计算机科学 1997 Vol. 24 No. 6

遗传优化模糊逻辑控制器*)

Optimization of Fuzzy Logic Controllers by Genetic Algorithms

胡 炜 沈 理

(中国科学院计算技术研究所 北京 100080)

摘 要 Fuzzy logic Controllers have been succesfully applied to a wide variety of practical problems. Designing FLCs by genetic algorithms is a very active research area. This paper details the methods of genetic optimizing domain-based FLCs and rule based FLCs respectively, and points out that optimization of FLCs includes two levels: optimization of FLCs structrue and optimization of FLCs' parameters. The paper also put forward some useful proposals finally.

关键词 Genetic algorithms, Fuzzy logic controllers, Optimization

一、引言

模糊逻辑控制器(FLC)已经广泛应用于工业过程、家电、运动控制中。典型的FLC由四部分组成:模糊化器、规则集、推理机和反模糊化器。FLC作为一种万能逼近器,适合控制各类非线性的、多输入的、无数学模型的系统。其主要优点表现在:(1)利用语言变量、语言项,直观易懂,符合人类的思考方式,易于利用专家已有的知识。(2)开发成本低、周期短。(3)FLC具有良好的鲁棒性和稳定性,对环境的改变有很强的适应能力。

FLC也有自己的问题。对于一个简单的控制对象,比如一个二输入一输出的系统,设计者能够根据经验总结出有效的规则,并大致确定输入输出变量的模糊划分。随着系统复杂度的提高,直观经验越来越难于获得,而且往往表达不清楚,难以直接利用。因此寻求一种自动设计和优化FLC的方法是亟待解决的问题。目前已经尝试过多种方法,如有教练和无教练的神经网络方法、模拟退火等,都取得了积极的效果。考虑到FLC的优化涉及大范围、多参数、复杂和不连续的搜索表面,人们想到了用遗传算法(Genetic Algorithm,简称GA)来进行优化,这是目前很活跃的研究领域。

遗传算法是一种并行的概率搜索方法,它模拟生物界的进化过程,利用精心设计的遗传算子,将问题的解一步步导向适应值(fitness)最优的区域。GA是适合设计FLC的。GA的运行仅由适应值驱动而不需要被优化对象的局部信息。FLC的适应值,根据控制的要求和设计者对FLC的要求,总是能得到的。另外优化FLC也正好合符GA的所谓"building block"假设[9],"building block"指长度较短的、性能较好的基因(gene)片段。模糊集的划分有相当的重叠,即相邻的隶属函数是相交的,这使得相邻的隶属函数之间能产生强烈的相互作用和组合效应,而远离的隶属函数之间则没有或只有极小的相互影响。因此若按顺序将这些模糊集排列起来形成GA的个体,良好的模糊集能容易地形成所谓的"building block";模糊规则集也具有类似的性质,考虑一个两输入一输出的规则表,不失一般性,假定任一输入都激活规则表中的四条规则,那么这四条规则是相邻的,且排列成正方形。若规则采用矩阵类型的编码,这四条规则就可能构成一个"building block",即使规则采用一维编码,也可以构成两组由两条规则组成的"building block"。

二、用GA设计FLC

FLC可以看作一种学习分类器系统。GA应用于学习分类器系统主要有两种方法:密歇根方法和匹兹堡方法[1]。前者将每一条规则作为一个个体,整个群体代表一个系统(一个控制器),系统性能的优劣由整个群体来决定。遗传算子作用于个体之间或单个个体上。个体的好坏由个体所获得的"分数"

*)863国家高技术研究发展计划资助项目。胡炜 博士研究生,研究方向:进化计算、模糊逻辑系统;沈理,研究员、博士生导师,研究方向:软计算、模糊逻辑系统。

(credit)来衡量，分数是由相应的分配算法来确定的，该分配算法是实现密歇根方法的关键：遗传算子作用于个体水平，而性能评价则是基于多条规则或整个群体的，这不可避免地会造成规则之间的冲突，这种冲突只能由各规则的得分来调和。那么如何将对群体的评价合理地分配给各个体？目前有一些这样的分配算法，但都不太理想。这是密歇根方法面临的主要问题。

