第 33 卷第 10 期
2016 年 10 月

控 制 理 论 与 应 用
Control Theory & Applications

Vol. 33 No. 10
Oct. 2016

DOI: 10.7641/CTA.2016.60098

基于加权Karnik-Mendel算法的区间二型模糊逻辑系统降型

陈 阳[1,2†], 王大志[1]

(1. 东北大学 电力系统与电力传动研究所, 辽宁 沈阳 110819; 2. 辽宁工业大学 理学院, 辽宁 锦州 121001)

摘要: 二型模糊逻辑系统是当前的学术研究的热点问题, 而降型是该系统中非常重要的一个模块. Karnik-Mendel (KM)算法是被用来计算和完成区间二型模糊逻辑系统降型的标准算法. 通过比较离散版本KM算法中求和运算和连续版本的KM(continuous version of KM, CKM)算法中求积分运算, 本文利用数值积分技术中牛顿- 柯斯特求积公式将标准KM算法扩展成3种不同形式的加权KM(weighted KM, WKM)算法. 而KM算法只是WKM算法中的一种特殊情况. 3个计算机仿真例子用来阐述和分析WKM算法的表现, 与传统的KM算法相比, WKM算法有较小的绝对误差和较快的收敛速度, 给二型模糊逻辑系统设计者和应用者提供了潜在的应用价值.

关键词: 区间二型模糊逻辑系统; 降型; Karnik-Mendel算法; 积分; 加权Karnik-Mendel算法; 计算机仿真

中图分类号: TP182; TP11　　　**文献标识码**: A

Type-reduction of interval type–2 fuzzy logic systems with weighted Karnik-Mendel algorithms

CHEN Yang[1,2†], WANG Da-zhi[1]

(1. Institute of Electric Power System and Motor Drives, Northeast University, Shenyang Liaoning 110819, China;
2. College of Science, Liaoning University of Technology, Jinzhou Liaoning 121001, China)

Abstract: Studies on type–2 fuzzy logic systems is a hot topic in the current academic area. While type-reduction is one of the most important blocks in the systems. KM algorithms are standarded algorithms which are used to compute and perform the type-reduction of interval type–2 fuzzy logic systems. By comparing the sum operation in discretized version KM algorithms and the integral operation in continuous version of KM (CKM) algorithms, the paper extends the standarded KM algorithms to three different forms of weighted KM (WKM) algorithms according to the Newton-Cotes quadrature formulas of numerical integration techniques. And the KM algorithms become a special case of the WKM algorithms. Three computer simulation examples are used to illustrate and analyze the performance of the WKM algorithms. Compared with the traditional KM algorithms, the WKM algorithms have smaller absolute error and faster convergence speed, which provide the potential application value for designers and adopters of type–2 fuzzy logic systems.

Key words: interval type–2 fuzzy logic systems; type-reduction; Karnik-Mendel algorithms; integration; weighted Karnik-Mendel algorithms; computer simulation

1 引言(Introduction)

在生产生活中, 诸如金融系统、自主移动机器人、智能控制器、设备监测和诊断等过程都具有高不确定性、非线性和时变等特征. 与传统的一型模糊集相比, 区间二型模糊集可以更好地建模并通过调整参数以减小不确定性所带来的不良影响. 因此, 基于区间二型模糊集的区间二型模糊逻辑系统近年来成为一种热门的新兴技术.

区间二型模糊逻辑系统由5个模块组成, 包括模糊器、规则库、推理机、降型器和解模糊器. 其中降型模块在系统中起着至关重要的作用, 它的主要功能是将区间二型模糊集转化成一型模糊集, 而在一型模糊逻辑系统中没有降型模块. 由于涉及复杂的降型过程, 区间二型模糊逻辑系统中的运算比一型模糊逻辑系统要复杂得多. 此外, 区间二型模糊逻辑系统的前件或后件中至少存在一个区间二型模糊集, 而一型模糊逻辑系统只使用一型模糊集.

由于区间二型模糊集[1]的次隶属度恒等于1, 计算相对简单, 基于区间二型模糊集的区间二型模糊逻辑系统[2–4]被广泛应用于各个领域. KM(Karnik-Mendel)

收稿日期: 2016−02−24; 录用日期: 2016−08−05.
†通信作者. E-mail: chenyanghanyun@163.com.
本文责任编委: 陈增强.
国家自然科学基金项目(61374113), 辽宁省高校基本科研业务费项目(JL201615410)资助.
Supported by National Natural Science Foundation of China (61374113) and Fundamental Research Funds for Liaoning's Universities (JL2016 15410).

算法[5]被开发用来完成区间二型模糊逻辑系统的降型或计算区间二型模糊集的质心[6], 其在二型模糊逻辑系统的研究中起着很重要的作用[7–8]. Mendel和Wu[9]提出了连续版本的KM (continuous version of KM, CKM)算法. Mendel和Liu[10]对CKM算法进行了理论分析且证明了该算法的单调性和超指数收敛性. Liu等人[11]对EKM算法的初始化过程给出了理论解释, 利用数值积分技术将EKM算法扩展成加权EKM (WEKM)算法并计算区间二型模糊集质心的左端点. 以上工作为KM算法的应用奠定了丰富的理论基础.

受文献[5–6, 8, 11]启发, 本文针对计算完成区间二型模糊逻辑系统的质心降型过程, 提出连续版本的KM(CKM)算法. 通过比较离散版本的KM算法的求和运算和连续版本的KM算法的求积分运算, 使用数值积分技术将KM算法扩展成加权KM(WKM)算法. 新的WKM算法计算完成区间二型模糊逻辑系统的质心降型与KM算法相比更加准确, 且KM算法只是WKM算法的一种特殊情况.

本文余下的部分组织如下: 第1节提出关于区间二型模糊逻辑系统[8, 19]、区间二型模糊集的质心和KM算法等背景知识. 第2节介绍了牛顿–柯斯特(Newton-Cotes)求积公式、CKM算法和基于3种不同加权分配方法下的WKM算法计算区间二型模糊集的质心. 第3节利用3个数值仿真例子比较和分析4种算法的表现. 第4节给出结论与展望.

2 背景知识(Background knowledge)

2.1 区间二型模糊逻辑系统(Interval type–2 fuzzy logic systems)

常用的区间二型模糊逻辑系统有Mamdani和TSK两种结构. 一个Mamdani区间二型模糊逻辑系统有p个输入$x_1 \in X_1, \cdots, x_p \in X_p$和一个输出$y \in Y$且被$M$条模糊规则所描述, 其中第$l$条规则形式为: 如果$x_1$是$\tilde{F}_1^l$, 且$\cdots$, 且$x_p$是$\tilde{F}_p^l$, 则$y$是$\tilde{G}^l$ ($l = 1, \cdots, M$).

为简化表达, 假设采用单点模糊化, 即输入测量被建模成普通集合. 对每条模糊规则, 首先计算激发区间$F^l(x')$, 当$x = x'$时, 计算F^l:

$$\begin{cases} F^l(x') \equiv [\underline{f}^l(x'), \bar{f}^l(x')], \\ \underline{f}^l(x') \equiv T_{i=1}^p \underline{\mu}_{\tilde{F}_i^l}({x'}_i), \\ \bar{f}^l(x') \equiv T_{i=1}^p \bar{\mu}_{\tilde{F}_i^l}({x'}_i), \end{cases} \tag{1}$$

其中T表示取小或乘积t–范运算.

一个区间二型模糊集所有不确定性构成一个带状区域, 称之为足迹不确定性(FOU).

