

文章编号: 1001-0920(2005)09-0981-05

基于种群小生境微粒群算法的前向神经网络设计

王俊年^{1,2}, 申群太¹, 周少武², 沈洪远^{1,2}

(1. 中南大学 信息科学与工程学院, 长沙 410083; 2. 湖南科技大学 信息与电气工程学院, 湖南 湘潭 411201)

摘要: 根据自然界中鱼鸟等所具有的种群运动特征, 借鉴递阶编码的思想, 构造出一种种群小生境微粒群算法, 具有小生境内个体微粒自由运动特征分量和小生境种群运动特征分量分层递阶进化的特征, 克服了标准微粒群算法或其改进算法在多峰函数寻优时出现的微粒“早熟”现象。应用该算法进行三层前向神经网络连接权值和网络结构联合并行自适应设计, 在混沌时间序列预测中显示了良好的性能

关键词: 微粒群算法; 种群小生境; 神经网络; 结构优化

中图分类号: TP183 **文献标识码:** A

Evolutionary Design of Feed-forward Neural Network Based on Species Niche Particle Swarm Optimizer

WANG Jun-nian^{1,2}, SHEN Qun-tai¹, ZHOU Shao-wu², SHEN Hong-yuan^{1,2}

(1. Institute of Information Science and Engineering, Central South University, Changsha 410083, China; 2. Institute of Information and Electrical Engineering, Hunan University of Science and Technology, Xiangtan 411202, China. Correspondent: WANG Jun-nian, E-mail: wangjunnian@sina.com)

Abstract: A species niche particle swarm optimizer (SNPSO) is proposed according to the moving behavior of bird flock or fish school. The new algorithm has the characteristics of the particle's free moving offset and the niching population's moving offset evolved in a hierarchy model. Using these characteristics, our algorithm overcomes the "prematurity" in global optimization by the normal particle swarm optimization and its modified model. When using this algorithm in the feed forward structure and weights associated training, the neural network shows a good performance in chaotic time sequence forecasting.

Key words: Particle swarm optimization; Species niche; Neural network; Structure optimization

1 引言

由 Kennedy 等^[1-7]开发的微粒群算法 (PSO), 以其概念简单, 实现容易, 而广泛应用于神经网络训练^[7-13]。如F van den bergh 等分别将标准PSO^[8]、协同进化PSO (CPSO)^[9,10]和自适应PSO (APSO)^[11]分别应用于神经网络的权值训练, 使网络训练精度和训练时间有很大改善。但这些只是改善网络权值的训练, 而神经网络的设计包含了网络结构和网络权值两部分, 这两部分联合设计一直是神经网络进化设计的方向。D K Agrafiotis 等^[12]将二进制 PSO (BPSO) 与 BP 算法相结合 (BPSO-BP 算法), 利用

BPSO 优化前向网络的结构和BP 算法训练网络权值; 为克服BPSO-BP 算法中训练步长难以确定的缺点, 最近 Z W Zhang^[13]等提出前向网络的BPSO-PSO 算法, 采用BPSO 和 PSO 分别训练网络结构和权值。但文献[12, 13]都未能证明其算法的收敛性, 事实上, 如果将神经网络模型的优化设计看作是一个在网络结构与联结权值空间进行优化计算的问题, 这就是一个多峰函数寻优问题。BPSO-BP 和 BPSO-PSO 算法都是在全局解空间内某一段时间单独运行BPSO 或PSO, 无法获得全局最优解。

为将微粒群算法从单模优化问题求解扩展到多

收稿日期: 2004-10-15; 修回日期: 2005-03-07

基金项目: 国家自然科学基金项目 (60474029); 湖南省教育厅科研项目 (03C499)。

作者简介: 王俊年(1968—), 男, 副教授, 博士生, 从事智能控制、智能信号处理的研究; 申群太(1945—), 男, 教授, 博士生导师, 从事智能控制等研究。

