text{s. t. } \sum_{i=1}^n c_i/h_i \leq U_{\text{ref}}$$

其中*J<sub>i</sub>(h<sub>i</sub>)*为回路的代价函数,是采样周期的函数。*U<sub>ref</sub>*为CPU利用率的设定值。

可以看出,式(1)给出的最优反馈调度描述的是一个非线性规划问题。为了获取线性约束,通常用采样频率*f<sub>i</sub> = 1/h<sub>i</sub>*代替采样周期作为变量。文献[5,6]指出,对于式(1)中的代价函数,在较小的采样周期下,可以近似的处理为:*J<sub>i</sub>(f<sub>i</sub>) = α<sub>i</sub> + β<sub>i</sub>/f<sub>i</sub><sup>2</sup>*。其中:

$$\begin{cases} \beta = J_h(h_0)/(2h_0) \\ \alpha = J(h_0) - \beta h_0^2 \end{cases} \quad (2)$$

*h<sub>0</sub>*为0时刻回路采样周期。

求解这个线性约束优化问题,由Kuhn-Tucker条件,可得到:

$$\begin{cases} \nabla V(f^*) + \lambda c = 0 \\ \lambda [U_{\text{ref}} - c^T f^*] = 0 \end{cases} \quad (3)$$

其中*∇V*是梯度, *c* = [c<sub>1</sub>, …, c<sub>n</sub>]<sup>T</sup>, *λ*为Lagrange乘子, *λ* ≥ 0。

求解式(3),可以得到:

$$\begin{cases} \lambda = (1/U \sum_1^n C_i^{2/3} (2\beta_i)^{1/3})^3 \\ f_i = (2\beta_i / (\lambda C_i))^{1/3} \end{cases} \quad (4)$$

对于式(3)的求解过程,往往都要涉及函数梯度、Hessian矩阵等复杂运算,且需要经过相当多次的迭代,计算工作量比较大。而为了处理负载的动态变化,反馈调度器往往需要在线使用,因此在实际应用中必须得考虑到反馈调度本身的资

源开销。如果反馈调度消耗过多的计算资源,那么必然会对系统的控制性能造成较大的负面影响<sup>[8-9]</sup>。

为了克服数学最优反馈调度方法计算量过大的缺点,需要使用一种快速、智能的反馈调度策略。对于系统给定的资源限制(最高CPU利用率),根据系统反馈回的实际利用率,在线的得到各个回路的采样时间,使得系统整体控制性能最优。于是,我们提出一种基于支持向量机的资源调度策略。

## 2 基于 SVM 的动态预测反馈调度

SVM是根据Vapnik的结构风险最小化原则,尽量提高学习机的泛化能力,由有限的训练样本得到的小的误差能够保证对独立测试集仍保持小的误差,它比经验风险原理的人工神经网络学习算法具有更强的理论依据和更好的泛化性能<sup>[10-11]</sup>。另外,支持向量机算法是一个凸二次优化问题,能够保证找到的极值解是全局最优解。算法在很大程度上解决了模型选择、过拟合、非线性、维数灾难等问题。

基于SVM的动态预测实际上是一个数据泛化拟合问题,即先根据输入输出样本进行学习,然后对不在学习样本集中的输入数据,计算出相应的输出值。

支持向量回归算法完全根据部分训练样本构造回归函数,不需要关于问题和样本集或是回归函数结构的先验信息,因此,对于给定的随机分布的样本点,不同结构的支持向量回归机(Support Vector Regression,SVR)方法将给出基本相同的结果<sup>[15]</sup>,这是它优于神经网络的一个方面。以下我们将采用支持向量回归机常选择的损失函数——ε-不敏感损失函数进行仿真研究。

### 2.1 ε-支持向量回归机

设给定训练集

$$T = \{(x_1, y_1), \dots, (x_l, y_l)\} \in (X \times Y)^l \quad (5)$$

其中:*x<sub>i</sub>* ∈ *X* = *R<sup>n</sup>*, *y<sub>i</sub>* ∈ *Y* = *R*, *i* = 1, …, *l*。假定训练集是按*X* × *Y*上的某个概率分布 *P(x,y)* 选取的独立同分布的样本点。所考虑的超平面为 *f(x) = ⟨w, x⟩ + b*, 其中 *w* ∈ *R<sup>n</sup>*, *b* ∈ *R*。