当采用质心降型时, 每条规则的激发区间与其后件区间二型模糊集结合以产生激发输出集$\tilde{B}^l$(由它的FOU描述).

$$\tilde{B}^l: \begin{cases} \mathrm{FOU}(\tilde{B}^l) = [\underline{\mu}_{\tilde{B}^l}(y|x'), \bar{\mu}_{\tilde{B}^l}(y|x')], \\ \underline{\mu}_{\tilde{B}^l}(y|x') = \underline{f}^l(x') * \underline{\mu}_{\tilde{G}^l}(y), \\ \bar{\mu}_{\tilde{B}^l}(y|x') = \bar{f}^l(x') * \bar{\mu}_{\tilde{G}^l}(y), \end{cases} \tag{2}$$

其中$*$也表示取小或乘积t–范运算.

合并所有的激发输出集$\tilde{B}^l$以完成区间二型模糊集$\tilde{B}$的聚合运算.

$$\tilde{B}: \begin{cases} \mathrm{FOU}(\tilde{B}) = [\underline{\mu}_{\tilde{B}}(y|x'), \bar{\mu}_{\tilde{B}}(y|x')], \\ \underline{\mu}_{\tilde{B}}(y|x') = \underline{\mu}_{\tilde{B}^1}(y|x') \vee \cdots \vee \underline{\mu}_{\tilde{B}^M}(y|x'), \\ \bar{\mu}_{\tilde{B}}(y|x') = \bar{\mu}_{\tilde{B}^1}(y|x') \vee \cdots \vee \bar{\mu}_{\tilde{B}^M}(y|x'), \end{cases} \tag{3}$$

其中$\vee$表示取大运算. 最终, 通过降型计算$\tilde{B}$质心$C_{\tilde{B}}$得出降型集$Y_C(x')$, 即

$$Y_C(x') = C_{\tilde{B}}(x') = 1/[l_{\tilde{B}}(x'), r_{\tilde{B}}(x')], \tag{4}$$

其中$l_{\tilde{B}}(x')$和$r_{\tilde{B}}(x')$可由KM算法计算出.

2.2 区间二型模糊集的质心(Centroid of an interval type–2 fuzzy set)

一个区间二型模糊集$\tilde{A}$的质心$C_{\tilde{A}}(x)$是其所有n_{A}个嵌入式一型模糊集A_{e}的质心$C_{\tilde{A}}(A_{\mathrm{e}})$的并, 即[6]

$$C_{\tilde{A}}(x) = 1/\bigcup_{\forall A_{\mathrm{e}}} c_{\tilde{A}}(A_{\mathrm{e}}) = 1/\bigcup_{\forall A_{\mathrm{e}}} \frac{\sum_{i=1}^{N} x_i \mu_{A_{\mathrm{e}}}(x_i)}{\sum_{i=1}^{N} \mu_{A_{\mathrm{e}}}(x_i)} = 1/[c_{\mathrm{l}}(\tilde{A}), c_{\mathrm{r}}(\tilde{A})], \tag{5}$$

其中:

$$c_{\mathrm{l}}(\tilde{A}) = \min_{\forall A_{\mathrm{e}}} c_{\tilde{A}}(A_{\mathrm{e}}) = \min_{\forall \theta_i \in [\underline{\mu}_{\tilde{A}}(x_i), \bar{\mu}_{\tilde{A}}(x_i)]} \Big(\sum_{i=1}^{N} x_i \theta_i / \sum_{i=1}^{N} \theta_i\Big), \tag{6}$$

$$c_{\mathrm{r}}(\tilde{A}) = \max_{\forall A_{\mathrm{e}}} c_{\tilde{A}}(A_{\mathrm{e}}) = \max_{\forall \theta_i \in [\underline{\mu}_{\tilde{A}}(x_i), \bar{\mu}_{\tilde{A}}(x_i)]} \Big(\sum_{i=1}^{N} x_i \theta_i / \sum_{i=1}^{N} \theta_i\Big), \tag{7}$$

$x_i (x_1 < x_2 < \cdots < x_N)$为主变量的采样值.

2.3 KM算法(KM algorithms)

KM算法是一种用来估计两个端点c_{l}和c_{r}值的优化算法, 即

$$c_{\mathrm{l}} = \min c_{\mathrm{l}}(k) = \frac{\sum_{i=1}^{k} x_i \bar{\mu}_{\tilde{A}}(x_i) + \sum_{i=k+1}^{N} x_i \underline{\mu}_{\tilde{A}}(x_i)}{\sum_{i=1}^{k} \bar{\mu}_{\tilde{A}}(x_i) + \sum_{i=k+1}^{N} \underline{\mu}_{\tilde{A}}(x_i)}, \tag{8}$$

$$c_{\mathrm{r}} = \max c_{\mathrm{r}}(k) = \frac{\sum_{i=1}^{k} x_i \underline{\mu}_{\tilde{A}}(x_i) + \sum_{i=k+1}^{N} x_i \bar{\mu}_{\tilde{A}}(x_i)}{\sum_{i=1}^{k} \underline{\mu}_{\tilde{A}}(x_i) + \sum_{i=k+1}^{N} \bar{\mu}_{\tilde{A}}(x_i)}, \tag{9}$$

其中L和R是由KM算法计算的在上级和下级隶属函数之间切换的转折点.

表1给出离散版本的KM算法计算一个区间二型集质心的具体过程.

表 1 KM算法计算一个区间二型模糊集的质心[5]

Table 1 Compute the centroid of an interval type–2 fuzzy set by KM algorithms [5]

步骤	KM算法计算c_l, $c_l = \min\limits_{\forall\theta_i\in[\underline{\mu}_{\tilde{A}}(x_i),\bar{\mu}_{\tilde{A}}(x_i)]}(\sum\limits_{i=1}^N x_i\theta_i/\sum\limits_{i=1}^N\theta_i)$
1	初始化θ_i, 设置$\theta_i = [\underline{\mu}_{\tilde{A}}(x_i)+\bar{\mu}_{\tilde{A}}(x_i)]/2,\ i=1,\cdots,N$, 计算$c' = c(\theta_1,\cdots,\theta_N) = \sum\limits_{i=1}^N x_i\theta_i/\sum\limits_{i=1}^N\theta_i$
2	找到$k(1\leqslant k\leqslant N-1)$满足$x_k\leqslant c'\leqslant x_{k+1}$
3	当$i\leqslant k$, 设置$\theta_i=\bar{\mu}_{\tilde{A}}(x_i)$; 当$i\geqslant k+1$, 设置$\theta_i=\underline{\mu}_{\tilde{A}}(x_i)$, 计算$c_l(k)=\dfrac{\sum\limits_{i=1}^k x_i\bar{\mu}_{\tilde{A}}(x_i)+\sum\limits_{i=k+1}^N x_i\underline{\mu}_{\tilde{A}}(x_i)}{\sum\limits_{i=1}^k\bar{\mu}_{\tilde{A}}(x_i)+\sum\limits_{i=k+1}^N\underline{\mu}_{\tilde{A}}(x_i)}$
4	核对是否$c_l(k)=c'$, 若成立, 终止且设置$c_l(k)=c_l$, $k=L$; 若不成立, 转入第5步
5	设置$c'=c_l(k)$且返回第2步
步骤	KM算法计算c_r, $c_r = \max\limits_{\forall\theta_i\in[\underline{\mu}_{\tilde{A}}(x_i),\bar{\mu}_{\tilde{A}}(x_i)]}(\sum\limits_{i=1}^N x_i\theta_i/\sum\limits_{i=1}^N\theta_i)$
1	初始化θ_i, 设置$\theta_i = [\underline{\mu}_{\tilde{A}}(x_i)+\bar{\mu}_{\tilde{A}}(x_i)]/2,\ i=1,\cdots,N$, 计算$c' = c(\theta_1,\cdots,\theta_N) = \sum\limits_{i=1}^N x_i\theta_i/\sum\limits_{i=1}^N\theta_i$
2	找到$k(1\leqslant k\leqslant N-1)$满足$x_k\leqslant c'\leqslant x_{k+1}$
3	当$i\leqslant k$, 设置$\theta_i=\underline{\mu}_{\tilde{A}}(x_i)$; 当$i\geqslant k+1$, 设置$\theta_i=\bar{\mu}_{\tilde{A}}(x_i)$, 计算$c_r(k)=\dfrac{\sum\limits_{i=1}^k x_i\underline{\mu}_{\tilde{A}}(x_i)+\sum\limits_{i=k+1}^N x_i\bar{\mu}_{\tilde{A}}(x_i)}{\sum\limits_{i=1}^k\underline{\mu}_{\tilde{A}}(x_i)+\sum\limits_{i=k+1}^N\bar{\mu}_{\tilde{A}}(x_i)}$
4	核对是否$c_r(k)=c'$, 若成立, 终止且设置$c_r(k)=c_r$, $k=R$; 若不成立, 转入第5步
5	设置$c'=c_r(k)$且返回第2步