如果将完整的规则集(整个 FLC)作为一个个体，多个不同的 FLC 形成一个群体，遗传算子作用于个体水平，性能评价也在个体水平进行，这就解决了个体之间相互冲突的问题，这种方法就是匹兹堡方法。匹兹堡方法解决了密歇根方法的困难，也带来的新的问题。首先是计算量显著增加，因为需要评价多个 FLC。同时，个体的适应值(性能)同样难以反馈到单条规则上，这就是 Carse[13]所谓的"强化信息的带宽很窄"，从而难以提高单条规则的性能，使得 GA 的学习效率低。

考虑到匹兹堡方法和密歇根方法各自的优缺点，人们提出了一些对两者折衷的方法，如 Furuhashi 等的 Nagoya 方法[4]。在 Nagoya 方法中，个体也由单个的 FLC 构成，但这些 FLC 不包含完整的规则集，只有一些随机产生的规则，而且这些规则的数目也是可变的。Nagoya 方法可以看作是匹兹堡方法的一种改进。

2.1 基于领域的 FLC 和基于规则的 FLC

在[20]中，作者将传统的模糊逻辑系统分为三类：纯模糊逻辑系统、高木-关野(Takagi-Sugeno，简称 TSK)系统和具有模糊化与反模糊化的模糊逻辑系统。模糊控制器由于涉及精确的输入输出量，一般都采用后两种类型。

FLC 又可以分为基于领域的 FLC 和基于规则的 FLC[1]。传统的 FLC 大多是基于领域的，其特点是具有覆盖输入输出空间的全局的模糊划分，对每一输入划分的组合均有一模糊规则相对应，亦即具有完整的规则集。基于规则的 FLC 一般没有完整的规则集，规则的数目是可变的，更重要的是，隶属函数是局部定义的，依赖于相应的规则。基于规则的 FLC 由于具有可变数目的规则，能应用于从简单到复杂的各类控制对象，具有很强的可伸缩性，尤其适合多输入的、复杂的控制任务。与基于领域的 FLC 相比，基于规则的 FLC 还有一个显著的优点：可以不用预先规定隶属函数的个数。基于领域的 FLC 大都需要预先确定模糊划分(隶属函数)的个数，以此来设计规则集。规则集中可能的最大规则数为各输入变量模糊划分数的乘积，若划分太细，规则数量急剧增长，使 FLC 结构变得复杂，增加了学习的负担；若个数太少，又不能满足对 FLC 性能的要求。在基于规则的 FLC 中，规则的定义不需要全局隶属函数，因而规则的数目是可以自由变动的。基于规则的 FLC 的主要缺点是可靠性不能得到充分的保证，因为局部定义的隶属函数不能保证覆盖全部的输入空间，造成在输入空间中的某些区域 FLC 没有相应的规则与之对应。另外，局部定义的模糊集隶属函数也削弱了模糊集的语义特性，使得模糊控制规则可理解性降低。

2.2 GA 设计基于领域的 FLC

用 GA 设计 FLC，从方法看，一般都采用匹兹堡方法或匹兹堡方法的改进，如 Nagoya 方法。从 FLC 的类型看，既有基于领域的 FLC，也有基于规则的 FLC，还有模糊神经网[8]，模糊神经网具有全局的隶属函数，可以看作是一种基于领域的 FLC。

传统的 FLC 一般都是基于领域的，具有思路直观、可靠性强、规则易于理解的优点。由于其规则和隶属函数的个数事先确定，GA 优化时个体的编码长度就是一定的，这合乎传统 GA 的要求，无需对 GA 作大的改动。个体长度的固定也有助于估计搜索空间的规模、把握学习的进程。

用 GA 优化 FLC 时，优化的主要对象是 FLC 的隶属函数和规则集，其它部分如反模糊化方法、模糊蕴涵等很少涉及。在基于领域的 FLC 中，隶属函数是全局定义的，因而与规则相对独立的。GA 在设计这类 FLC 时，有下面几种不同的类型：