ε-不敏感损失函数定义为:

$$c(x, y, f(x, \alpha)) = |y - f(x, \alpha)|_\varepsilon = \begin{cases} 0, & |y - f(x, \alpha)| \leq \varepsilon \\ |\varepsilon|, & |y - f(x, \alpha)| = \varepsilon, \text{ 其他} \end{cases} \quad (6)$$

其中, *α*为回归权系数,是待定的常数, *ε*为一正常数。引入两个松弛变量: *ξ<sub>i</sub>, ξ<sub>i</sub><sup>\*</sup>* ≥ 0, *i* = 1, 2, …, *l*。那么,优化方程为:

$$\min \frac{1}{2} \|w\|^2 + C \sum_{i=1}^l (\xi_i + \xi_i^*) \quad (7)$$

$$\text{s. t. } f(x_i) - y_i \leq \xi_i + \varepsilon$$

$$y_i - f(x_i) \leq \xi_i^* + \varepsilon$$

通过引入拉格朗日函数求解这个优化函数,根据极值条件,可得到下面的式子:

$$\begin{cases} w = \sum_{i=1}^l (\alpha_i - \alpha_i^*) x_i \\ \sum_{i=1}^l (\alpha_i - \alpha_i^*) = 0 \\ C - \alpha_i - \gamma_i = 0 \\ C - \alpha_i^* - \gamma_i^* = 0 \end{cases} \quad (8)$$

其中 *α<sub>i</sub>, α<sub>i</sub><sup>\*</sup>, γ<sub>i</sub>* ≥ 0, *i* = 1, …, *l*。由KKT<sup>[10]</sup>条件,有 *α<sub>i</sub><sup>\*</sup> × α<sub>i</sub>* = 0 成立,也就是只有支持向量对应的拉格朗日乘子不等于零。

因此,可以只采用训练样本中的少数支持向量就可以实现函数估计。

选取 RBF 核函数  $K(x, x_i) = \exp\left(-\frac{\|x - x_i\|^2}{\gamma^2}\right)$ ,  $\gamma$  为核参数。那么,通过求解优化问题可以得到最优解  $\alpha = (\alpha_1, \alpha_1^*, \dots, \alpha_l, \alpha_l^*)^T$ 。

于是,可以得到支持向量机的回归估计公式:

$$f(x) = \sum_{i=1}^l (\alpha_i - \alpha_i^*) K(x, x_i) + b \quad (9)$$

其中,  $b$  按下列方式计算,选择位于开区  $(0, C/l)$  中的  $\alpha_j$  或  $\alpha_k$ ,若是选到的是  $\alpha_j$ ,则:

$$b = y_j - \sum_{i=1}^l (\alpha_i - \alpha_i^*) K(x_i, x_j) + \varepsilon \quad (10)$$

若选到的是  $\alpha_k$ ,则:

$$b = y_k - \sum_{i=1}^l (\alpha_i - \alpha_i^*) K(x_i, x_k) - \varepsilon \quad (11)$$

在获得  $k$  时刻的回归权系数  $\alpha$  和偏移值  $b$  后即可得到  $k+1$  时刻的回路采样周期的预测值:

$$\hat{h}_{k+1} = f(x_{k+1}) = \sum_{i=1}^k (\alpha_i - \alpha_i^*) K(x, x_i) + b \quad (12)$$

## 2.2 支持向量机的反馈调度

本文提出的基于支持向量机的反馈调度模型如图 1 所示。

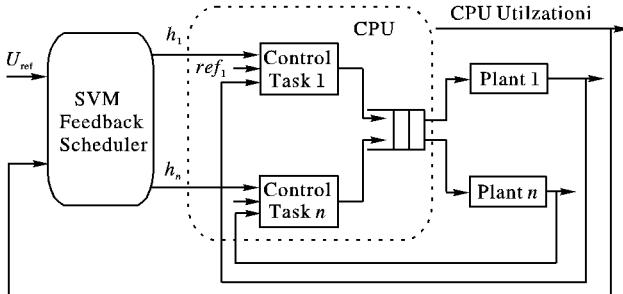


图 1 基于支持向量机的反馈调度

其中:  $U_{ref}$  为 CPU 利用率的设定值;  $h_i$  为第  $i$  个控制回路的采样周期值。由于 CPU 利用率是无法直接监测到的,而  $u = \sum_{i=1}^n c_i/h_i$ ,因此,可以将前一采样时刻的任务执行时间作为反馈调度器的输入来取代不可直接获得的 CPU 利用率,其实质都是反馈当前 CPU 计算资源状况。