3 WKM算法(WKM algorithms)

在提出WKM算法之前, 首先给出两部分预备知识: Newton-Cotes求积公式和CKM算法.

3.1 牛顿–柯特斯求积公式 (Newton-Cotes quadrature formulas)

数值积分的特点是以一些离散节点上的函数值$f(x_i)$的线性组合近似估计定积分$\int_a^b f(x)\mathrm{d}x$, 从而将定积分的计算归结为函数值的计算.

定义 1(求积公式)　假设$a=x_0<x_1<\cdots<x_n=b$, 定积分

$$\int_a^b f(x)\mathrm{d}x = Q(f)+E(f), \tag{10}$$

那么有公式如下:

$$Q(f)=\sum_{l=0}^n w_l f(x_l) = w_0f(x_0)+w_1f(x_1)+\cdots+w_nf(x_n) \tag{11}$$

被称为求积公式或数值积分. 其中$\{w_l\}_{l=0}^n$称为权重系数, $\{x_l\}_{l=0}^n$称为求积节点, $E(f)$表示截断误差, 称为求积公式的余项.

接下来, 以下的复合梯形法则––复合辛普森(Simpson)法则和复合辛普森(Simpson)3/8法则分别使用直线, 二次多项式函数和三次多项式函数来估计$f(x)$.

定理 1(复合梯形法则)　考虑区间$[a,b]$上的函数$y=f(x)$. 将$[a,b]$划分成宽度为$h=(b-a)/n$的n个子区间$\{x_{l-1},x_l\}_{l=1}^n$, 其中等间隔节点为$x_l=x_0+lh\ (l=0,1,2,\cdots,n)$. 具有复合梯形规则的定积分的数值估计为

$$\int_a^b f(x)\mathrm{d}x = \frac{h}{2}[f(a)+f(b)+2\sum_{l=1}^{n-1}f(x_l)]+E_{\mathrm{T}}(f,h). \tag{12}$$

若函数f在$[a,b]$上是二阶连续可导的, 则误差项

$$E_{\mathrm{T}}(f,h)=-\frac{(b-a)f''(\zeta)}{12}h^2,$$

其中$a<\zeta<b$.

定理 2(复合梯形法则)　考虑区间$[a,b]$上的函数$y=f(x)$. 将$[a,b]$划分成宽度为$h=(b-a)/2n$的$2n$个子区间$\{x_{l-1},x_l\}_{l=1}^{2n}$, 其中等间隔节点为$x_l=x_0+lh(l=0,1,2,\cdots,2n)$. 具有复合辛普森规则的定积分的数值估计为

$$\int_a^b f(x)\mathrm{d}x = \frac{h}{3}[f(a)+f(b)+2\sum_{l=1}^{n-1}f(x_{2l})+4\sum_{l=0}^{n-1}f(x_{2l+1})]+E_{\mathrm{S}}(f,h). \tag{13}$$

若函数f在$[a,b]$上是四阶连续可导的, 则误差项

$$E_{\mathrm{S}}(f,h)=-\frac{(b-a)f^{(4)}(\zeta)}{180}h^4,$$

其中$a<\zeta<b$.

定理 3(复合梯形3/8法则)　考虑区间$[a,b]$上的函数$y=f(x)$. 将$[a,b]$划分成宽度为$h=(b-a)/3n$的$3n$个子区间$\{x_{l-1},x_l\}_{l=1}^{3n}$, 其中等间隔节点为$x_l=x_0+lh\ (l=0,1,2,\cdots,3n)$. 具有复合辛普森3/8规则的定积分的数值估计为

$$\int_a^b f(x)\mathrm{d}x = \frac{3h}{8}[f(a)+f(b)+\sum_{l=1}^n 2f(x_{3l})+\sum_{l=1}^n 3f(x_{3l-2})+\sum_{l=1}^n 3f(x_{3l-1})]+E_{\mathrm{SC}}(f,h). \tag{14}$$

若函数f在$[a,b]$上是四阶连续可导的, 则误差项

$$E_{\mathrm{SC}}(f,h)=-\frac{(b-a)f^{(4)}(\zeta)}{80}h^4,$$

其中$a<\zeta<b$.

本文假设积分式(12)–(14)是可测的, 即积分是有勒贝格(Lebesgue)意义的.

3.2 CKM算法(CKM algorithms)

本节中CKM算法[6,9–10] 被提出用来研究区间二型模糊集质心计算的理论性质.

假设$a=x_1<x_2<\cdots<x_N=b$, 其中a,b分别为主变量x采样值的左右端点值, 那么, 连续版本下的式(8)–(9)为

$$c_{\mathrm{l}}=\min_{\zeta\in[a,b]}f_{\mathrm{l}}(\zeta)=\min_{\zeta\in[a,b]}\frac{\int_a^\zeta x\bar{\mu}_{\tilde{A}}(x)\mathrm{d}x+\int_\zeta^b x\underline{\mu}_{\tilde{A}}(x)\mathrm{d}x}{\int_a^\zeta \bar{\mu}_{\tilde{A}}(x)\mathrm{d}x+\int_\zeta^b \underline{\mu}_{\tilde{A}}(x)\mathrm{d}x},\tag{15}$$

$$c_{\mathrm{r}}=\max_{\zeta\in[a,b]}f_{\mathrm{r}}(\zeta)=\max_{\zeta\in[a,b]}\frac{\int_a^\zeta x\underline{\mu}_{\tilde{A}}(x)\mathrm{d}x+\int_\zeta^b x\bar{\mu}_{\tilde{A}}(x)\mathrm{d}x}{\int_a^\zeta \underline{\mu}_{\tilde{A}}(x)\mathrm{d}x+\int_\zeta^b \bar{\mu}_{\tilde{A}}(x)\mathrm{d}x}.\tag{16}$$

3.3 WKM算法(WKM algorithms)

CKM算法在表2中给出(与文献[6, 9–11]的形式有所不同), 利用它们可以获得对KM算法更好的理论上的理解. 以第2.1节和第2.2节两节作为基础, 本节将给出一类新的KM算法, 称为加权KM算法–WKM算法, 它们的结果可与KM算法相比较.

WKM算法是CKM算法的数值实现. 从表1和表2的比较可得, 连续版本的KM算法与离散版本的KM算法类似, 但将离散版本中的求和运算都转换成了连续版本下的求定积分运算, 即KM算法中在采样点x_i的求和运算起到了相关函数的积分的作用.

利用式(11), 为每个采样值x_i的隶属函数分配相应的权重w_i, 可计算出更准确的c_{l}和c_{r}值, 表3给出了WKM算法计算一个区间二型模糊集质心的具体过程.

