Chapter 8
Type-Reduction

Type-Reduction (TR)

- TR is the mapping of a T2 FS into a T1 FS
- For a GT2 FS the result of TR is a T1 FS
- For an IT2 FS the result of TR is a T1 interval fuzzy number

Interval Weighted Average (IWA)

$$y = \frac{\sum_{i=1}^{n} x_i w_i}{\sum_{i=1}^{n} w_i}$$

$$x_i \in [a_i, b_i] \qquad w_i \in [c_i, d_i]$$

$$Y_{IWA} = \frac{\sum_{i=1}^{n} X_i W_i}{\sum_{i=1}^{n} W_i} = [y_l, y_r]$$

$$y_l = \min_{\forall w_i \in [c_i, d_i]} \frac{\sum_{i=1}^{n} a_i w_i}{\sum_{i=1}^{n} w_i} \qquad y_r = \max_{\forall w_i \in [c_i, d_i]} \frac{\sum_{i=1}^{n} b_i w_i}{\sum_{i=1}^{n} w_i}$$

Computing the IWA: 1

1. Can't use interval arithmetic because the w_i appear in both the numerator and denominator of y.

2. Can't use calculus because, when the derivative of y with respect to w_k is set equal to zero, the result no longer depends upon w_k.

3. Need to examine the derivative of y with respect to w_k and develop algorithms for solving the two optimization problems

Computing the IWA: 2

$$\frac{\partial y(w_1,\ldots,w_n)}{\partial w_k} = \frac{\partial}{\partial w_k}\left[\frac{\sum_{i=1}^{n} x_i w_i}{\sum_{i=1}^{n} w_i}\right] = \frac{\partial}{\partial w_k}\left[\frac{x_k w_k + \sum_{i=1, i\neq k}^{n} x_i w_i}{w_k + \sum_{i=1, i\neq k}^{n} w_i}\right]$$

$$= \frac{x_k\left(w_k + \sum_{i=1, i\neq k}^{n} w_i\right) - \left(x_k w_k + \sum_{i=1, l\neq k}^{n} x_i w_i\right)\times 1}{\left(w_k + \sum_{i=1, i\neq k}^{n} w_i\right)^2}$$

$$= \frac{x_k}{\sum_{i=1}^{n} w_i} - \frac{\sum_{i=1}^{n} x_i w_i}{\left(\sum_{i=1}^{n} w_i\right)^2} = \frac{x_k}{\sum_{i=1}^{n} w_i} - \frac{\sum_{l=1}^{n} x_i w_i}{\sum_{i=1}^{n} w_i} \times \frac{1}{\sum_{i=1}^{n} w_i}$$

$$= \frac{x_k - y(w_1,\ldots,w_n)}{\sum_{i=1}^{n} w_i}$$

Computing the IWA: 3

$$\frac{\partial y(w_1,\ldots,w_n)}{\partial w_k} = \frac{x_k - y(w_1,\ldots,w_n)}{\sum_{i=1}^{n} w_i}$$

$$\frac{\partial y(w_1,\ldots,w_n)}{\partial w_k} \begin{cases} \geq 0 & \text{if } x_k \geq y(w_1,\ldots,w_n) \\ < 0 & \text{if } x_k < y(w_1,\ldots,w_n) \end{cases}$$

$$\begin{cases} \text{If } x_k > y(w_1,\ldots,w_n) \\ \qquad y(w_1,\ldots,w_n) \text{ increases (decreases) as } w_k \text{ increases (decreases)} \\ \text{If } x_k < y(w_1,\ldots,w_n) \\ \qquad y(w_1,\ldots,w_n) \text{ increases (decreases) as } w_k \text{ decreases (increases)} \end{cases}$$

Computing the IWA: 4

- The maximum value that w_k can attain is d_k and the minimum value that it can attain is c_k.