支持向量机反馈调度器设计的基本思路就是利用支持向量机来逼近数学求解方法——序列二次规划 (Sequential Quadratic Programming, SQP) 的最优解,并进行在线使用。基于这种方法,数据样本集的产生只需利用 SQP 对最优反馈调度问题进行离线求解。在对数据样本进行(归一化)预处理后,就可以用于对支持向量机进行训练和预测。

本文考虑  $\varepsilon$ -SVM 回归问题。为了产生数据样本集,取执行时间  $c_1$  和  $c_2$  的变化范围均为  $[8, 10]$ ,步长为 0.2;  $U_{ref}$  的变化范围为  $[0.4, 1]$ ,步长为 0.0125; 时间单位均为毫秒。运用 SQP 算法进行离线求解,可以得到相应的 2400 组样本数据。之后,对所得的输入数据进行训练,得到预测模型,加入到反馈调度器中进行在线预测,从而实现计算资源的动态分配。

## 3 性能评估

考虑采用一个处理器同时控制两个倒立摆,其状态空间模型分别为:

$$\begin{cases} \dot{x}(t) = \begin{pmatrix} 0 & 33.6 \\ 1 & 0 \end{pmatrix} x(t) + \begin{pmatrix} 1 \\ 0 \end{pmatrix} u(t) \\ y(t) = (33.6 \ 0) x(t) \end{cases} \quad (13)$$

$$\begin{cases} \dot{x}(t) = \begin{pmatrix} 0 & 98 \\ 1 & 0 \end{pmatrix} x(t) + \begin{pmatrix} 1 \\ 0 \end{pmatrix} u(t) \\ y(t) = (98 \ 0) x(t) \end{cases} \quad (14)$$

各回路控制器采用 LQ 最优控制设计<sup>[16]</sup>,状态反馈增益  $K$  分别设计为  $[11.6374, 67.2149], [19.8245, 196.0051]$ 。选择  $\varepsilon$ -SVM 回归问题,选用最常用的 RBF 核函数,对训练数据进行离线训练,得到相应的预测模型。仿真中,采用多次尝试和验证的方法得到惩罚因子  $C = 8$ ,核函数参数  $g = 0.015625$ 。为了说明控制质量和计算资源利用的改善程度,对本文采用的支持向量机反馈调度与传统开环调度(控制回路均采用固定采样周期)、理想情况下(忽略任务的执行时间)进行了比较。

所有的仿真中,各回路是固定优先级的。其中,调度任务具有最高优先级,任务 2 的优先级最低。两个任务初始化采样时间  $h_0$  分别为 33 ms 和 20 ms,执行时间均在  $[8, 10]$  范围内随机变化。在时间  $t = 2$  s 时,改变 CPU 利用率的设定值  $U_{ref}$  从 0.8 改变为 0.5。反馈调度任务作为一个周期为 100 ms 的周期任务执行,它的调度开销忽略不计。

图 2、3 描述了在不同的策略下各个控制回路的输出响应。由图可以看出,相比理想状况,考虑控制算法的执行时间对系统的整体性能有一定的影响。在  $t = 2$  s 之前,三种策略下两个控制回路均运行稳定。此时系统实际最大 CPU 利用率为  $U_{max} = 10/33 + 10/20 = 0.803 \approx 0.8$ 。而此时的 CPU 利用率设定值是  $U_{ref} = 0.8$ ,也就是,CPU 有充足的计算资源去执行控制器任务。因此,两个倒立摆都运行正常。在  $t = 2$  s 时,计算资源受到限制,CPU 利用率设定值变为  $U_{ref} = 0.5$ ,而实际最小 CPU 利用率为  $U_{min} = 8/33 + 8/20 = 0.642 > 0.5$ ,系统过载。在传统开环调度策略下,系统变为不稳定:由于回路 2 的优先级较低,导致其得不到充足的 CPU 计算时间,而使这个回路的倒立摆失控。而本文提出的支持向量机反馈调度方法能使系统始终运行在稳定状态下。当系统出现过载时,支持向量机反馈调度器根据计算资源约束,在线对系统进行优化:根据实际 CPU 利用率,得出最优化理论下的各回路采样时间,并且在线调整,保证了各回路的稳定,实现系统整体控制性能的优化。