KM算法只是WKM算法的一个特例, 其权重为$w_i=1(i=1,\cdots,N)$. 在表3中, 由求积式(11)[12], 可赋予权重多种分配方法. 本文只考虑第2.1节所描述的的数值积分方法: 复合梯形法则、复合Simpson法则和复合Simpson3/8法则, 其求积式分别为(12)–(14). 相关的WKM算法分别称为TWKM, SWKM和S3/8WKM算法. 表4中给出了KM, TWKM, SWKM, S3/8WKM算法的权重赋值方法. 对所有3种WKM算法, $[a,b]$上的采样点都是等间距分布的, 即

$$x_i=a+\frac{i-1}{N-1}(b-a),\ i=1,\cdots,N.$$

表 2 CKM算法计算一个区间二型模糊集的质心

Table 2 Compute the centroid of an interval type–2 fuzzy set by CKM algorithms

步骤	CKM算法计算c_{l}, $c_{\mathrm{l}}=\min_{\forall\theta(x)\in[\underline{\mu}_{\tilde{A}}(x),\bar{\mu}_{\tilde{A}}(x)]}\left(\frac{\int_a^b x\theta(x)\mathrm{d}x}{\int_a^b\theta(x)\mathrm{d}x}\right)$.
1	初始化$\theta(x)=[\underline{\mu}_{\tilde{A}}(x)+\bar{\mu}_{\tilde{A}}(x)]/2$, 计算初始值$\zeta$, 使$\zeta=\frac{\int_a^b x\theta(x)\mathrm{d}x}{\int_a^b\theta(x)\mathrm{d}x}$,
2	当$x\leqslant\zeta$, 设置$\theta(x)=\bar{\mu}_{\tilde{A}}(x)$; 当$x\geqslant\zeta$, 设置$\theta(x)=\underline{\mu}_{\tilde{A}}(x)$, 计算$\zeta_{\mathrm{l}}=\frac{\int_a^\zeta x\bar{\mu}_{\tilde{A}}(x)\mathrm{d}x+\int_\zeta^b x\underline{\mu}_{\tilde{A}}(x)\mathrm{d}x}{\int_a^\zeta\bar{\mu}_{\tilde{A}}(x)\mathrm{d}x+\int_\zeta^b\underline{\mu}_{\tilde{A}}(x)\mathrm{d}x}$.
3	核对是否$\lvert\zeta-\zeta_{\mathrm{l}}\rvert<\varepsilon$ (ε为给定的算法误差边界), 若成立, 终止且设置$c_{\mathrm{l}}=\zeta_{\mathrm{l}}$, 若不成立, 转入第4步.
4	设置$\zeta=\zeta_{\mathrm{l}}$且返回第2步.
步骤	CKM算法计算c_{r}, $c_{\mathrm{r}}=\max_{\forall\theta(x)\in[\underline{\mu}_{\tilde{A}}(x),\bar{\mu}_{\tilde{A}}(x)]}\left(\frac{\int_a^b x\theta(x)\mathrm{d}x}{\int_a^b\theta(x)\mathrm{d}x}\right)$.
1	初始化$\theta(x)=[\underline{\mu}_{\tilde{A}}(x)+\bar{\mu}_{\tilde{A}}(x)]/2$, 计算初始值$\zeta$, 使$\zeta=\frac{\int_a^b x\theta(x)\mathrm{d}x}{\int_a^b\theta(x)\mathrm{d}x}$.
2	当$x\leqslant\zeta$, 设置$\theta(x)=\underline{\mu}_{\tilde{A}}(x)$; 当$x\geqslant\zeta$, 设置$\theta(x)=\bar{\mu}_{\tilde{A}}(x)$, 计算$\zeta_{\mathrm{r}}=\frac{\int_a^\zeta x\underline{\mu}_{\tilde{A}}(x)\mathrm{d}x+\int_\zeta^b x\bar{\mu}_{\tilde{A}}(x)\mathrm{d}x}{\int_a^\zeta\underline{\mu}_{\tilde{A}}(x)\mathrm{d}x+\int_\zeta^b\bar{\mu}_{\tilde{A}}(x)\mathrm{d}x}$.
3	核对是否$\lvert\zeta-\zeta_{\mathrm{r}}\rvert<\varepsilon$ (ε为给定的算法误差边界), 若成立, 终止且设置$c_{\mathrm{r}}=\zeta_{\mathrm{r}}$, 若不成立, 转入第4步.
4	设置$\zeta=\zeta_{\mathrm{r}}$, 且返回第2步.

在表4中, 除了KM算法的权重值分配, 3种WKM算法的权重值分配可从式(12)–(14)通过如下的步骤获得:

1) 由$x_i(i=1,\cdots,N)$, $x_1=a$, $x_N=b$取代式(12)中的$x_{\mathrm{l}}(l=0,1,\cdots,n)$, $x_0=a$, $x_n=b$; 式(13)中的$x_{\mathrm{l}}(l=0,1,\cdots,2n)$, $x_0=a$, $x_{2n}=b$; 式(14)中的$x_{\mathrm{l}}(l=0,1,\cdots,3n)$, $x_0=a$, $x_{3n}=b$.

2) 表2中两个积分之间的商运算可抵消式(12)–(14)中的系数, 即$h/2$, $h/3$, $3h/8$.

3) 表4中TWKM和SWKM算法的权重值取式(12)–(13)的扩号中系数的1/2, 而S3/8WKM算法的权重值取式(14)的扩号中系数的1/3.

4) SWKM和S3/8WKM算法的采样点数N不局限于$N=2n+1$和$N=3n+1$(像式(13)–(14)所要求

的满足$N=1\mathrm{mod}(2)$和$N=1\mathrm{mod}(3)$).

表 3 WKM算法计算一个区间二型模糊集的质心

Table 3 Compute the centroid of an interval type–2 fuzzy set by WKM algorithms

步骤	WKM算法计算c_l, $c_\mathrm{l}=\min\limits_{\forall\theta_i\in[\underline{\mu}_{\tilde{A}}(x_i),\bar{\mu}_{\tilde{A}}(x_i)]}(\sum\limits_{i=1}^{N}w_ix_i\theta_i/\sum\limits_{i=1}^{N}w_i\theta_i)$.
1	初始化θ_i, 设置$\theta_i=[\underline{\mu}_{\tilde{A}}(x_i)+\bar{\mu}_{\tilde{A}}(x_i)]/2,\ i=1,\cdots,N$, 计算$c'=c(\theta_1,\cdots,\theta_N,w_1,\cdots,w_N)=\sum\limits_{i=1}^{N}w_ix_i\theta_i/\sum\limits_{i=1}^{N}w_i\theta_i$.
2	找到$k(1\leqslant k\leqslant N-1)$满足$x_k\leqslant c'\leqslant x_{k+1}$
3	当$i\leqslant k$, 设置$\theta_i=\bar{\mu}_{\tilde{A}}(x_i)$; 当$i\geqslant k+1$, 设置$\theta_i=\underline{\mu}_{\tilde{A}}(x_i)$, 计算$c_\mathrm{l}(k)=\dfrac{\sum\limits_{i=1}^{k}w_ix_i\bar{\mu}_{\tilde{A}}(x_i)+\sum\limits_{i=k+1}^{N}w_ix_i\underline{\mu}_{\tilde{A}}(x_i)}{\sum\limits_{i=1}^{k}w_i\bar{\mu}_{\tilde{A}}(x_i)+\sum\limits_{i=k+1}^{N}w_i\underline{\mu}_{\tilde{A}}(x_i)}$
4	核对是否$c_\mathrm{l}(k)=c'$, 若成立, 终止且设置$c_\mathrm{l}(k)=c_\mathrm{l}$, $k=L$; 若不成立, 转入第5步
5	设置$c'=c_\mathrm{l}(k)$且返回第2步

