第 34 卷　第 2 期　　电 子 科 技 大 学 学 报　　Vol.34　No.2
2005 年 4 月　　Journal　of　UEST　of　China　　Apr. 2005

线性矩阵不等式问题的进化计算解决方法

崔梦天[1]，张世禄[1]，赵海军[2]

(1. 西华师范大学数学与信息学院　四川 南充　637002;　2. 西华师范大学计算机学院　四川 南充　637002)

【摘要】针对决定模糊控制中稳定性的线性矩阵不等式问题，提出了用进化计算来解决模糊控制中线性矩阵不等式的新算法。实验证明，该算法解“用于实现模糊控制的增益调度和稳定性的线性矩阵不等式”是有效的。

关　键　词　进化计算;　线性矩阵不等式;　正定矩阵;　半正定矩阵

中图分类号　O241.6;　TP368.5　　　文献标识码　A

Solution to Linear Matrix Inequalities Via Evolutionary Computation

CUI Meng-tian，ZHANG Shi-lu，ZHAO Hai-jun

(1. School of Mathematics and Information ,China-West Normal University　Sichuan Nanchong　637002

2. School of Computer Science ,China-West Normal University　Sichuan Nanchong　637002)

Abstract　This paper presented a new ideas for resolving linear matrix inequalities using evolutionary computation, which decides the stability of the fuzzy model-based control. Experiments proved it is a effective algorithm to achieve gain scheduling and stability checking of fuzzy controllor. In the end ,two examples conducted in simulations demonstrated the effectiveness of the proposed algorithm for solving matrix inequalities.

Key words　evolutionary computation;　linear matrix inequalities;　positive definite matrices; positive semi-definite matrices

模糊集合理论提供了系统的、以语言表示不确定和模糊信息的计算工具，通过使用由隶属函数表示语言变量，它还可以进行数值计算。合理选择模糊规则是模糊推理系统的关键因素，并有效地对特定应用领域中的人类专门知识进行建模[1]。尽管模糊推理系统拥有模糊IF-THEN格式的结构化知识表示，但仍缺少对变化的外部环境进行适应的能力。而进化计算的根本优点是对外部很强的适应性和提供优化算法以适应要解决的问题的需求以及对优化算法的各种参数进行调谐，以便能更好地执行算法。

基于模型的模糊控制的一个核心问题是它的稳定性[2]。一般都是通过线性矩阵不等式来实现模糊控制增益调度和稳定性检测问题。为最优解决这些线性矩阵不等式，必须获得该问题的有效计算。针对该问题，本文引入了进化计算，提出了用进化计算来解决矩阵不等式的新算法。

1　基于模型的模糊控制的稳定性

对于一个模糊模型系统，模糊模型包含一组IF-THEN规则。在实际中，大多数有效系统是非线性的。为能在线性模糊模型上应用，必须对这些非线性模型进行改造。为此，设计了下列模糊控制器，通过采用引入一个反馈增益K_i(i=1，2，…，r)的方法来实现对非线性模型的改造。控制规则i如下：

收稿日期：2004－05－08

作者简介：崔梦天(1972－)，女，硕士生，讲师，主要从事算法和计算机辅助教育的教学方面的研究；张世禄(1944－)，男，教授，硕士生导师，主要从事程序设计方法学、算法和计算机辅助教育的教学方面的研究.

$$\begin{cases} x_i \text{ is } \boldsymbol{M}_{i1} \quad u(t) = -\boldsymbol{K}_1[x(t) - x_r] + u_r \\ x_i \text{ is } \boldsymbol{M}_{ii} \quad u(t) = -\boldsymbol{K}_i[x(t) - x_r] + u_r \\ x_i \text{ is } \boldsymbol{M}_{ir} \quad u(t) = -\boldsymbol{K}_r[x(t) - x_r] + u_r \end{cases} \tag{1}$$