模问题的优化, R Brits^[14, 15]等将常用于遗传算法的小生境技术应用在微粒群算法中, 构造出一种能够并行求解多峰函数优化问题的小生境微粒群算法(NichePSO). NichePSO 算法在一定程度上达到了保护微粒群多样性的目的, 在多峰函数优化问题求解中显示了较好的性能, 但这种算法中小生境的划分(即子微粒群的划分)依赖于个体微粒的适应值, 过分强调了个体微粒之间的竞争和差异, 而忽略了子微粒群中个体微粒之间的共性和个体微粒与子微粒群之间的属性关系. 自然界中的鱼鸟等群体的运动, 除个体运动特征不同外, 同一物种(种群)内个体的运动有其相同的地方. 对于同样的食物源, 不同的种群能够以不同的运动方式或路径找到它, 而且不同种群之间还可能有机会通过学习和交流来改善自己的种群运动属性.

基于上述思想, 借鉴分层递阶遗传算法原理和小生境技术, 本文在分析神经网络输入输出映射的基础上, 根据神经网络训练特点设计出一种新的多种群并行进化种群小生境微粒群算法(SNPSO), 在前向神经网络结构和权值联合进化设计中显示了良好的性能.

2 种群小生境微粒群算法(SNPSO)

2.1 GCPSO 介绍

标准版本的PSO 算法在第 $t+1$ 代对第 i 个微粒在第 d 维($1 \leq d \leq D$) 上的速度和位置采用下列公式进行操作^[1-3]:

$$\begin{aligned} v_{id}(t+1) = & av_{id}(t) + b_1 r_1(t) [p_{id}(t) - x_{id}(t)] + \\ & b_2 r_2(t) [p_{gd}(t) - x_{id}(t)], \end{aligned} \quad (1)$$

$$x_{id}(t+1) = x_{id}(t) + v_{id}(t+1). \quad (2)$$

其中: a 为惯性权重; 加速常数因子 b_1, b_2 为非负常数; r_1, r_2 为两个在 $[0, 1]$ 范围内变化的随机数; v_{id} 为微粒 i 在第 d 维的速度, 范围是该维的最小速度 $v_{\min, d}$ 与最大速度 $v_{\max, d}$ 之间.

在式(1)中, 如果 $x_{id} = p_{id} = p_{gd}$, 即微粒 i 的当前位置是它经历过的最好位置, 也是整个微粒群迄今为止经历过的最好位置, v_{id} 的变化仅仅依赖于 av_{id} . 当一个微粒接近全局最优解时, 其速度将接近于零, 意味着算法最终使所有的微粒在最优解附近都停止运动, 不能保证收敛于全局最优. 为克服这种缺陷, Van den Bergh 对基本 PSO 算法进行了如下改进^[5], 称之为一般收敛 PSO (GCPSO).

令 X_T 是全局最优微粒, 则 X_T 的速度和位置更新方式重新定义为

$$\begin{aligned} v_{T,d}(t+1) = & \\ & -x_{T,d}(t) + p_{g,d}(t) + av_{T,d}(t) + \end{aligned}$$

$$\eta(t) (1 - 2r_2(t)), \quad (3)$$

$$x_{T,d}(t+1) =$$

$$p_{g,d}(t) + av_{T,d}(t) + \eta(t) (1 - 2r_2(t)). \quad (4)$$

其中: $-x_{T,d}$ 项将微粒的位置进行了重置, $av_{T,d}$ 项隐含了重新搜索的方向, 而 $\eta(1 - 2r_2)$ 项保证了重新搜索时微粒的速度不为零.