1)已知模糊控制规则集，优化隶属函数；

2)已知隶属函数，优化规则集；

3)规则集与隶属函数由 GA 分阶段优化；

4)同时优化规则集与隶属函数。

2.2.1 优化模糊集隶属函数。

当已知规则集时，输入输出变量的模糊划分实际上也大致确定了，GA 主要用于细调隶属函数的位置、形状等参数。最早进行这方面研究的是 Karr[7]。作者以倒摆问题为例，先给出倒摆控制器的模糊规则集，利用规则集对倒摆进行控制，同时用 GA 优化变量的隶属函数参数。四个输入变量均划分为三个三角形的模糊集，分别为负、零、正。四个输入变量的模糊集共有 81 种组合，相应地能构造 81 条模糊控制规则。将各隶属函数的参数排列起来，形成 GA 的个体。结果表明 GA 优化后的隶属函数远远优于手工设计的。

Park 等也在文[16]中使用了相似的优化隶属函数的方法。以一输入一输出的直流电机转速控制为例，作者首先给定输入输出变量的模糊关系矩阵(规则)和隶属函数的个数。隶属函数的形状采用等腰三角形，每一三角形包括两个参数，分别决定三角形的顶点位置和基底长度，实验结果同样证明了GA优化的有效性。

2.2.2 优化模糊控制规则。有时设计者可以事先确定输入输出隶属函数的形状、位置等参数，这时只需用GA搜索优化的规则集。Thrift[18]用GA设计了一个二输入一输出的FLC规则集。控制对象是一个在一维无摩擦轨道上运行的小车，目的是让小车尽快停在轨道上一个指定位置，例如轨道中点。输入变量是小车的位移和速度，输出变量是作用在小车上的力。输入输出变量均划分为五个三角形的模糊集，依次为负中(NM)、负小(NS)、零(Z)、正小(PS)和正中(PM)，这样就构成一个5×5的规则表，表中每一项可从{0,1,2,3,4,5}(分别对应于{NM,NS,Z,PS,PM,-}，"-"代表控制器无动作)中任意取值。将这个二维表按一定顺序展开成一维，就形成了GA的个体。用某一个体的规则控制小车的运行，所得的结果作为该个体的适应值。Thrift所使用的交换算子为经典的两点交换，突变算子定义为选中的突变基因变为其相邻的模糊集或"-"。小车模拟运行的时间为10秒，时间间隔为0.02秒。个体的适应值定义为500—T，其中T为小车在25次不同初始状态的模拟中停在中点的平均用时。群体规模为31，经过100代后，得出的FLC优于优化"bang-bang"控制。

优化输入输出模糊关系矩阵也就是优化控制规则。在Park[16]的直流电机转速控制中，将输入变量划分为六个等腰三角形的模糊集，输出变量划分为五个等腰三角形模糊集，构成一6×5的模糊关系矩阵，将矩阵编码成一维的个体，共有30个基因，每个基因在[0,1]中取值。采用实数编码，突变算子定义为基因值在原有的基础上增减10%。群体规模为50，运行100代后得到了良好的模糊关系矩阵。

2.2.3 分阶段优化控制规则和隶属函数。在设计FLC时，很少能预先确定规则和隶属函数的参数，因此既需要优化规则集，也需要优化隶属函数。Kinzel[9]认为如果同时优化FLC的隶属函数和规则，势必大大增加搜索的复杂性。Kinzel还认为FLC结构的优化集中在GA学习的前期，而后期主要是参数的细调，尤其是隶属函数参数的调整。据此作者提出GA设计FLC的三个步骤：

①根据已有的知识产生一"好"的规则集；

②对输入输出空间进行手工的划分，建立隶属函数，并以此为基础，用GA优化规则集；

③在②的基础上，用GA细调隶属函数参数。

Kinzel认为在FLC的多维(维数等于变量数)规则表中，相邻的项(规则)之间有强烈的相互作用，若把多维的规则表转化为传统的一维个体，就可能破坏规则间的相互关系，给优化带来困难。因此文[9]采用的是多维的个体，保持规则表的原样。Kinzel的方法直观简易、计算量小，很有实用价值，但由于FLC中隶属函数与规则的复杂相互作用，这种分步优化的方法易于陷入局部极点，很难达到全局最优。