式中 x_r 是一个状态参考轨迹；u_r是响应输入轨迹；$\boldsymbol{K}_i$是反馈增益矩阵。

1.1 基本稳定性条件

如果局部反馈规则通过$u(t)= -\boldsymbol{K}_i(t)$给出，对于模糊模型来说，它的稳定性条件为：

$$\boldsymbol{G}_{ii}{}^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{G}_{ii} < 0, \qquad (\frac{\boldsymbol{G}_{ij} + \boldsymbol{G}_{ji}}{2})^T \boldsymbol{P} + \boldsymbol{P}(\frac{\boldsymbol{G}_{ij} + \boldsymbol{G}_{ji}}{2}) \leq 0, \qquad i < j \tag{2}$$

式中 反馈增益用$\boldsymbol{K}_i(i=1,2,\cdots,r)$来表示，$\boldsymbol{G}_{ij} = \boldsymbol{A}_i - \boldsymbol{A}_i\boldsymbol{K}_j$ 。式(2)变为：

$$\boldsymbol{G}_{ii}{}^T\boldsymbol{P} + \boldsymbol{P}\boldsymbol{G}_{ii} + (s-1)\boldsymbol{Q} + 2a\boldsymbol{P} < 0 \tag{3}$$

$$(\frac{\boldsymbol{G}_{ij} + \boldsymbol{G}_{ji}}{2})^T \boldsymbol{P} + (\frac{\boldsymbol{G}_{ij} + \boldsymbol{G}_{ji}}{2}) - \boldsymbol{Q} + 2a\,\boldsymbol{P} \leq 0, i < j \tag{4}$$

式中 $\boldsymbol{P}$为正定矩阵，$\boldsymbol{Q}$为半正定矩阵，$a>0$。

1.2 用线性矩阵不等式进行稳定性分析

为了说明进化计算，本文提供了一系列线性矩阵不等式。式(3)和(4)经过重新整理，用松弛的稳定性情况，列出最优化式如下：最大化a $\boldsymbol{X}$，$\boldsymbol{Y}$，$\boldsymbol{M}_1$，$\cdots$，$\boldsymbol{M}_r$ 使 $\boldsymbol{X}>0$，$\boldsymbol{Y}\geq 0$

$$-\boldsymbol{X}\boldsymbol{A}^{\mathrm{T}}{}_i - \boldsymbol{A}_i\boldsymbol{X} + \boldsymbol{M}^{\mathrm{T}}{}_i\boldsymbol{B}^{\mathrm{T}}{}_i + \boldsymbol{B}_i\boldsymbol{M}_j - (s-1)\boldsymbol{Y} - 2a\boldsymbol{X} > 0，i<j \tag{5}$$

$$2\boldsymbol{Y} - \boldsymbol{X}\boldsymbol{A}^{\mathrm{T}}{}_i - \boldsymbol{A}_i\boldsymbol{X} - \boldsymbol{X}\boldsymbol{A}^{\mathrm{T}}{}_j - \boldsymbol{A}_j.\boldsymbol{X} + \boldsymbol{M}^{\mathrm{T}}{}_j\boldsymbol{B}^{\mathrm{T}}{}_i + \boldsymbol{B}_i\boldsymbol{M}_i + \boldsymbol{M}^{\mathrm{T}}{}_i\boldsymbol{B}^{\mathrm{T}}{}_j + \boldsymbol{B}_j\boldsymbol{M}_i - 4a\boldsymbol{X} > 0 \tag{6}$$

式中 $\boldsymbol{X} = \boldsymbol{P}^{-1}$；$\boldsymbol{M}_i = \boldsymbol{K}_i\boldsymbol{X}$；$\boldsymbol{Y} = \boldsymbol{X}\boldsymbol{Q}$