步骤	WKM算法计算c_r, $c_\mathrm{r}=\max\limits_{\forall\theta_i\in[\underline{\mu}_{\tilde{A}}(x_i),\bar{\mu}_{\tilde{A}}(x_i)]}(\sum\limits_{i=1}^{N}w_ix_i\theta_i/\sum\limits_{i=1}^{N}w_i\theta_i)$
1	初始化θ_i, 设置$\theta_i=[\underline{\mu}_{\tilde{A}}(x_i)+\bar{\mu}_{\tilde{A}}(x_i)]/2,\ i=1,\cdots,N$. 计算$c'=c(\theta_1,\cdots,\theta_N,w_1,\cdots,w_N)=\sum\limits_{i=1}^{N}w_ix_i\theta_i/\sum\limits_{i=1}^{N}w_i\theta_i$.
2	找到$k(1\leqslant k\leqslant N-1)$满足$x_k\leqslant c'\leqslant x_{k+1}$.
3	当$i\leqslant k$, 设置$\theta_i=\underline{\mu}_{\tilde{A}}(x_i)$; 当$i\geqslant k+1$, 设置$\theta_i=\bar{\mu}_{\tilde{A}}(x_i)$, 计算$c_\mathrm{r}(k)=\dfrac{\sum\limits_{i=1}^{k}w_ix_i\underline{\mu}_{\tilde{A}}(x_i)+\sum\limits_{i=k+1}^{N}w_ix_i\bar{\mu}_{\tilde{A}}(x_i)}{\sum\limits_{i=1}^{k}w_i\underline{\mu}_{\tilde{A}}(x_i)+\sum\limits_{i=k+1}^{N}w_i\bar{\mu}_{\tilde{A}}(x_i)}$.
4	核对是否$c_\mathrm{r}(k)=c'$, 若成立, 终止且设置$c_\mathrm{r}(k)=c_\mathrm{r}$, $k=R$; 若不成立, 转入第5步.
5	设置$c'=c_\mathrm{r}(k)$且返回第2步.

根据表4和式(12)–(14)的关系, TWKM, SWKM和S3/8WKM算法分别以一、二、三阶多项式估计数值积分隶属函数, 如第2.1节所解释的, 它们是NewtonCotes公式的特例.

表3中的CKM算法和表4中的WKM算法在计算区间二型模糊集的质心(完成区间二型模糊逻辑系统质心降型)之间联系为:

1) WKM算法基于应用在采用数据x_i $(i=1,\cdots,N)$上的求和运算来计算质心值, 且当迭代中止时找到用来估计质心值的最优转折点. 而CKM算法用积分运算来计算质心值, 且获得区间二型模糊集的准确质心值. 在理论上, 当采样大小$N\rightarrow+\infty$时, WKM算法的解趋于CKM算法.

2) 对于WKM算法, 增加采样值大小N会得到更准确的计算结果. 而对于CKM算法, 通过设置更小的误差边界ε可控制两相邻迭代值之差以获得更准确的计算结果.

3) WKM算法以求和运算进行数值计算, 而CKM算法以求积分运算进行象征性地计算. WKM算法是利用数值积分方法的CKM算法的数值实现.

表 4 WKM算法的权重赋值方法

Table 4 Weight assignment method of WKM algorithms

算法	积分法则	权重值
KM	—	$w_i=1\ (i=1,\cdots,N)$
TWKM	复合梯形法则	$w_i=\begin{cases}1/2,\ i=1,N,\\ 1,\ \ i\neq1,N.\end{cases}$
SWKM	复合Simpson法则	$w_i=\begin{cases}1/2,\ i=1,N,\\ 1,\ i=1\ \mathrm{mod}\ (2),\ i\neq1,N,\\ 2,\ i=0\ \mathrm{mod}\ (2),\ i\neq N.\end{cases}$
S3/8WKM	复合Simpson3/8法则	$w_i=\begin{cases}1/3,\ i=1,N,\\ 2/3,\ i=1\ \mathrm{mod}\ (3),\ i\neq1,N,\\ 1,\ i=2\ \mathrm{mod}\ (3),\ i\neq N,\\ 1,\ i=0\ \mathrm{mod}\ (2),\ i\neq N.\end{cases}$

注: mod表示求余算子. $i=j\ \mathrm{mod}\ (d)$表示$i=nd+j$, 其中n为一个整数.

4 仿真(Simulation)

本节给出3个数值仿真例子. 在前两个例子中, FOU是从两条模糊规则中导出且由分段线性函数或高斯隶属函数[10,15]所限定. 在第3个例子中, FOU取成具有不确定标准偏差的高斯二型主隶属函数[11]. 即所研究的二型模糊逻辑系统的FOU在降型前已经被确定. 假定主变量$x\in[0,10]$且x被均匀采样, 取算法误差边界$\varepsilon=10^{-6}$.

例 1 FOU由分段线性函数组成.

如图1所示, FOU的上边界是由两个分段线性函数之间取大组成, 即

$$u_1(x)=\begin{cases}\dfrac{x-1}{2},\ 1\leqslant x\leqslant 3,\\ \dfrac{7-x}{4},\quad 3<x\leqslant 7,\\ 0,\qquad (x<1)\cup(x>7),\end{cases}\tag{17}$$

$$u_2(x)=\begin{cases}\dfrac{x-2}{5},\quad 2\leqslant x\leqslant 6,\\ \dfrac{16-2x}{5},\ 6<x\leqslant 8,\\ 0,\qquad (x<2)\cup(x>8).\end{cases}\tag{18}$$

FOU的下边界是由另两个分段线性函数之间取大组成, 即

$$u_3(x)=\begin{cases}\dfrac{x-1}{6}, & 1\leqslant x\leqslant 4,\\ \dfrac{7-x}{6}, & 4<x\leqslant 7,\\ 0, & (x<1)\cup(x>7),\end{cases} \tag{19}$$

$$u_4(x)=\begin{cases}\dfrac{x-3}{6}, & 3\leqslant x\leqslant 5,\\ \dfrac{8-x}{9}, & 5<x\leqslant 8,\\ 0, & (x<3)\cup(x>8),\end{cases} \tag{20}$$

其中符号∪表示或.