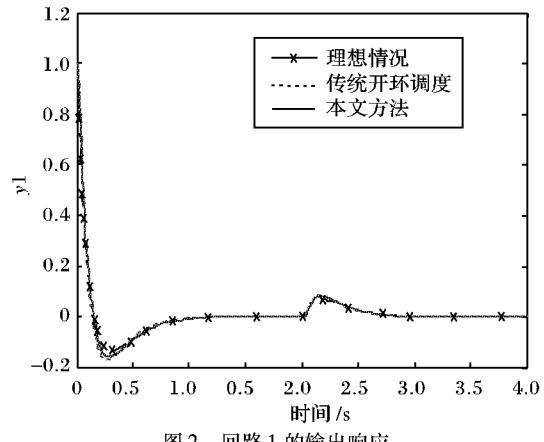


图 4、5 分别是传统方法和 SVM 反馈调度下部分运行时间的系统调度图。图 4 中,第三条曲线描述了系统的 CPU 利用率:在  $t = 2$  s 之前,系统可使用的 CPU 资源为 80%;当在

$t = 2$  s 之后, 系统的计算资源出现变化, 可使用的 CPU 资源变为 50%。由于任务 2 的优先级较低, 计算资源发生变化后对其产生的影响最大, 由图 4 可以看出:  $t = 2$  s 后, 由于被其他高优先级的任务所抢占, 任务 2 获得的计算资源很少, 从而导致该回路的不稳定。

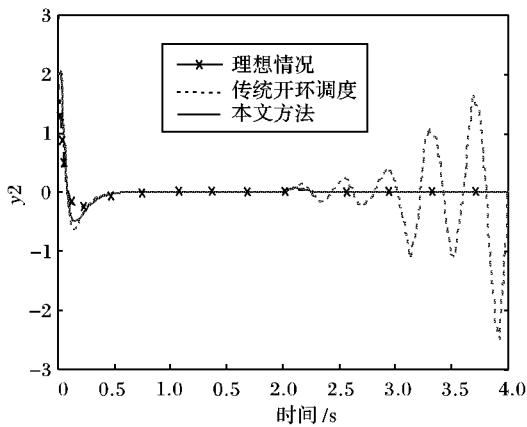


图 3 回路 2 的输出响应

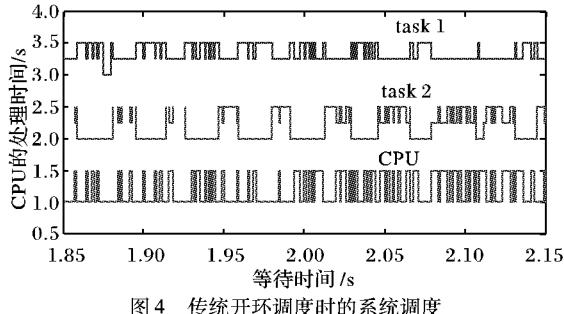


图 4 传统开环调度时的系统调度

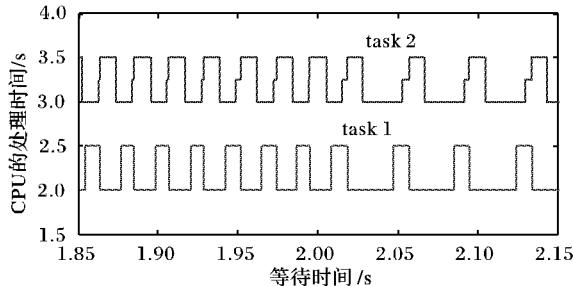


图 5 SVM 反馈调度时的系统调度

从图 5 中能够清晰地看出: 在系统 CPU 资源受到约束时, 方法能够调整各回路的采样周期, 使得各回路都能很好地得到任务处理时间, 保证了系统稳定。这是由于: 在  $t = 2$  s 之前, 在每个采样时间点, SVM 反馈调度器根据实际可利用的 CPU 资源(此时为 80%), 以及任务执行时间  $c_1$  和  $c_2$  在线预测各回路下个周期的采样时间, 保证了系统的稳定。在  $t = 2$  s 后, 由于 CPU 计算资源发生变化(变为 50%), SVM 反馈调度器能够根据监测到的资源变化, 得到优化后的回路采样时间, 做出及时的调整, 使各回路仍然稳定运行。