2.2.4 同时优化规则和隶属函数。既然分阶段优化的方法不能达到最优，那么自然考虑要同时优化隶属函数和控制规则，如文[8,10,11,13,17]等。这方面开创性的工作见[19]，尽管文中的控制器还不是严格意义上的FLC。Lee和Takagi[10]的同时优化规则和隶属函数是早期有代表性的工作。Lee和Takagi以一个简化的倒摆系统为例，用GA设计了一个二输入一输出TSK型的FLC。个体分为两部分：输入变量的隶属函数参数和控制规则的后件参数。每个输入变量均划分为10个三角形的模糊集，每一模糊集由3个参数表示，共有10×10=100条规则，每一规则有3个后件参数，因此每一个体有2×10×3+10×10×3=360个参数。采用二进制编码，每参数用8个二进制位表示，故个体总长为360×8=2880位，搜索空间是十分巨大的。为了减少规则的数目，文[10]在适应值函数中增加了一个惩罚项，使得个体中活动的规则越多，个体的适应值越小，最后得到只有4条规则的FLC。

另一典型的例子是Li RenHou和Zhang Yi的文[13]，也采用TSK型的FLC，用GA产生出的FLC成功地控制了双倒摆系统。双倒摆系统是拥有六输入一输出的复杂系统，作者将每一输入变量均划分为正、负两个对称的模糊集，因此一共有12个隶属函数、64条规则，每条规则有7个后件参数。为了减少参数的数目，缩小搜索空间，文[13]利用双倒摆系统本身的对称性和线性系统控制理论，将需要优化的参数个数削减到56个。考虑到双倒摆系统的复杂性，这种计算上的节省是惊人的，尤其是这样的压缩不会削弱FLC的性能。

除了TSK型的FLC外，输入输出均模糊化的

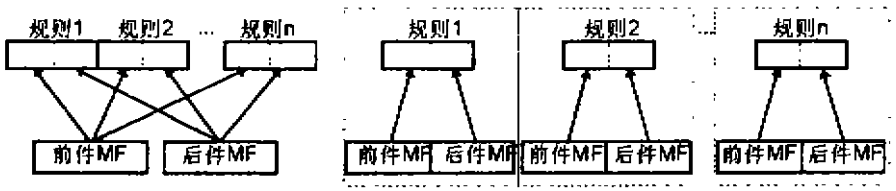
FLC 也常常被采用。在文[5]中，Homaifar 通过固定输入输出变量隶属函数的个数，固定三角形隶属函数顶点的位置，并限制隶属函数基底的变化范围，缩小了隶属函数编码的长度。不过这种压缩，尤其是固定隶属函数顶点的位置，可能损及 FLC 的性能。Homaifar 将 GA 对 FLC 的优化过程分为两个阶段：进化阶段和精化阶段。进化阶段的目的是搜索出基本满足要求的 FLC，确定 FLC 的结构和主要的规则；精化阶段的主要任务是细调 FLC 的参数，尤其是隶属函数的参数，以提高 FLC 的性能。两个阶段的区别主要体现在 GA 适应值函数不同。Homaifar 以汽车的倒车问题为例，在进化阶段，凡使汽车倒到终点的 FLC(个体)均获得相同的正的适应值，比如 2.0；在精化阶段，希望汽车不仅能到达终点，还应该尽可能快，因此将适应值函数修改为 100－T，T 为汽车到达终点所需的时间，这样就使得 GA 倾向于选择倒车更快的个体。遗憾的是，文[5]中没有做有无精化阶段的对照实验，也没有说明精化过程对隶属函数和规则的影响。

一般认为，对 FLC 的优化大体上可分为两个阶段，前期主要对 FLC 结构进行调整，如隶属函数的个数及其大致分布、主要的控制规则；而后期则主要对 FLC 的参数进行优化，特别是隶属函数的有关参数。后期基本上可以看作数值优化。经典的 GA 对这样的问题，特别是在最优点附近，效率是不高的，最好是能结合局部搜索的方法。在文[12]中，Liska 也采用了两阶段寻优的技术：首先用实编码的 GA 进行搜索，确定优化的 FLC 的结构和各参数大致的取值范围，再使用共轭梯度下降法利用训练数据对隶属函数参数进行优化。Liska 的方法取得了良好的效果。美中不足的是共轭梯度下降法需要训练数据对，而良好的训练对并不总是能得到的。