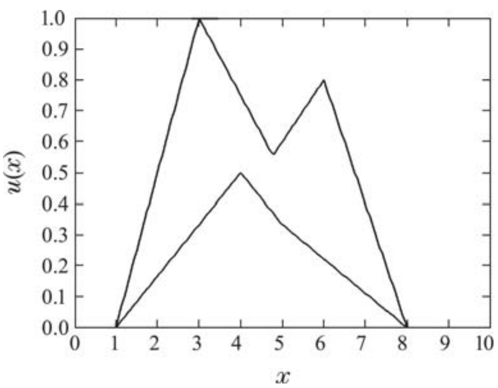


图 1 例1的FOU

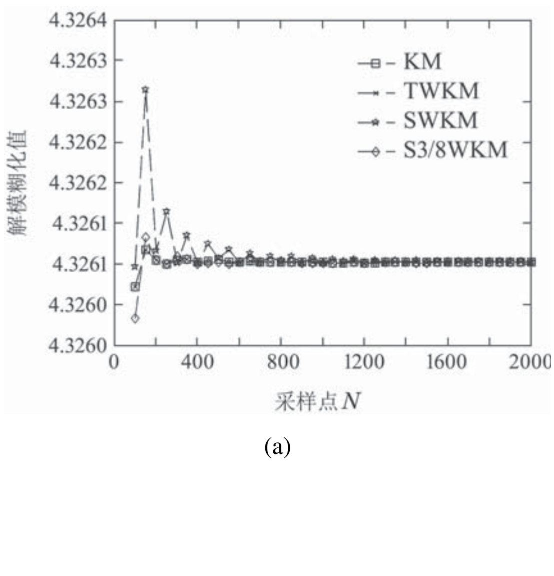
Fig. 1 FOU of example one

主变量x的采样点个数N被确定,

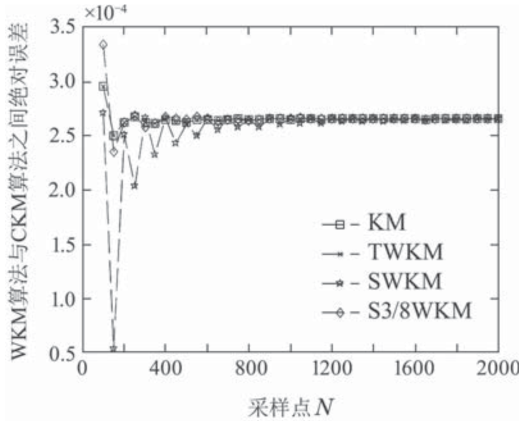
$$[x_1,x_N]=[a,b],$$

$$x_i=x_1+\frac{i-1}{N-1}(b-a),$$

且N以步长为50从100到2000变化. 4种WKM算法的解模糊化结果在图2(a)中给出, 且以WKM算法与CKM算法之间绝对误差$|y_1-y_1^*|$为因变量, N为自变量的函数图在图2(b)中给出.



(a)



(b)

图 2 例1的计算结果

Fig. 2 Computation results of example one

例 2 FOU由高斯隶属函数组成.

如图3所示, FOU的上边界是由两个高斯隶属函数之间取大组成, 即

$$u_1(x)=\exp(-\frac{(x-3)^2}{8}), \tag{21}$$

$$u_2(x)=0.8\exp(-\frac{(x-6)^2}{8}). \tag{22}$$

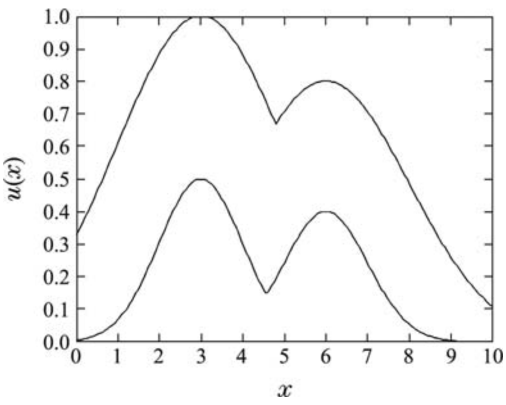


图 3 例2的FOU

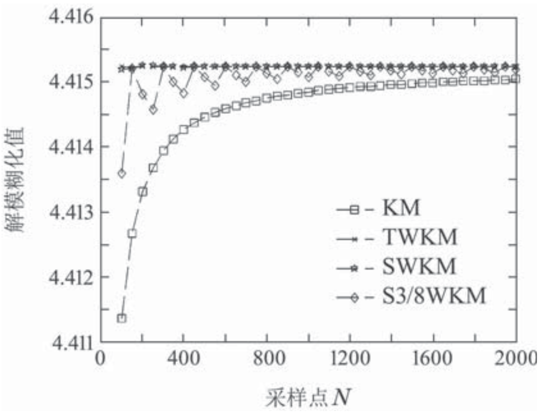
Fig. 3 FOU of example two

FOU的下边界是由另两个高斯隶属函数之间取大组成, 即

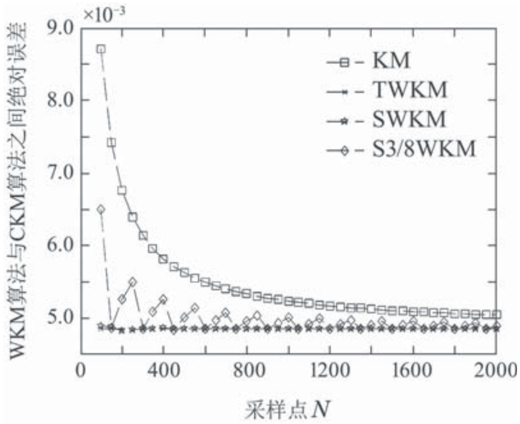
$$u_3(x)=0.5\exp(-\frac{(x-3)^2}{2}), \tag{23}$$

$$u_4(x)=0.4\exp(-\frac{(x-6)^2}{2}). \tag{24}$$

首先用CKM算法计算出的准确的解模糊化值$y_2^*=4.420076$. 4种WKM算法的解模糊化结果在图4(a)中给出, 且以WKM算法与CKM算法之间绝对误差$|y_2-y_2^*|$为因变量, N为自变量的函数图在图4(b)中给出.



(a)



(b)

图 4　例2的计算结果

Fig. 4　Computation results of example two

例 3　FOU为具有不确定标准偏差的高斯二型主隶属函数.

如图5所示, FOU的上边界为高斯函数, 即

$$u_1(x) = \exp[-\frac{1}{2}(\frac{x-5}{1.75})^2]. \tag{25}$$

FOU的下边界为另一个高斯函数, 即

$$u_2(x) = \exp[-\frac{1}{2}(\frac{x-5}{0.25})^2]. \tag{26}$$

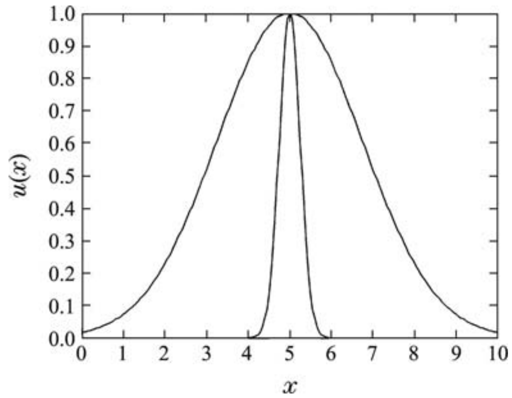
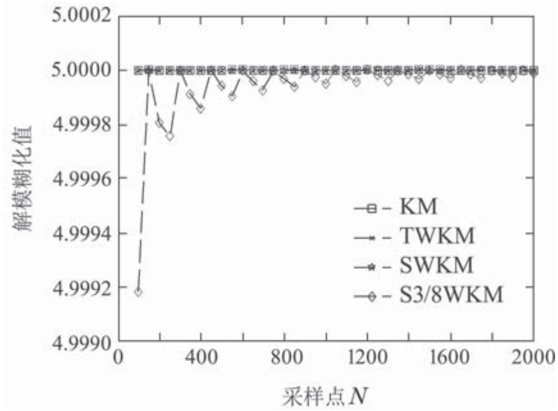


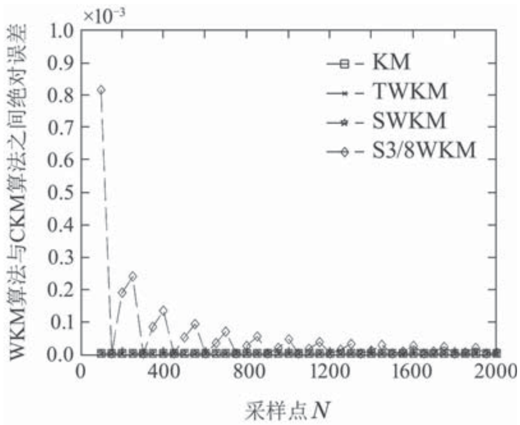
图 5　例3的FOU

Fig. 5　FOU of example three

由CKM算法计算出的准确的解模糊化值$y_3^* =$ 4.999996. 4种WKM算法的解模糊化结果在图6(a)中给出, 且以WKM算法与CKM算法之间绝对误差$|y_3 - y_3^*|$为因变量, N为自变量的函数图在图6(b)中给出.