图 6 描绘了三种调度策略下系统总控制代价变化情况。由图 6 可知, 在传统开环调度策略下, 系统最终趋向不稳定。与其他两种反馈调度方法相比, 采用开环调度时的系统整体 QoC 最差。而使用支持向量机反馈调度, 很好地解决了在计算资源约束下, 系统整体控制性能的优化。结果显示了支持向量机在有效地处理资源限制和工作负载动态变化上的能力。

#### 4 结语

实际的实时控制系统常常运行在不可预期的开放环境中, 因此资源的反馈调度尤为重要。本文提出了基于 SVM 的计算资源反馈调度, 实现在系统的控制质量和处理器计算资源约束之间取得较好的平衡。同时克服了最优反馈调度开销大、不宜在线使用的缺点, 获得了更低的调度开销。同时, 支持向量机反馈调度还具有较好的适应性、鲁棒性和容错性等。

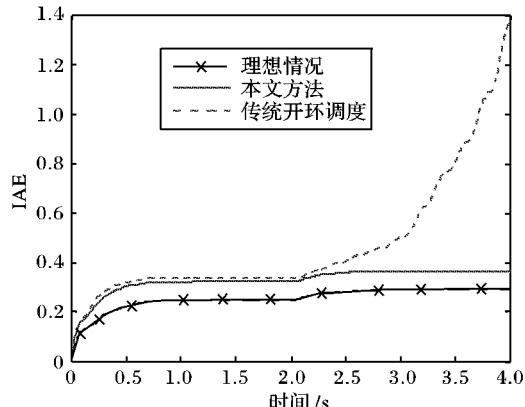


图 6 不同策略下的系统总绝对误差积分

#### 参考文献:

- [1] ARZEN K E, BERNHARDSSON B, EKER J, et al. Integrated control and scheduling, ISRN LUTFD2/TFRT7586SE[R]. Sweden: Lund Institute of Technology, 1999.
- [2] LU C, STANKOVIC J A, TAO G, et al. Feedback control real-time scheduling: framework, modeling, and algorithms[J]. Real-time Systems, 2002, 23(1/2): 85 – 126.
- [3] FENG XIA, SUN YOU-XIAN. Control-scheduling codesign: a perspective on integrating control and computing[C]// Dynamics of Continuous, Discrete and Impulsive Systems-Series B: Applications and Algorithms, Special Issue on ICSCA'06. Waterloo, Canada: Watum Press, 2006: 1352 – 1358.
- [4] ARZEN K E, CERVIN A. Control and embedded computing: survey of research directions[C]// Proceeding of 16th IFAC World Congress. Prague, Czech Republic: Elsevier, 2005: 20 – 23.
- [5] CERVIN A, EKER J, BERNHARDSSON B, et al. Feedback-feedforward scheduling of control tasks[J]. Real-Time Systems, 2002, 23(1/2): 25 – 53.
- [6] EKER J, HAGANDER P, ARZEN K E. A feedback scheduler for real-time controller tasks[J]. Control Engineering Practice, 2000, 8(12): 1369 – 1378.
- [7] FENG XIA, LI SHAN-BIN, SUN YOU-XIAN. Neural network based feedback scheduler for networked control system with flexible workload[C]// Proceeding of Lecture Notes in Computer Science 2005. Heidelberg: Springer-Verlag, 2005: 242 – 251.
- [8] ARZEN K E, CERVIN A, HANRIKSSON D. Resource-constrained embedded control systems: possibilities and research issues[C]// Proceeding of Co-design of Embedded Real-Time Systems Workshop. Porto, Portugal: [s. n.], 2003: 28 – 34.
- [9] MARTI P. Analysis and design of real-time control systems with varying control timing constraints[D]. Spain: Department of Automatic Control, Technical University of Catalonia, 2002.
- [10] VAPNIK V. 统计学习理论的本质[M]. 张学工, 译. 北京: 清华大学出版社, 2000.