2.3 优化基于规则的 FLC

FLC 实质上是一种插值器，插值基函数就是输入隶属函数，隶属函数的个数比形状对 FLC 的性能影响更大[11]。在上面的讨论中，我们注意到对于基于领域的 FLC，无论采用哪一种优化方法，输入隶属函数的个数总需要事先确认，据此才能构造出规则。这极大地限制了 GA 对 FLC 结构的优化，使得 GA 对 FLC 的优化主要集中在对 FLC 的参数的数值优化方面。

为了更好地利用 GA 全局优化的特点，加强对 FLC 结构的学习，历史上曾提出过多种改进的方法，如 Lee[10]对规则个数的惩罚方法，Kim[4]的“多层次搜索(Multiresolutional Search)”方法等，都取得了一定效果。Lee 和 Takagi 从一个较复杂的 FLC 开始，例如每个输入变量有 10 个隶属函数，在进化过程中，使选择的压力偏向于结构较简单(规则较少)的 FLC。这是一种有效的方法，不过计算量大，同时也难以维持适当的选择压力。Kim 的方法恰好相反，从一个较简单的 FLC(例如只有 3 个隶属函数)开始，若 GA 没有找到结构合适的 FLC，则增加一个隶属函数。该方法的核心是所谓的分层分布式遗传算法(HDGA)，在有希望的搜索方向上分配更多的计算资源，希望节省计算。Kim 的方法也有一定的效果，不过同样面对着计算量大的问题。Kim 相当于在传统优化方法上套了一层，用以优化 FLC 的结构，但这不是一种根本的解决方法。



[a] 基于领域的FLC　　[b] 基于规则的FLC

图 1　两种 FLC 结构示意图(MF：隶属函数)

基于领域的 FLC 的性能最终取决于输入隶属函数的个数，一旦输入隶属函数的数目确定以后，FLC 可能的最大规则数也随之确定。换句话说，隶属函数个数的决定性作用表现在对可能的最大规则数的限制上。有时由于规则条数偏少，一味调整规则前后件参数是无济于事的；若规则数目太多，又会加重学习的负担。那么能否在 GA 进化的过程中自主地改变规则的个数呢？在基于领域的 FLC 中，规则集的规模是由输入变量隶属函数的个数决定的，而隶属函数的个数必须预先确定，否则就不能构造出规则。为了使得规则数能在 GA 进化过程中自主地改变，必须抛弃基于领域的 FLC 和全局定义的隶属函数，采用基于规则的 FLC 和局部定义的隶属函数。

基于领域的FLC与基于规则的FLC结构示意图如图1。由图1可以看出，在基于规则的FLC中，隶属函数不再与规则相分离，而是依附于相应的规则，不与其他规则中的隶属函数发生直接的关系。这样，当增删一条规则时，意味着增删相应的隶属函数。

用GA优化基于规则的FLC有一些成功的例子。Cooper和Vidal在文[2]中提出"紧缩编码方法(compact encoding scheme)"，利用GA优化出一个基于规则的FLC。GA的个体由数目不定的规则构成，每条规则又由输入和输出模糊集隶属函数组成。群体规模为200时，经过30代，产生出一个只有三条规则的FLC，成功地实现了复杂的双倒摆系统的控制。基于规则的FLC最大优点是对问题的良好的伸缩能力，规则数不再随着输入空间的细化而指数增长(这些规则中有许多是非本质的、冗余的)。进化过程能保证得到最关键的、最能反映模糊关系特征的控制规则。另外，它还允许规则不包含所有的输入变量，例如在上述双倒摆系统中，允许存在只有五个或更少输入变量的规则。这种不完整条件规则有利于降低系统的维数，获得对系统的更深入的认识。一条不完整的规则相当于几条完整规则的总和，是可以理解的，也符合人类思维的特点。