- Consequently, $y(w_1,...,w_n)$ *attains its minimum value*, y_l, if

$$w_k = \begin{cases} c_k & \forall k \text{ such that } x_k > y(w_1,...,w_n) \\ d_k & \forall k \text{ such that } x_k < y(w_1,...,w_n) \end{cases}$$

- Similarly, $y(w_1,...,w_n)$ *attains its maximum value*, y_r, if

$$w_k = \begin{cases} d_k & \forall k \text{ such that } x_k > y(w_1,...,w_n) \\ c_k & \forall k \text{ such that } x_k < y(w_1,...,w_n) \end{cases}$$

Computing the IWA: 5

$$y_l = y_l(L) = \frac{\sum_{i=1}^{L} a_i d_i + \sum_{i=L+1}^{n} a_i c_i}{\sum_{i=1}^{L} d_i + \sum_{i=L+1}^{n} c_i} \qquad y_r = y_r(R) = \frac{\sum_{i=1}^{R} b_i c_i + \sum_{i=R+1}^{n} b_i d_i}{\sum_{i=1}^{R} c_i + \sum_{i=R+1}^{n} d_i}$$

$$a_1 \le a_2 \le \cdots \le a_n \qquad b_1 \le b_2 \le \cdots \le b_n$$

- *L* and *R* are called *switch points*, and it these switch points that remain to be determined
- There are no closed-form solutions for finding *L* and *R*

Computing the IWA: 6

Alternate ways to express y_l

$$y_l = \min_{k=1,2,\ldots,n} y_l(k) = \min_{k=1,2,\ldots,n} \frac{\sum_{i=1}^{k} a_i d_i + \sum_{i=k+1}^{n} a_i c_i}{\sum_{i=1}^{k} d_i + \sum_{i=k+1}^{n} c_i}$$

$$\begin{cases} y_l = \dfrac{\sum_{i=1}^{n} a_i \left\{ \delta_l^i d_i + (1-\delta_l^i) c_i \right\}}{\sum_{i=1}^{n} \left\{ \delta_l^i d_i + (1-\delta_l^i) c_i \right\}} \\ \delta_l^i = \begin{cases} 1 & \text{if } a_i \le y_l \\ 0 & \text{otherwise} \end{cases} \end{cases}$$

Computing the IWA: 7

Alternate ways to express y_r

$$y_r = \max_{k=1,2,\ldots,n} y_r(k) = \max_{k=1,2,\ldots,n} \frac{\sum_{i=1}^{k} b_i c_i + \sum_{i=k+1}^{n} b_i d_i}{\sum_{i=1}^{k} b_i c_i + \sum_{i=k+1}^{n} b_i d_i}$$

$$\begin{cases} y_r = \dfrac{\sum_{i=1}^{n} b_i \left\{ \delta_r^i c_i + (1-\delta_r^i) d_i \right\}}{\sum_{i=1}^{n} \left\{ \delta_r^i c_i + (1-\delta_r^i) d_i \right\}} \\ \delta_r^i = \begin{cases} 1 & \text{if } b_i \geq y_r \\ 0 & \text{otherwise} \end{cases} \end{cases}$$