(a)



(b)

图 6　例3的计算结果

Fig. 6　Computation results of example three

表5中给出CKM算法计算3个例子的质心区间的左和右质心区间y_l和y_r的具体迭代结果.

表 5　对3个例子的y_l和y_r的CKM迭代算法计算结果($\varepsilon = 10^{-6}$)

Table 5　Computation results of CKM iteration algorithms for three examples of y_l and y_r($\varepsilon = 10^{-6}$)

t	0	1	2	3	4	5
$y_{1\mathrm{l}}$	4.320794	3.679657	3.661355	3.661338	3.661338	
$y_{1\mathrm{r}}$	4.320794	4.975022	4.991388	4.991396	4.991396	
$y_{2\mathrm{l}}$	4.395260	3.255309	3.156405	3.155741	3.155741	
$y_{2\mathrm{r}}$	4.395260	5.576350	5.683753	5.684411	5.684411	
$y_{3\mathrm{l}}$	4.999999	3.819380	3.606842	3.595568	3.595539	3.595539
$y_{3\mathrm{r}}$	4.999999	6.180619	6.393158	6.404423	6.404453	6.404453

注: t为迭代次数, $t = 0$代表初始化过程.

为了再次衡量KM, TWKM, SWKM, S3/8WKM算法在3个例子中的表现, 对每个采样值N, 定义并且计算相对误差$|y_i - y_i^*|/|y_i^*|(i = 1, 2, 3)$. 3个例子的与采样个数$N$相关的相对误差平均值在表6中给出, 并且表6中最后一行给出每种算法在3个例子下的总平均相对误差平均值.

表 6 当$N = 100 : 50 : 2000$时, 相对误差$|y_i - y_i^*|/|y_i^*|(i = 1, 2, 3)$的平均值

Table 6 Mean relative error $|y_i - y_i^*|/|y_i^*|(i = 1, 2, 3)$ for $N = 100 : 50 : 2000$

算法	KM	TWKM	SWKM	S3/8WKM
例1	0.00006126	0.00006126	0.00005886	0.00006145
例2	0.00124329	0.00109645	0.00109676	0.00112994
例3	0.00000086	0.00000086	0.00000086	0.00001125
总平均值	0.00043513	0.00038619	0.00038549	0.00040088

从图2, 4, 6, 8和表6, 可得到如下结论:

1) 观察图2, 4, 6和8, 4种算法的绝对误差都收敛. 在例1中, SWKM算法得到最小的绝对误差(变化幅度最大), KM算法和TWKM算法得到几乎相同的绝对误差(变化幅度很小), 而S3/8 WKM算法得到最大的绝对误差(变化幅度也较大). 在例2中, 所有WKM算法的绝对误差小于KM算法, 其中TWKM算法和SWKM算法的收敛速度也快于KM算法. TWKM算法和SWKM算法的绝对误差最小(两者几乎相同且变化幅度很小), S3/8 WKM算法的绝对误差居中(变化幅度较大), 而KM算法的绝对误差最大(变化幅度最大). 在例3中, S3/8WKM算法的绝对误差最大(变化幅度最大), 另3种算法的绝对误差较小且几乎相同(变化幅度很小).

2) 观察表6, KM算法的最大平均相对误差为0.124329%, 而WKM算法的最大平均相对误差为0.112994%; KM算法的总平均相对误差为0.043513%, 而WKM算法的最大总平均相对误差为0.040088%.

3) 从1)和2)的分析可知, 当选择恰当的WKM算法时, 其计算精度和误差稳定性均优于KM算法.

为了更好地应用这些算法, 下面研究算法的计算时间. 与计算解模糊化值不同, 算法的计算时间取决于具体的软件和硬件环境, 且它们的计算结果是不可重复的. 仿真平台为Microsoft Windows XP Professional系统, 具有E5300@2.60GHz和2.00 GB内存的双核CPU的戴尔台式机. 算法由MATLAB 2013a编程, 图7–9显示了在采样点个数等于100 : 50 : 2000下的总计算时间.

如果不考虑不同的采样个数N对计算时间的波动影响, 4种算法大约随着采样值N的大小变化呈线性方式变化. 可取得算法的最小二乘回归模型$t = a + bN$, 其中t为计算时间, 回归系数在表7中给出. 定义4种算法的计算时间差异率为

$$(\max_{i=1,\cdots,4}\{t_i\} - \min_{i=1,\cdots,4}\{t_i\})/\max_{i=1,\cdots,4}\{t_i\}, \quad (27)$$

其中$t_i(i = 1, \cdots, 4)$表示4种算法的计算时间.

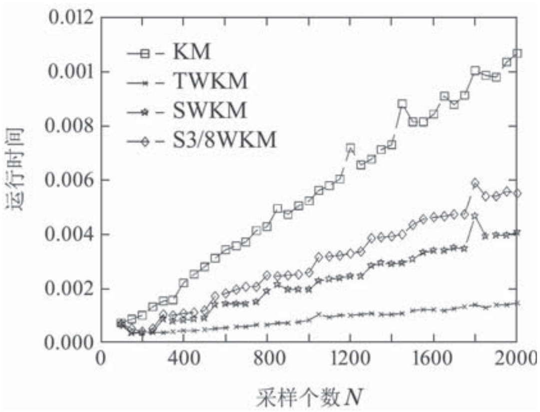


图 7 例1的算法运行时间比较

Fig. 7 Comparison of run time in example one

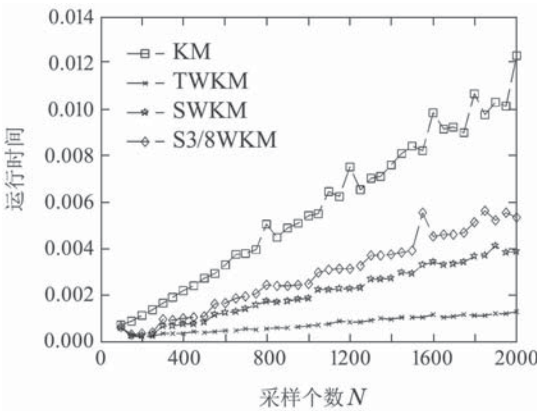


图 8 例2的算法运行时间比较

Fig. 8 Comparison of run time in example two

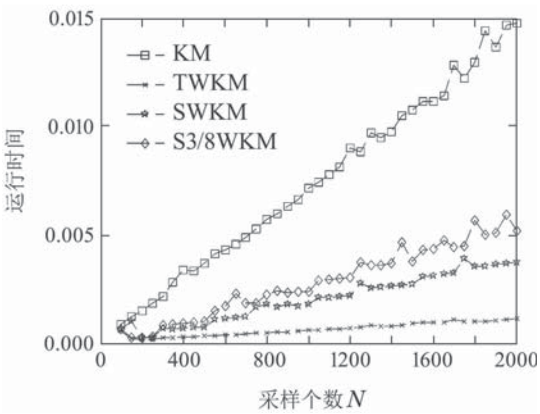


图 9 例3的算法运行时间比较

Fig. 9 Comparison of run time in example three

从图7–9和表7可观察到, 总体来说, WKM算法的计算速度快于KM算法. 4种算法计算速度的大小关

系为TWKM > SWKM > S3/8WKM > KM. 其中TWKM算法的计算速度最快, SWKM算法的计算速度稍快于S3/8WKM算法, 这是因为前者的权重分配方法比后者略简单. 当采样个数$N = 100 : 50 : 2000$时, 4种算法在所阐述的3个例子下的计算时间差异率为9.24%∼92.77%.