2.2 基本概念

定义 1 具有某些相同飞行特征的微粒个体集合称为种群, 种群搜索的空间是解空间的一个子空间, 称为种群小生境. 体现种群集体飞行有别于其他种群的位置或速度分量称为种群飞行特征, 简称为种群特征, 它是个体微粒所属种群小生境的唯一标识. 在此基础上进行种群小生境划分. 设种群小生境 P_j 的个体微粒 X_i 表示为 $X_i = \{\rho_j, x_{i,1}, \dots, x_{i,l}\}$, 其中, ρ_j 表示特定种群小生境的位置特征分量, $x_{i,l}$ 表示个体在种群小生境中的自由位置分量. 显然, 对于 ρ_j 的不同定义, 对应于一系列不相交的种群小生境, 所有这些种群小生境的并集构成了完整的解空间.

定义 2 种群特征分量 ρ_j ($j = 1, 2, \dots, n$, 其中 n 为预定的种群数) 的进化, 称为种群小生境进化, 简称种群进化. 种群小生境进化相对个体进化要漫长得多, 所以其进化周期 T 在这里用个体层进化代数的正整数表示. 用两次种群层进化操作间该小生境中最优个体微粒适应值提高的效率表示种群小生境的进化能力, 即

$$E(t) = \frac{f_{\max}(t) - f_{\max}(t-T)}{f_{\max}(t)}. \quad (5)$$

其中 $f_{\max}(t)$ 表示进化到第 t 时该小生境中最优个体微粒的适应值. 取小正数 ϵ 为种群进化能力阈值, 若第 t 代时种群小生境的进化能力 $E(t) > \epsilon$, 则称该小生境为成长种群小生境, 反之, 称之为成熟种群小生境.

定义 3 进入成熟期的种群小生境中的最优个体称为超级个体, 由一定数量超级个体组成的集合叫超级个体集合; 若成熟种群向超级个体集合提供超级个体时, 超级个体容量已达到设定值, 则用新进入的超级个体代替原来适应值最低的超级个体; 然后重新对超级个体的适应值进行排序.

2.3 算法设计

在以上基本概念定义的基础上设计小生境微粒群算法. 设 ρ_j 代表种群小生境 j 的位置特征分量, $P_j X_i$ 代表种群小生境 P_j 中第 i 个微粒目前的自由位置矢量, $P_j Y_i$ 表示该微粒目前为止经过的最好位置, $P_j \hat{Y}$ 表示小生境 P_j 中所有微粒到目前为止经历过的最好位置; P_j 中的第 i 个微粒可表示为 $b(\rho_j,$

$P_j X_i$), 其适应值函数表示为 $f(b(\rho_j, P_j X_i))$. 则种群小生境微粒群算法结构可表述如下:

```

初始化产生  $n$  个种群  $b(\rho_j, P_j X_1, P_j X_2, \dots, P_j X_m)$ ,  $j = 1, \dots, n; i = 1, \dots, m$ 
repeat
  for swam  $j$  (1, ...,  $n$ )
    for particle  $i$  (1, ...,  $m$ )
      if  $f(b(\rho_j, P_j X_i)) < f(b(\rho_j, P_j Y_i))$ 
        then  $P_j Y_i = P_j X_i$ 
      if  $f(b(\rho_j, P_j X_i)) < f(b(\rho_j, P_j Y))$ 
        then  $P_j Y = P_j X_i$ 
    endfor
    利用式(1) ~ (3) 更新微粒  $b(\rho_j, P_j X_i)$ 
    if  $\text{mod}(t, T) = 0$ 
      then 利用式(1) ~ (3) 更新  $\rho_j$ 
      if  $E_j(t) < \epsilon$ 
        then  $P_j$  进入超级个体集合
        超级个体重新排序
    endfor
  until 符合算法停止条件
    
```

3 SNPSO 算法收敛性分析

根据定义 1, 以种群特征分量 ρ_j 为标识划分出的一系列不相交小生境, 其并集 $\{b(\rho_j, P_j X_i | j = 1, \dots, n; i = 1, \dots, m)\}$ 构成了完备的解空间

定理 1 当参数 $|a| < 1, b > 0, 2a - b + 2 > 0$ 时, GCPSO 收敛于种群小生境 P_j .