2 正定和半正定矩阵的表达法

这里描述了一种用进化计算解方程式(2)～(6)的方法。描述控制系统的稳定性情况的线性矩阵不等式，它包含一组正定矩阵$\boldsymbol{P}$和半正定矩阵$\boldsymbol{Q}$。令$\boldsymbol{Q}_i\,(i=1, 2, \cdots, N)$为正定或半正定矩阵，这样

$$\boldsymbol{P}=\boldsymbol{Q}_i(x) > 0, \quad i= 1, 2, \cdots, n_1 \tag{7}$$

$$\boldsymbol{Q}=\boldsymbol{Q}_i(x) \geq 0，i= n_1+1, 2, \cdots, N \tag{8}$$

式中 N表示确保系统稳定的正定矩阵$\boldsymbol{P}$加上半正定矩阵$\boldsymbol{Q}$的数量；$\boldsymbol{Q}_i\,(i=1, 2, \cdots, N)$是矩阵。令$\boldsymbol{Q}_i\,(x)$为$\boldsymbol{Q}_i = \begin{bmatrix} \boldsymbol{N}(x) & 0 \\ 0 & \boldsymbol{M}(x) \end{bmatrix}$，这里：

$$\boldsymbol{N}(x) = \begin{bmatrix} \boldsymbol{Q}_1 & & 0 \\ & \mathrm{O} & \\ 0 & & \boldsymbol{Q}_{n_1} \end{bmatrix}，\boldsymbol{M}(x) = \begin{bmatrix} \boldsymbol{Q}_{n_1+1}(x) & & 0 \\ & \mathrm{O} & \\ 0 & & \boldsymbol{Q}_{\boldsymbol{N}}(x) \end{bmatrix}$$

令$L_{\min}(\boldsymbol{Q}\ (x))(i=1, 2, \cdots, N)$为$\boldsymbol{Q}_i(x)(i=1, 2, \cdots, N)$的最小特征值，那么对于$\boldsymbol{Q}_i(x)(i=1,2, \cdots, n_1)$成为正定的必要条件是：

$$L_{\min}(\boldsymbol{Q}_i(x))>0 \qquad i=1，2，\cdots，n_1 \tag{9}$$

对于$\boldsymbol{Q}_i(x)(i=n_1+1, 2, \cdots, N)$成为半正定的必要条件是：

$$L_{\min}(\boldsymbol{Q}_i(x)) \geq 0 \qquad i=n_1+1，\cdots，N \tag{10}$$

式中 $\boldsymbol{Q}_i(x)(i=1, 2, \cdots, N)$是对称矩阵。很容易得到，如果$\boldsymbol{Q}\ (x)$满足条件：

$$L_{\min}(\boldsymbol{Q}_i(x))>0 \tag{11}$$

则 矩阵$\boldsymbol{Q}\ (x)\geq 0$；而如果$\boldsymbol{Q}_i(x)$满足条件：

$$L_{\min}(\boldsymbol{Q}_i(x)) \geq 0 \tag{12}$$

则 矩阵$\boldsymbol{Q}_i(x)>0$；通过GAs算法解不等式矩阵式(2)～(6)。$\boldsymbol{K}_i$，$\boldsymbol{P}$和$\boldsymbol{Q}$ 以及与控制系统相关的其他参数都可以在基因中被编码[3, 4]，并提前设定好种群代的数量，完成了期望值a(或其他最优化目标)后，最优化过程将终止。在第3部分，用进化算法表达。

3　进化计算算法的实现

GAs是不严格地建立在自然选择和自然进化概念基础上的一种非导数的随机优化技术。从本质上说，它们是并行搜索算法，是在自然界基因中发现的操作，通过参数空间来指导搜索。无论理论上还是实践上，GAs都已被证明在复杂空间中，能提供强大的搜索能力，提供了有效的方法解决高效搜索的问题。

GAs算法是基于种群数量最优化技术。在GAs中，有3个基本操作：选择，交换，变异[4, 5]。选择是从当前一代生成一组新的种群。选择操作决定哪些个体参与生成下一代的个体。个体基因链按照它们的适应性函数值被复制在交配池中。交换处理分两步：1) 来自交配池的成员随机交配。2) 每对基因链按照下列方式交换：位置L沿着基因链在区间[1，l-1]随机被选择，这里，l是链的长度。两个链通过在位置K和L之间交换产生。变异，是基因链位置的值随机交替的过程。变异能防止整个种群在任何局部的任一位收敛到一个值，并且更为重要的是，它能防止种群收敛并停滞在任何局部点。因此，由交换操作中得到的好染色体不会丢失。这里，用进化计算算法解矩阵不等式。算法如下：