(下转第 541 页)

$$[x_6]_A^{\leq} = \{x_6\}$$

由表2可以得到:

$$x_1^{*D*D} = \{x_1, x_2, x_5, x_6\}$$

$$x_2^{*D*D} = \{x_2, x_5, x_6\}$$

$$x_3^{*D*D} = \{x_2, x_3, x_4, x_5, x_6\}$$

$$x_4^{*D*D} = \{x_4, x_6\}$$

$$x_5^{*D*D} = \{x_5\}$$

$$x_6^{*D*D} = \{x_6\}$$

显然有  $x^{*D*D} = [x]_A^{\leq}$ 。

在基于优势关系的信息系统中的多个属性并不是同等重要的,因而有必要研究信息冗余问题,建立优势与劣势关系上的属性约简方法,合理简化信息决策问题。

**定义8<sup>[9]</sup>** 设  $S = (U, A, V, f)$  是一个基于优势关系的信息系统,给出优势关系  $R_B^{\leq}$  及劣势关系  $R_B^>$ ,若  $R_B^{\leq} = R_A^{\leq}$  或  $R_B^> = R_A^>$ ,称  $B$  为优势协调集或劣势协调集。若  $B$  是优势协调集或劣势协调集,且  $B$  的任意真子集都不是优势协调集或劣势协调集,称  $B$  为优势关系或劣势关系下的属性约简集,简称优势约简集或劣势约简集。

同时可以证明,在一个基于优势关系的信息系统中,优势约简集和劣势约简集是等价的。

**定义9** 设形式背景  $(U, D, I)$  是由基于优势关系的信息系统  $S = (U, A, V, f)$  导出的形式背景,对属性集  $B \subseteq D$ ,如果对任意  $x \in U$  都有  $x^{*B*B} = x^{*D*D}$ ,则  $B$  被称为优势关系下的对象粒协调集。如果  $B$  的任意子集都不是优势关系下的对象粒协调集,则  $B$  称为该形式背景的优势关系下的对象粒约简。

**定义10** 设  $(U, D, I)$  是一个基于优势关系的形式背景,定义:

$$M^*(x, y) = y^{*D} - x^{*D}$$

称  $M^*(x, y)$  为  $(U, D, I)$  的优势关系下的对象粒辨识属性集,称  $\{M^*(x, y) : (x, y) \in U \times U\}$  为  $(U, D, I)$  的优势关系下的对象粒辨识矩阵。

**推论1** 在形式背景  $(U, D, I)$  中,设  $x_i, x_j \in U$ ,若  $x_i \in [x_j]_D^{\leq}$ ,则  $M^*(x_j, x_i) = \emptyset$ 。

对于表1的信息系统,可以计算它的优势约简集是  $\{a_2, a_3\}$ ,同时根据定义10可以得到表2的形式背景的对象粒辨识矩阵。

表3 对象粒辨识矩阵

	$x_1$	$x_2$	$x_3$	$x_4$	$x_5$	$x_6$
$x_1$	$\emptyset$	$\emptyset$	$a_{22}$	$a_{22}$	$\emptyset \emptyset$	
$x_2$	$a_{12} a_{13} a_{32}$	$\emptyset$	$a_{12} a_{13} a_{22}$	$a_{13} a_{22}$	$\emptyset \emptyset$	
$x_3$	$a_{32}$	$\emptyset$	$\emptyset$	$\emptyset$	$\emptyset \emptyset$	
$x_4$	$a_{12} a_{32} a_{33}$	$a_{12} a_{32} a_{33}$	$a_{12} a_{32} a_{33}$	$\emptyset$	$a_{33}$	$\emptyset$
$x_5$	$a_{12} a_{13} a_{23} a_{32}$	$a_{23}$	$a_{12} a_{13} a_{23} a_{22} a_{32}$	$a_{13} a_{22} a_{33}$	$\emptyset a_{23}$	
$x_6$	$a_{12} a_{13} a_{32} a_{33}$	$a_{33}$	$a_{12} a_{13} a_{22} a_{33}$	$a_{13} a_{22}$	$a_{33} \emptyset$	