根据定义 1, 以种群特征分量 ρ_j 为标识划分出的一系列不相交小生境, 其并集 $\{b(\rho_j, P_j X_i | j = 1, \dots, n; i = 1, \dots, m)\}$ 构成了完备的解空间

IC Trelea^[7] 将 GCPSO 的微粒 i 在 d 维上的运动看作一个离散时间动态系统, 证明了当

$$a < 1, b > 0, 2a - b + 2 > 0, \quad (6)$$

系统稳定在平衡点(即系统收敛), 此亦即 GCPSO 的收敛条件

GCPSO 在 (a, b) 平面上的收敛域如图 1 所示, 仅且仅当系数 a, b 落在三角形内时, 微粒的运动收敛于其平衡点

定理 2 种群小生境微粒群算法(SNPSO) 以概率 1 收敛于全局最优

证明 设问题 P 为 SNPSO 的优化对象, f^* 为其全局最优解的适应值, 即

$$f^* = \max\{f(x) | x \in P\}.$$

又设超级个体集合为 $\{f_1^*(x), f_2^*(x), \dots, f_N^*(x)\}$, 由定理 1 知, 选择合适的参数 a, b 时, SNPSO 中各小生境的 GCPSO 进化以概率 1 收敛于该小生境的超级个体 若设

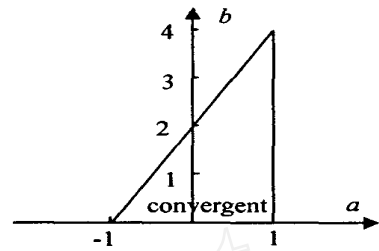


图 1 GCPSO 在 (a, b) 平面上的收敛域

$$z(x) = \max\{f_1^*(x), f_2^*(x), \dots, f_N^*(x)\},$$

则定理 2 可表述为

$$\lim_t P(z(x) = f^*) = 1.$$

由 SNPSO 小生境的划分对解空间的完备性知道, 至少存在某一小生境 P^* , 当进化成熟以后其中的最优个体的适应值等于全局最优个体的适应值 f^* , 即

$$f^* = \max\{f(x) | x \in P^*\}. \quad (7)$$

因为在 P^* 进化成熟以后, 其中的最优个体即是一个超级个体, 所以

$$\begin{aligned} & \lim_t \max\{f(x) | x \in P^*\} = \\ & \lim_t \max\{f_1^*(x), f_2^*(x), \dots, f_N^*(x)\} = \\ & \lim_t z(x). \end{aligned} \quad (8)$$

由式(7)和(8), 得

$$\begin{aligned} & \lim_t P(z(x) = f^*) = \\ & \lim_t P(\max\{f(x) | x \in P^*\} = \\ & f^*) = 1. \end{aligned}$$

命题得证

4 基于种群小生境 PSO 的前向神经网络设计

4.1 前向神经网络结构

因含有一个隐层的 3 层前向神经网络能够以任意精度逼近有界非线性函数^[16], 故这里采用 3 层前向神经网络且为全力连接 网络的输入输出节点数一般由应用问题确定, 因此, 网络的复杂程度主要取决于隐层的节点数

假定隐层节点数 n_{hid} 足够多的初始网络隐层激励函数和输出层分别为 $\psi(x)$ 和 $\phi(x)$, 给定样本集 $\Omega = \{P_i, q_i | 1 \leq i \leq N, P_i \in R^r, q_i \in R\}$, $P_i = (p_{i,1}, p_{i,2}, \dots, p_{i,r})$, 则前向神经网络的输出为

$$\hat{q}_i = k \phi \left[\sum_{n=1}^{n_{hid}} V_{n,i} \psi \left(\sum_{m=1}^r W_{m,n} P_{i,m} \right) \right]. \quad (9)$$

其中: W, V 为权值, 满足 $-1 < W, V < 1; k = \max(q_1, q_2, \dots, q_N)$.