本文所提出的算法既可用来研究二型逻辑系统的降型又可研究估计理论. 如果只考虑计算准确度需求, 观察表6中的计算准确度统计, 得出SWKM算法是最好的选择. 二型模糊逻辑系统的设计过程中需要实时计算且采样率$1/N$通常是固定的. 例1为涉及到线性函数的二型模糊逻辑系统降型, 具有如图1所示的FOU. 例2和例3为涉及到非线性函数的二型模糊逻辑系统的降型, 具有如图3–5所示的FOU. 同时考虑表6中的计算准确度统计和图7–9, 建议对涉及线性函数和非线性函数的二型模糊逻辑系统降型计算使用TWKM算法. 估计理论需要得到高精度且采样率$1/N$是改变的. 所以建议使用SWKM算法研究估计理论.

表 7 4种算法的最小二乘计算回归模型系数

Table 7 Compute the regression model coefficients by least squre for four types of algorithms

回归系数	KM		TWKM		SWKM		S3/8WKM	
	$a/10^{-3}$	$b/10^{-3}$	$a/10^{-3}$	$b/10^{-3}$	$a/10^{-3}$	$b/10^{-3}$	$a/10^{-3}$	$b/10^{-3}$
例1	0.0053	0.0641	0.0006	0.2345	0.0020	0.0851	0.0029	−0.0028
例2	0.0056	−0.0160	0.0006	0.1661	0.0020	−0.0136	0.0028	−0.0837
例3	0.0072	0.0244	0.0005	0.1435	0.0019	0.0926	0.0028	−0.1123
平均值	0.0060	0.0242	0.0006	0.1814	0.0020	0.0547	0.0028	−0.0663

最后需要指出, 在KM算法和WKM算法之间的比较上, 本文只关注这些算法的理论表现. 3个数值例子可得出, 当使用相当的采样点个数时, 3种WKM算法与KM算法相比在计算准确度上有较大的提高. 尽管如此, 如果准确度需求并不高, 简单的KM算法就可以得到足够好的结果, 那么WKM算法的优势就无从体现了.

5 结论与展望(Conclusion and expectation)

文献[11]比较了离散版本和连续版本的EKM法, 而本文比较了离散版本和连续版本的KM算法. 当区间二型模糊逻辑系统的输出集的FOU完全确定时, 连续版本的KM算法可用来准确地完成该系统的质心降型. 结合数值积分技术, 利用3种加权分配方法下的估计法则将KM算法扩展成加权KM(WKM)算法. 3个数值例子对4种算法的计算准确度和计算时间进行了分析和说明. 在相同的采样率下, WKM算法与KM算法相比具有较小的绝对误差和较快的收敛速度.

在以后的工作中, 基于本文和文献[11, 13–14], 作者将进一步研究利用WKM算法、WEKM算法设计区间二型模糊逻辑系统和普通二型模糊逻辑系统的降型[8, 16], 利用智能优化算法[17–18]完成区间二型模糊逻辑系统和普通二型模糊逻辑系统的优化等.

参考文献(References):

[1] PAN Yongping, SUN Zonghai, HUANG Daoping. A survey of type–2 fuzzy sets and systems [J]. *Control Theory & Applications*, 2011, 28(12): 1693 – 1703.
(潘永平, 孙宗海, 黄道平. II型模糊集合与系统研究进展 [J]. 控制理论与应用, 2011, 28(12): 1693 – 1703.)

[2] MENDEL J M. *Uncertain Rule-Based Fuzzy Logic systems: Introduction and New Directions* [M]. Englewood Cliffs, NJ, USA: Prentice-Hall, 2001: 1 – 547.

[3] CHEN Y, WANG D Z, TONG S C. Forecasting studies by designing Mamdani interval type–2 fuzzy logic systems [J]. *Neurocomputing*, 2016, 174(part b): 1133 – 1146.

[4] HAGRAS H, WAGNER C. Towards the wide spread use of type–2 fuzzy logic systems in real world applications [J]. *IEEE Computational Intelligent Magazine*, 2012, 7(3): 14 – 24.

[5] MENDEL J M. On KM algorithms for solving type–2 fuzzy set problems [J]. *IEEE Transactions on Fuzzy Systems*, 2013, 21(3): 426 – 446.

[6] KARNIK K K, MENDEL J M. Centroid of a type–2 fuzzy set [J]. *Information Sciences*, 2001, 132(1): 195 – 220.

[7] MENDEL J M. Advances in type–2 fuzzy sets and systems [J]. *Information Sciences*, 2007, 177(1): 84 – 110.

[8] MENDEL J M. General type–2 fuzzy logic systems made simple: A tutorial [J]. *IEEE Transactions on Fuzzy Systems*, 2014, 22(5): 1162 – 1182.

[9] MENDEL J M, WU H W. Type–2 fuzzistics for symmetric interval type–2 fuzzy sets: Part 1, forward problems [J]. *IEEE Transactions on Fuzzy Systems*, 2007, 14(6): 360 – 377.

[10] MENDEL J M, LIU F L. Super-exponential convergence of the Karnik-Mendel algorithms for computing the centroid of an interval type–2 fuzzy set [J]. *IEEE Transactions on Fuzzy Systems*, 2007, 15(2): 309 – 320.

[11] LIU X W, MENDEL J M, WU D R. Stuy on enhanced Karnik-Mendel algorithms: Initialization explanations and computation improvements [J]. *Information Sciences*, 2012, 184(1): 75 – 91.

[12] MATHEWS J H, FINK K D. *Numerical Methods Using MATLAB* [M]. Upper Saddle River, NJ: Prentice-Hall Inc, 2004: 1 – 655.

[13] HU Huaizhong, ZHAO Ge, YANG Huanan. Fast algorithm to calculate generalized centroid of interval type–2 fuzzy set [J]. *Control and Decision*, 2010, 25(4): 637 – 640.
(胡怀中, 赵戈, 杨华南. 一种区间型二型模糊集的快速解法 [J]. 控制与决策, 2010, 25(4): 637 – 640.)

[14] HU H Z, WANG Y, CAI Y L. Advantages of the enhanced opposite direction searching algorithm for computing the centroid of an interval type–2 fuzzy set [J]. *Asian Journal of Control*, 2012, 14(5): 1422 – 1430.

[15] MENDEL J M, WU H W. New results about the centroid of an interval type–2 fuzzy set, including the centroid of a fuzzy granule [J]. *Information Sciences*, 2007, 177(2): 360 – 377.

[16] LIU F L. An efficient centroid type-reduction strategy for general type–2 fuzzy logic system [J]. *Information Sciences*, 2008, 178(1): 2224 – 2236.

[17] WU D R, TAN W W. Genetic learning and performance evaluation of interval type–2 fuzzy logic controllers [J]. *Engineering Applications of Artificial Intelligence*, 2006, 19(8): 829 – 841.

[18] ZHAI D Y, HAO M S, MENDEL J M. Universal image noise removal filter based on type–2 fuzzy logic system and QPSO [J]. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 2012, 20(supp02): 207 – 232.

[19] ZHANG Biao, ZHOU Shaosheng. Stability analysis and control design for interval type–2 stochastic fuzzy systems [J]. *Control Theory & Applications*, 2015, 32(7): 985 – 992.
(张彪, 周绍生. 区间二型随机模糊系统的稳定性分析和控制设计 [J]. 控制理论与应用, 2015, 32(7): 985 – 992.)

作者简介:

陈　阳　(1981–), 男, 讲师, 博士研究生, 主要研究方向为二型模糊逻辑系统、模糊推理及其控制、永磁驱动等, E-mail: chenyanghanyun@163.com;

王大志　(1963–), 男, 教授, 博士生导师, 研究方向为电力系统与电力传动, E-mail: wangdazhi@ise.neu.edu.cn.