(上接第538页)

- [11] CORTES C, VAPNIK V. Support vector networks[J]. Machine Learning, 1995, 20(3): 273–297.
- [12] CASTANE R, MARTI P, VELASCO M, et al. Resource management for control tasks based on the transient dynamics of closed-loop systems[C]// Proceedings of the 18<sup>th</sup> Euromicro Conference on Real-Time Systems (ECRTS06). Washington, DC: IEEE Computer Society, 2006: 171–182.
- [13] JIN HONG, WANG HUI, WANG HONGAN, et al. Optimization design of controller periods using evolution strategy[J]. LNAI, 2005,

取  $B = \{a_{22}, a_{23}, a_{32}, a_{33}\}$ ,可以验证  $B$  是表2的优势关系下的对象粒协调集,也是该形式背景的优势关系下的对象粒约简。因此优势关系下的对象粒约简与优势约简集是等价的。

### 3 结语

基于优势关系的信息系统是一类在现实生活中普遍存在的信息系统,本文对形式概念分析在基于优势关系的信息系统中的应用进行了研究,证明了由基于优势关系的信息系统导出的形式背景中的对象粒与优势类是等价的,对象粒协调集与优势协调集是等价的,这些结论进一步丰富了形式概念分析理论,对基于优势关系的信息系统的知识获取的研究有着积极的意义。

#### 参考文献:

- [1] POLKOWSKI L, SKOWRON A. Rough sets in knowledge discovery 1- Methodology and applications [M]. Wurzburg: Physica-Verlag, 1998.
- [2] KAZIMIERZ Z. Rough approximation of a preference relation by a multi-attribute dominance for deterministic, stochastic and fuzzy decision problems [J]. European Journal of Operational Research, 2004, 159(1): 196–206.
- [3] SALVATORE G, BENEDETTO M, ROMAN S. Rough sets methodology for sorting problems in precedence of multiple attributes and criteria[J]. European Journal of Operational Research, 2002, 138(2): 247–259.
- [4] WILLE R. Restructuring lattice theory: an approach based on hierarchies of concepts[C]// Ordered Sets. Dordrecht: Reidel, 1982: 445–470.
- [5] 张文修, 姚一豫, 梁怡. 粗糙集与概念格[M]. 西安: 西安交通大学出版社, 2006.
- [6] 曲开社, 翟岩慧. 偏序集、包含度与形式概念分析[J]. 计算机学报, 2006, 29(2): 219–226.
- [7] 曲开社, 翟岩慧, 梁吉业, 等. 形式概念分析对粗糙集理论的表示及扩展[J]. 软件学报, 2007, 18(9): 214–218.
- [8] 梁吉业, 王俊红. 基于概念格的规则产生集挖掘算法[J]. 计算机研究与发展, 2004, 41(8): 1339–1344.
- [9] 张文修, 仇国芳, 基于粗糙集的不确定决策[M]. 北京, 清华大学出版社, 2005.
- [10] GANTER B, WILLE R. Formal concept analysis: Mathematical foundations[M]. Berlin: Springer-Verlag, 1999.
- [11] 宋笑雪, 张文修. 形式概念分析与集值信息系统[J]. 计算机科学, 2007, 34(11): 129–131.
- [12] WU WEI ZHI, LEUNG Y, MI J S. Granular computing and knowledge reduction in formal contexts[C]// IEEE Transactions on Knowledge and Data Engineering. Washington, DC: IEEE Computer Society, 2008.

3801(1): 1100–1105.

- [14] SETO D, LEHOCZKY J P, SHA L, et al. On task schedulability in real-time control systems[C]// Proceeding of 17<sup>th</sup> IEEE RTSS. Washington, DC: IEEE press, 1996: 13–21.
- [15] HWANG C, HONG D H, KYUNG H S. Support vector interval regression machine for crisp input and output data[J]. Fuzzy Sets and Systems, 2006, 157(8): 1114–1125.
- [16] ARZEM K J, WITTENMARK B. Computer controlled systems[M]. New Jersey, US: Prentice Hall, 1997.