定义一个布尔变量 $L = (a_1, a_2, \dots, a_n, \dots, a_{n_{hid}})$, 其中元素 a_n 为 1 或 0, 分别对应隐层节点存在或不存在 作如下运算:

$$\rho = \prod_{n=1}^{n_{hid}} a_n 2^{-n_{hid}}, n = 1, \dots, n_{hid}, \quad (10)$$

则必有 $0 < \rho < 1$. 这样, 当 ρ 在 $(0, 1)$ 区间内任取一随机数, 都有一个 n_{hid} 维的布尔随机向量 L 与之对应, 也就是说, 前向网络的一层节点可由随机数 ρ 确定的布尔向量 L 确定. 这时, 前向网络的输出表达式可写为

$$\hat{q}_i = k \Phi \left[\sum_{n=1}^{n_{hid}} V_{n,i} \times a_n \times \Phi \left(\sum_{m=1}^m W_{m,n} P_j \right) \right], \quad (11)$$

其中 $a_n \in \{0, 1\}$.

神经网络的性能指标取^[17]

$$J(a_1, a_2, \dots, a_{n_{hid}}, V, W) = \ln e + \lambda \frac{N_{hid}}{N} = \ln E + \frac{\lambda}{N} \sum_{n=1}^{n_{hid}} a_n \quad (12)$$

其中: 第 1 项表示模型输出与样本的拟合程度, $e = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^T (y_i - \hat{y}_i)$; 第 2 项用于惩罚网络模型的复杂度; λ 为调节约束强度因子; n_{hid} 为隐层最大节点数; N 为训练样本数

4.2 基于种群小生境 PSO 的前向神经网络设计策略

1) 编码 对于一个 $r \times n_{hid} \times 1$ 网络, 可以表示为微粒

$$X_i = \{\rho_i, W_i^{n,m}, V_i^n\}$$

其中: $n = 1, 2, \dots, n_{hid}$; $m = 1, 2, \dots, r$; $W_i^{n,m}$ 为输入层第 m 个节点与隐层第 n 个节点之间的连接权值; V_i^n 为第 n 个隐层节点与输出节点之间的连接权; ρ_i

$(0, 1)$, 由 ρ_i 唯一确定的布尔向量 $L = (a_1, a_2, \dots, a_{n_{hid}})$ 确定了唯一的隐层节点数

2) 初始化小生境 为保证初始化的小生境均匀分布在解空间, 采用 Faure 序列产生均匀分布在 $(0, 1)$ 区间的 M 个随机数 $\rho_i, i = 1, 2, \dots, M$, 用以确定 M 个初始化小生境. 每个小生境中随机产生 $3 \sim 5$ 个 $n + m \times n$ 阶序列 $(b_1, b_2, \dots, b_j, \dots, b_{n+m \times n}), b_j \in (0, 1), j = 1, 2, \dots, n + m \times n$.

3) 适应值选择 个体微粒 X_i 的适应值按下式计算:

$$f_i = J(a_1, a_2, \dots, a_{n_{hid}}, V_i, W_i). \quad (13)$$

4) 小生境内微粒群进化 为避免初始化的小生境内微粒个体相互很近时小生境内微粒群进化收

敛到局部最优解, 这里用 GCPSO 算法代替标准 PSO 算法. 小生境内个体微粒 X_i 运动公式为: 当 X_i 不是该小生境内当前最优微粒时, 其运动迭代按式 (1) 和 (2) 进行; 当 X_i 为该小生境内当前最优微粒时, 其运动迭代按式 (3) 和 (4) 进行.

5) 成熟小生境的种群层进化 因为种群小生境是均匀分布在解空间内的, 所以成熟小生境的种群层进化局部进化 PSO, 即 ρ_i 分量的运动公式为

$$v_{i\rho}(t) = av_{i\rho}(t) + b_1 r_1(t) (p_{i\rho}(t) - \rho_i(t)), \quad (14)$$

$$\rho_i(t+1) = \rho_i(t) + v_{i\rho}(t+1). \quad (15)$$

6) 算法终止条件与最优解 以超级个体集合中适应值最好的个体微粒作为当前全局最优解. 当该最优解的适应值达到预定要求或进化代数达到预定代数时, 算法终止并输出该最优解. 将该最优解按式 (10) 和 (11) 解码为前向神经网络模型.