由于规则集规模可变，相应的个体长度也发生变化，需要对经典的遗传算子作一些改进，并增加一些新的遗传算子，如规则的增删等。由于个体长度是可变的，因而规则(基因)一般与位置无关，因此经典的GA交换操作可能造成所谓的"结构/功能"问题，即相同或相近功能的基因可能通过交换集中在一个个体上，这一个体就可能会丢失负责其余功能的基因。Cooper的解决方法是在交换前依照相关隶属函数中心点的位置对规则进行排序，隶属函数相邻的规则在个体中也相邻，这样使得交换尽量在功能相似的基因间进行；另外尽量减少交换的发生，代之以其他的算子。

从图1中可以看到，由于每条规则都需要定义自己的隶属函数，需要优化的隶属函数参数的个数显著增加，使得匹兹堡方法本来就有的强化信息带宽狭窄的矛盾愈益突出，单条规则性能的改善更加困难。为此Furuhashi等对匹兹堡方法进行了改进，提出Nagoya方法。Nagoya方法将个体分为若干部分，每一部分都包含数目大致相等的规则，将突变算子施加于第一部分若干次，分别计算出各次突变后个体的适应值，选出其中最好的一次，替换原个体中相应的部分。对个体的其余部分也做类似的处理，直至各部分都得到更新。这时产生了一个新的性能有相当提高的个体，对群体中的各个体都执行上述步骤后，再进行选择、交换、繁殖等经典的操作。Nagoya方法与Cooper的方法很相似，也采用基于规则的FLC、可变长的编码，区别在于前者改进了突变算子，将类似于贪心法的局部搜索的方法结合进GA进化的过程，提高了学习效率。

基于规则的FLC最大的优点是伸缩性很好，大大减轻了对专家知识的依赖，尤其适合于复杂的、人们了解较少的控制对象。其缺点也突出，首先是适当的交换算子难以定义，在交换后容易出现"结构/功能"问题，因此寻找适合的交换算子、抑制交换带来的不利影响是用GA设计基于规则的FLC面临的主要问题。其次，局部定义的隶属函数不能保证覆盖整个输入空间，会出现在某些情况下FLC不能对输入作出反应，这在对可靠性要求很高的场合是不能允许的。最后，局部定义的隶属函数在很大程度上丧失了语义，所得到的规则可理解性较差。

三、需要注意的几个问题

FLC的优化可以看作是参数优化的问题，GA在其他数值优化领域所使用的行之有效的方法，大都可以应用于优化FLC中，如精英(elitist)策略、两点或多点交换、可变概率算子等。FLC也有自己的特点，在用GA优化FLC时，需要处理好以下几个方面的问题。

3.1 有效地嵌入专家的知识

为了提高学习效率，专家的知识和经验应该贯穿于学习过程的始终。利用知识主要体现在两方面：首先在初始化群体时，群体中应包含一个或多个由专家根据经验手工设计的、在一定程度上体现控制对象特点的、合符常规的个体(FLC)，如[9]中称的"好"的个体；群体中其余个体可以随机生成，并在进化过程中采用精英策略，保证群体能在一定基础上搜索，提高GA在进化初期的效率；其次在进化过程中应根据经验对参数的取值范围进行限制，例如模糊集"正大"的最高点应该位于模糊集"正中"的最高点的右面，"正中"与"正大"之间交叉区域不应超过各自的最高点位置，这些限制维护了FLC的语义特征，大大提高了GA的效率，又不致对FLC的性能造成明显的损害。GA是概率搜索的算法，算子是随机算子，进化过程中可能出现"坏"的，违背常识的规则，如有可能，应尽早予以剔除。在设计基于领域的

FLC时，还要注意正确选择输入输出变量隶属函数的个数。

3.2 选择模糊集隶属函数的形状

隶属函数的形状对FLC的性能影响很大，是GA优化的重要内容。广泛使用的隶属函数有三角形、高斯函数、梯形、径向基函数(radial basis function)等。选择隶属函数，除了根据控制对象本身的特点外，还要考虑有利于反模糊化，有利于减少参数个数。高斯函数和等腰三角形只有两个参数，一般三角形三个参数，梯形有四个参数。如果可能，要尽量采用参数较少的形状。此外，许多问题是关于平衡点(set point)左右(上下、正负)对称的，此时应该取对称的隶属函数和规则，尽量减少参数个数。如文[13]。