Begin

t=0;

Initialize $S(0):=\{a_1(0),\cdots a_u(0)\}\in I^u$

Evaluate Fitness:

$$\begin{cases} z\,, & if \sum_{k=1}^{N} S_k \geq 0 \\ \sum_{k=1}^{N} S_k \geq 0, & \text{otherwise} \end{cases}$$

Where　$S_k = L_{\min}(\boldsymbol{Q}_k(x))(k=1，2，\cdots，N)$

While (Fitness≠termination condition)Do

Begin

recombine $a_k'(t):=r(S(t))$

mutate　$a_k''(t):=m(a_k'(t))$

evaluate $S'(t):Ft(x(t)) \cup Ft(x'(t))$

select $S(t+1)$ from $S(t)$:

IF　$(Ft(x(t)) < Ft(x'(t)))$

$x(t+1) = x'(t)$

Else

$x(t+1) = x(t)$

$t=t+1$

end

end

解式(2)(3)的一般特征值问题可看成一种研究最佳$\boldsymbol{P}$和$\boldsymbol{Q}$的方法。用进化计算算法决定正定矩阵$\boldsymbol{X}$，半正定矩阵$\boldsymbol{Y}$。这里a_1，a_2，…，a_u是线性矩阵不等式的$\boldsymbol{P}$、$\boldsymbol{Q}$和$\boldsymbol{K}_i$的元素，例如式(2)～(6)。在种群$S(t)$中，被重组的种群$a_k'(t)$将通过重组操作r获得。变异操作器m将生成$a_k'(t)$，$S'(t)$是这两个操作之后，由$x'(t)$个体组成的新的种群。下一个种群中的个体，如$x(t+1)$，将通过估计适应性函数$F(t)$来选择。通过定义下列适应性函数，来实现矩阵不等式的一般特征值的求解。适应性函数为：

$$\begin{cases} z\,, & if \sum_{k=1}^{N} S_k \geq 0 \\ \sum_{k=1}^{N} S_k \geq 0, & \text{其他} \end{cases} \tag{13}$$

式中　z 是常数，P_k是包含$L_{\min}(\boldsymbol{Q}_k(x))$的罚因子。

4　例　证

例1　非线性系统：

$$\begin{cases} x_1(t+1) = -2.1x_1^2 + x_2 - 1 + u \\ x_2(t+1) = 0.2x_1 \end{cases} \tag{14}$$

下列给出了在固定点(0.425 0，0.241 3)周围的三个线性模型。

$$\boldsymbol{A}_1 = \begin{bmatrix} -1.542\,0 & 1 \\ 0.5 & 0 \end{bmatrix} \quad \boldsymbol{B}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \boldsymbol{A}_2 = \begin{bmatrix} -1.521\,1 & 1 \\ 26.489\,7 & 0 \end{bmatrix} \quad \boldsymbol{B}_2 = \begin{bmatrix} 1 \\ 0 \end{bmatrix} \quad \boldsymbol{A}_3 = \begin{bmatrix} -1.541\,8 & 1 \\ 0.5 & 0 \end{bmatrix} \quad \boldsymbol{B}_3 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

以上控制的增益调度问题能在式(3)(4)的稳定性条件下，可表示成一个最佳问题。上面所描述的进化计算最优化算法被应用在系统($\boldsymbol{A}_i x(t)+\boldsymbol{B}_i \boldsymbol{u}(t)$　i =1,2,3)，以便决定控制所期望的反馈增益。$\boldsymbol{P}$在25代之后可以得

到 $\boldsymbol{P}=\begin{bmatrix} 299.1256 & 82.4866 \\ 82.4866 & 710.4563 \end{bmatrix}$。