5 算法仿真研究

采用的前向网络初始结构为 4-25-1, 即 $n_{hid} = 25, r = 4$, 因此, 网络结构共有 125 个权参数和一个确定隐层节点数的变量 ρ_i , 分别取隐层和输出层激励函数为

$$\Phi(x) = \frac{1}{1 + e^{-\alpha x}}, \phi(x) = \frac{1}{1 + e^{-\beta x}}$$

其中: $\alpha = 10, \beta = 15$. 式 (12) 中的参数 $\lambda = 0.2$.

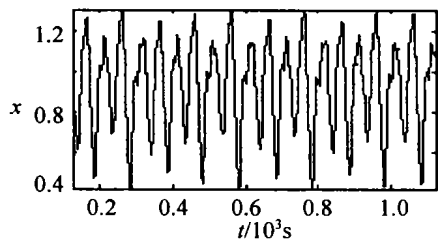
从 Mackey-Glass 混沌时间序列

$$x(t+1) = (1 - B)x(t) + A \frac{x(t - \tau)}{[x(t - \tau)]^{10}},$$

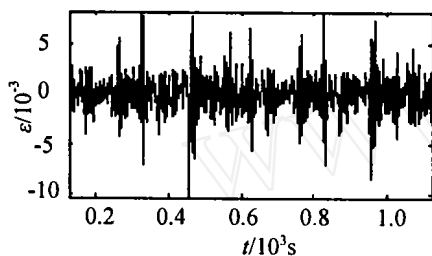
$$A = 0.2, B = 0.1, \tau = 17.$$

中, 按如下格式抽取 1 000 对输入输出数据: $[x(t - 18), x(t - 12), x(t - 6), x(t)]$, 其中 $t = 118 \sim 1117$. 前 500 个数据对用于对前向网络进行训练, 而剩下的 500 个数据对作为检验数据. 取参数 $a = 1.2, b_1 = b_2 = 0.1, r_1 = r_2 = 0.5$, 算法运行 1 000 步以后, 得到优化设计的一个 4-6-1 网络及其各权值. 由图 1(a)、图 (b) 可以看出训练数据与检验数据两者的期望值和预测值几乎一样, 两者的差别只在很小数量级上才能区分.

分别用 BPSO-BP 算法、BPSO-PSO 算法和本文的 SNPSO 算法对前向网络进行 1 000 步训练, 训练结果如表 1 所示. 从表 1 的训练结果可以看出, 本文的 SNPSO 使用的训练时间几乎是其他两种算法的一半, 而训练精度并没有降低. 另外, 从检验误差也可以看出, 采用 SNPSO 训练的网络泛化能力增加了很多.



(a) 混沌序列和提前 6 步预测



(b) 预测误差 ϵ

图 2 预测混沌序列以及期望值与输出值的误差 ϵ

表 1 BPSO-BP, BPSO-PSO 和 SNPSO 算法训练结果比较

算 法	网络 规模	训练时 间/s	平均误差	
			训练误差	检验误差
BPSO-BP	10	246.770	7.3685×10^{-5}	3.5376×10^{-4}
BPSO-PSO	9	167.174	1.3352×10^{-7}	1.9672×10^{-4}
SNPSO	6	91.326	1.5631×10^{-7}	-9.9720×10^{-6}