3.3 选择适合的适应值函数

选择适合的适应值函数被认为是GA中最关键、最具主观性的环节。适应值函数必须反映设计者对FLC各方面特性的要求。这些要求大体上可归为两类：FLC的性能和FLC结构的复杂度。性能包括多方面，如平衡点附近的精度、到达平衡点的时间、超调等。FLC结构复杂度包括隶属函数的个数和规则条数。性能和复杂度要求之间往往是矛盾的，各性能指标之间往往也会出现矛盾。这些互相矛盾的要求通常以加权和的形式构成适应值函数，选择适当的权重是设计适应值函数的关键。采用分阶段的适应值函数是解决这一困难的有效方法[5]。

3.4 适合的编码方式

将FLC编码成个体是用GA优化FLC的中心环节，涉及许多问题。例如采用二进制编码还是实数编码？经典的GA采用二进制编码，但有越来越多的研究表明实数编码在数值优化方面有更高的精度和效率[15]。二进制编码的优越性表现在可以选择参数的二值化位数，以便控制搜索空间的大小。这也正是二进制编码的缺点，设计者往往不能事先正确选择适当的二值化位数。另外的问题包括编码的长度是否可变、是否与位置相关等，都没有明确的结论，需要设计者根据实际情况进行取舍。寻找FLC的新的编码方法，如文[14]，是GA优化FLC研究的中心内容。

3.5 结合局部优化的方法

FLC的优化体现在两个层次上，即结构的优化和参数数值的优化。经验表明，作为全局优化器的GA在数值优化中，接近最优点时效率并不高。解决这个问题的思路有多种，如[5]的分段适应值函数，相当于改变适应值的缩放因子，增强对精度较高的FLC的选择。更直接的方法是在GA进化过程中结合局部优化，如Nagoya方法、Liska[12]的共轭梯度下降法、Ishigami[6]的模糊神经网误差反传法等，都能有效地调整隶属函数参数。共轭梯度下降法和误差反传法都需要训练数据对，但良好训练数据对并不总能得到。

四、总结

用GA实现FLC具有许多其他方法所没有的优越性：GA可以优化FLC从结构到参数数值的各个方面；可以不需要设计者对控制对象有任何先验的知识，可以没有良好的训练数据对；易于融入设计者的知识以提高学习效率；易于同其他方法结合使用。作为总结，以下是GA优化FLC时可以参考的一些原则：

1)选择适当类型的FLC。采用何种类型的FLC，要视控制对象的复杂程度和对问题的知识经验多寡而定。若FLC的结构和规模大致能够确定，采用基于领域的FLC较好，同时在可靠性要求很高的场合，或要求规则有明确语义的场合，也要选用基于领域的FLC；

2)尽量缩短编码长度，减少待优化的参数个数；进化过程中，限制各参数的取值范围；

3)分阶段优化规则和隶属函数，能大大节省计算量，能获得比较满意的结果，很有实用价值；

4)充分利用已有的知识和经验；

5)适应值函数的确定是一个过程，经过多次实践才能最后确定；

6)结合局部优化的方法细调隶属函数参数。

FLC的优化是参数优化的问题，因此从理论上讲，除GA外，其他的进化优化方法，如进化策略、进化规划等也同样可用来优化FLC。模糊控制规则是产生式规则，GA的一个分支遗传编程(Genetic Programming，简称GP)正是直接生成产生式规则的，Edmonds[8]等用GP产生了一个由模糊逻辑产生式规则构成的系统，应用于金融商业领域。用GP优化FLC也许是一个有意义的研究方向。

参考文献

[1] B. Carse. Fuzzy Sets and Systems, 80(3), 1996, 273-293

[2] M. G. Cooper. IEEE-FUZZ' 94, 1332-1337

[3] A. N. Edmonds. IEEE-FUZZ' 95, 765-770

[4] T. Furuhashi. LNAI 1011, Springer-Verlag, 1995, 173-189