模糊控制的反馈增益：$\boldsymbol{K}_1$＝[5.887 7　5.493 1]，$\boldsymbol{K}_2$＝[7.826 6　2.470 8]，$\boldsymbol{K}_3$＝[3.779 8　3.123 2]。图1说明每一代响应的最佳适应性值。

例2　主要说明如何使用线性矩阵不等式来决定系统的增益。为此，本文用著名的倒立摆(也称“车-摆系统”)这个例子来评价所提出的方法的优点和有效性。非线性模型通过下列两个线性模型给出[5]。

$$\boldsymbol{A}_1=\begin{bmatrix} 0 & 1 & 0 \\ 9.8 & 0 & 1 \\ 0 & -10 & -10 \end{bmatrix},\ \boldsymbol{B}_1=\begin{bmatrix} 0 \\ 0 \\ 10 \end{bmatrix} \qquad \boldsymbol{A}_2=\begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & -10 & -10 \end{bmatrix},\ \boldsymbol{B}_2=\begin{bmatrix} 0 \\ 0 \\ 20 \end{bmatrix}$$

把式(5)(6)所提供最优化算法应用在这个系统，可得到P和Q如下：

$$\boldsymbol{p}=\begin{bmatrix} 3328.76 & 47.0789 & 2.2565 \\ 47.0789 & 98.0663 & 0.6499 \\ 2.2565 & 0.6499 & 0.0599 \end{bmatrix}, \qquad \boldsymbol{Q}=\begin{bmatrix} 6978.08 & 18.8664 & 0.6184 \\ 18.8664 & 95.7969 & 0.2937 \\ 0.6184 & 0.2937 & 0.01398 \end{bmatrix}$$

在a =2.206 8时，得到增益值：$\boldsymbol{K}_1$=[66.992 1，90.032 4，6.641 5]，$\boldsymbol{K}_2$=[64.712 8，77.765 7，11.356 9]。图2说明每一代响应的最佳适应性值。

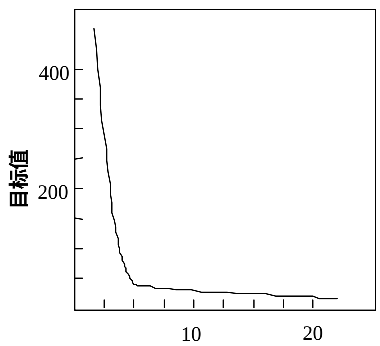

图1　例1的进化过程

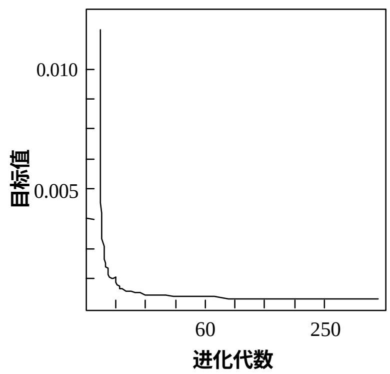

图2　例2的进化过程

5　结 束 语

本文介绍了用进化计算解关于模糊控制稳定性中的矩阵不等式问题的一种新观点。在模拟中的2个例子证明了所提出的方法。结果论述了用所提出的算法解矩阵不等式是有效的。

参 考 文 献

[1] 刘保踮. 随机规划与模糊规划[M]. 北京: 清华大学出版社, 1998

[2] aperelson A, Mcgregor D R. Genetic algorithm for inducing control rules for a dynamic system[C]. In Proceedings of the third international Conference on Genetic Algorithms, 1989, 177-182

[3] Goldberg D E. Genetic algorithms in search.optimization, and machine learning [J]. Fuzzy Sets and Systems. 1989, 6: 18-22

[4] 张智星. 神经—模糊与软计算[M]. 西安: 西安交通大学出版社, 2000

[5] Sanner R M, Slotine J J E. aussian networks for direct adaptive control[J]. IEEE Trans. Fuzzy sys. 1998 6(2),250-265

编　辑　刘文珍