6 结 论

神经网络的进化设计一直是神经网络和进化计算中的研究热点之一。本文根据自然界中鱼鸟等所具有的种群特征, 借鉴分层递阶编码的思想, 构造出一种基于种群特征分量的小生境微粒群优化算法, 用于前向神经网络的进化设计, 具有以下特点: 采用小生境内个体微粒自由运动特征的微粒群进化算法与小生境种群运动特征分量的微粒群进化分层递阶进行, 以及各小生境并行进化的方法, 有利于在并行计算机上运行; 神经网络结构和连接权值分层递阶进化思想模仿了人工设计神经网络时连接权值调节-网络结构调节-连接权值调节这样的循环过程, 容易理解; 微粒群算法简单, 计算量小; 小生境内子微粒群的进化和成熟小生境种群特征分量的进化分别采用全局收敛 PSO 和局部 PSO, 符合神经网络的参数调整特征

参考文献 (References)

[1] Kennedy J, Eberhart R. Particle Swarm Optimization [A]. *Proc IEEE Int Conf on Neural Networks* [C]. Perth, 1995: 1942-1948
 [2] Kennedy J. The Particle Swarm: Social Adaptation of

Knowledge [A]. *Proc IEEE Int Conf on Evolutionary Computation* [C]. Indianapolis, 1997: 303-308
 [3] Shi Yuhui, Eberhart R. A Modified Particle Swarm Optimizer [A]. *Proc IEEE Int Conf on Evolutionary Computation* [C]. Anchorage, 1998: 67-73
 [4] F van den Bergh, Engelbrecht A P. A New Locally Convergent Particle Swarm Optimization [A]. *Proc IEEE Int Conf on System, Man and Cybernetics* [C]. Hammamet, 2002
 [5] Clerc M, Kennedy J. The Particle Swarm — Explosion, Stability, and Convergence in a Multidimensional Complex Space [J]. *IEEE Trans on Evolutionary Computation*, 2002, 6(2): 58-73
 [6] Mendes R, Kennedy J, Neves J. The Fully Informed Particle Swarm: Simpler, Maybe Better [J]. *IEEE Trans on Evolutionary Computation*, 2004 8(6): 204-210
 [7] Ian Cristian Trelea. Particle Swarm Optimization Algorithm: Convergence Analysis and Parameter Selection [J]. *Information Processing Letters*, 2003, 85: 317-325
 [8] F van den Bergh. Particle Swarm Weight Initialization in Multi-layer Neural Networks [A]. *Development and Practice of Artificial Intelligence Techniques* [C]. Durban, 1999: 41-45
 [9] F van den Bergh, Engelbrecht A P. Cooperative Learning in Neural Networks Using Particle Swarm Optimizers [J]. *South Africa Computer J*, 2000, 26(11): 84 - 90
 [10] F van den Bergh, Engelbrecht A P. Training Product Unit Networks Using Cooperative Particle Swarm Optimizers [A]. *Proc IEEE Int Conf on Neural Networks* [C]. Washington DC, 2001.
 [11] Tsou D, MacNish C. Adaptive Particle Swarm Optimization for High-dimensional Highly Convex Search Spaces [A]. *Proc IEEE Int Conf on Intelligence Symposium* [C]. Canberra, 2003
 [12] Agrafiotis D K, Cedeno W. Feature Selection for Structure-activity Correlation Using Binary Particle Swarm [J]. *Med Chem*, 2002, 45(5): 1098-1016
 [13] Wang ZW, Durst GL, Eberhart R C. Particle Swarm Optimization and Neural Network Application for QSAR [A]. *Proc IEEE Int Conf on Parallel and Distributed Processing Symposium* [C]. New Mexico, 2004
 [14] Brits R, Engelbrecht A P, Fan den Bergh. A Niching Particle Swarm Optimizer [A]. *Proc Conf on Simulated Evolution and Learning* [C]. Singapore, 2002

(下转第 991 页)

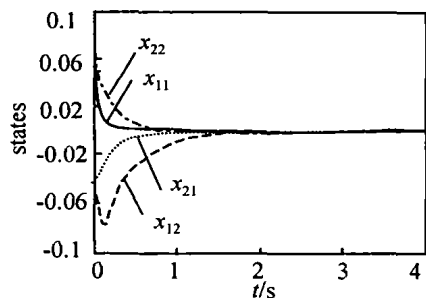


图1 第1,2个子系统的状态响应曲线

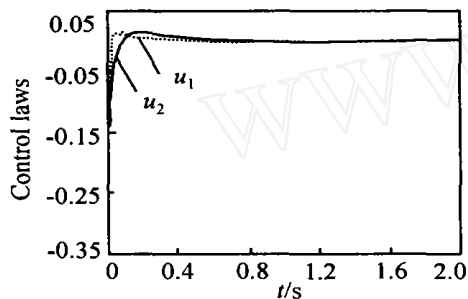


图2 第1,2个子系统的控制输入

5 结 语

本文研究了参数不确定T-S模糊交联系统的分散保成本控制问题。此方法可通过解线性矩阵不等式一次得到分散状态反馈保成本控制器,它简化了采用传统方法分步进行求解控制器的繁琐过程。算例显示了所给方法的有效可行。

参考文献(References)

[1] 张庆灵. 广义大系统的分散控制与鲁棒控制[M]. 西安: 西北工业大学出版社, 1997.
(Zhang Q L. *Decentralized Control and Robust Control for Large-Scale Descriptor Systems* [M]. Xi'an: Northwestern Polytechnical University Press, 1997.)

[2] Park J H, Lee S G. Robust Decentralized Stabilization of Uncertain Large-scale Discrete-time Systems[J]. *Int J of Systems Science*, 2002, 33(8): 649-654

[3] Yu L, Chu J. An LM I Approach to Guaranteed Cost Control of Linear Uncertain Time-delay Systems[J]. *Automatica*, 1999, 35(6): 1155-1159

[4] 陈国定, 俞立, 杨马英. 一类关联不确定离散系统的分散保成本控制[J]. *控制理论与应用*, 2002, 19(6): 909-911
(Chen G D, Yu L, Yang M Y. Decentralized Guaranteed Cost Control of a Class of Interconnected Uncertain Discrete-time Systems [J]. *Control Theory and Applications*, 2002, 19(6): 909-911.)

[5] Jadbabaie A, Jamshidi M, Titli A. Guaranteed-cost Design of Continuous-time Takagi-Sugeno Fuzzy Controllers via Linear Matrix Inequalities[A]. *IEEE Int Conf on Fuzzy Systems*[C]. Anchorage, 1998: 268-273

[6] Guan X P, Chen C L. Delay-dependent Guaranteed Cost Control for T-S Fuzzy Systems with Time Delays [J]. *IEEE Trans on Fuzzy Systems*, 2004, 12(2): 236-249

[7] Yang Y J, Yue D, Guo X J. Guaranteed-cost Design Based on T-S Fuzzy Model[A]. *Proc of the World Congress on Intelligent Control and Automation* [C]. Shanghai, 2002: 1862-1866

[8] Tanaka K, Sugeno M. Stability Analysis and Design of Fuzzy Control Systems [J]. *Fuzzy Sets and Systems*, 1992, 45(2): 135-156

[9] Xie L H. Output Feedback H_∞ Control of Systems with Parameter Uncertainty [J]. *Int J of Control*, 1996, 63(4): 741-750

(上接第985页)

[15] Brits R, Engelbrecht A P, Fan den Bergh. Scalability of Niche PSO [A]. *Proc IEEE Int Conf on Intelligence Symposium* [C]. Indianapolis, 2003: 228-234

[16] Fredric M Ham, Ivica Kostanic. *Principles of Neurocomputing for Science and Engineering* [M]. McGraw Hill, 2001.

[17] 李祥飞, 邹恩, 张泰山. 基于混沌变量的前向神经网络

络结构优化设计[J]. *控制与决策*, 2003, 18(6): 705-707.

(Li X F, Zou N, Zhang T S. Optimization Design of Feed-forward Neural Network Structure Based on Chaos Variables [J]. *Control and Decision*, 2003, 18(6): 705-707.)