KM Algorithms: 1

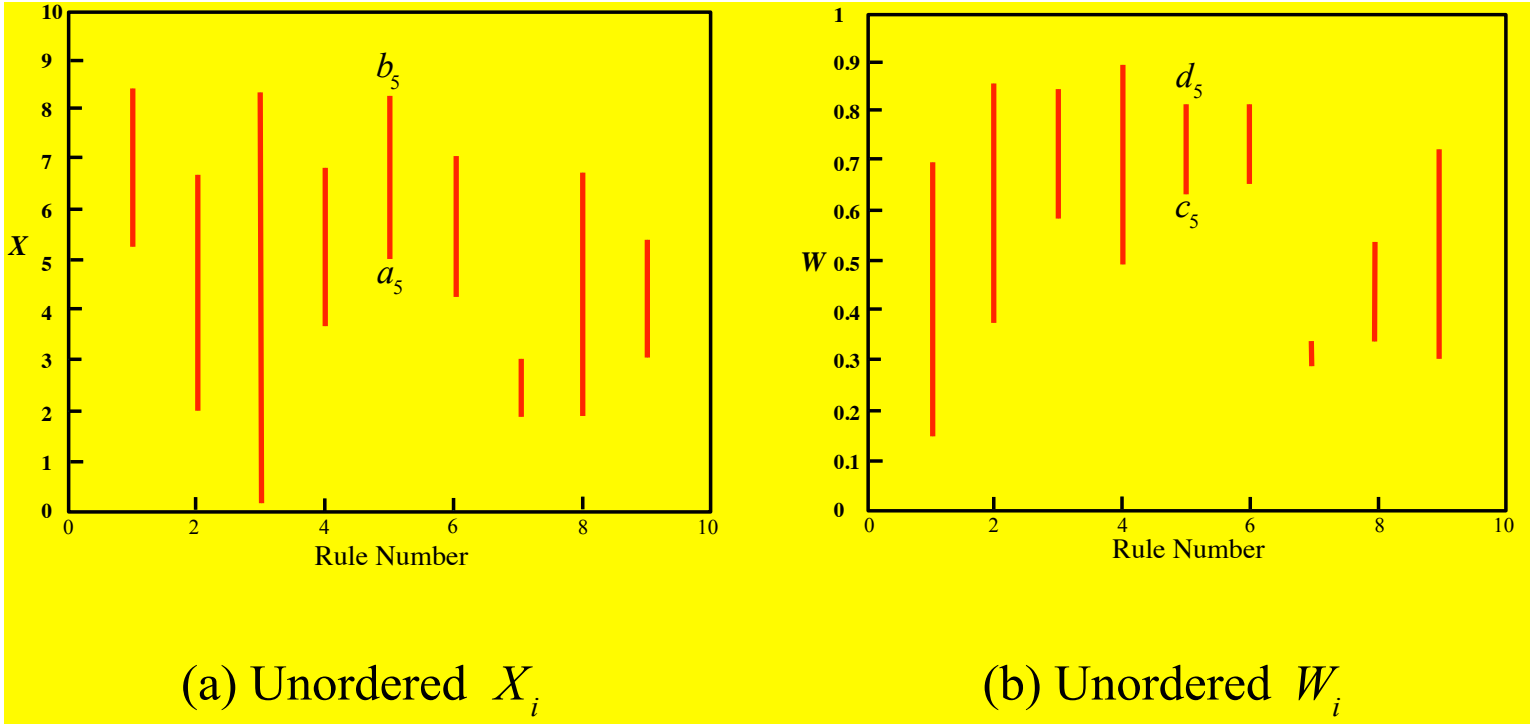
$$a_1 \le a_2 \le \cdots \le a_n \qquad b_1 \le b_2 \le \cdots \le b_n$$

Step	KM algorithm for $y_l(L)$	KM algorithm for $y_r(R)$
	$y_l(L) = \min_{\forall w_i \in [c_i, d_i]} \left(\sum_{i=1}^{n} a_i w_i \Big/ \sum_{i=1}^{n} w_i \right)$	$y_r(R) = \max_{\forall w_i \in [c_i, d_i]} \left(\sum_{i=1}^{n} b_i w_i \Big/ \sum_{i=1}^{n} w_i \right)$
1	Initialize w_i by setting $w_i = (c_i + d_i)/2$, $i = 1,\ldots,n$, and then compute $y' = y(w_1,\ldots,w_n) = \sum_{i=1}^{n} a_i w_i \Big/ \sum_{i=1}^{n} w_i$	$y' = y(w_1,\ldots,w_n) = \sum_{i=1}^{n} b_i w_i \Big/ \sum_{i=1}^{n} w_i$
2	Find $k \in \{1,2,\ldots,n-1\}$ such that $a_k \le y' \le a_{k+1}$	Find $k \in \{1,2,\ldots,n-1\}$ such that $b_k \le y' \le b_{k+1}$
3	Set $w_i = d_i$ when $i \le k$, and $w_i = c_i$ when $i \ge k+1$, and then compute $y_l(k) \equiv \dfrac{\sum_{i=1}^{k} a_i d_i + \sum_{i=k+1}^{n} a_i c_i}{\sum_{i=1}^{k} d_i + \sum_{i=k+1}^{n} c_i}$	Set $w_i = c_i$ when $i \le k$, and $w_i = d_i$ when $i \ge k+1$, and then compute $y_r(k) = \dfrac{\sum_{i=1}^{k} b_i c_i + \sum_{i=k+1}^{n} b_i d_i}{\sum_{i=1}^{k} c_i + \sum_{i=k+1}^{n} d_i}$
4	Check if $y_l(k) = y'$. If yes, stop and set $y_l(k) = y_l(L)$ and call k L. If no, go to Step 5.	Check if $y_r(k) = y'$. If yes, stop and set $y_r(k) = y_r(R)$ and call k R. If no, go to Step 5.
5	Set $y' = y_l(k)$ and go to Step 2.	Set $y' = y_r(k)$ and go to Step 2.

KM Algorithms: 2

- See Example 8.1 in the textbook for illustrations of numerical details

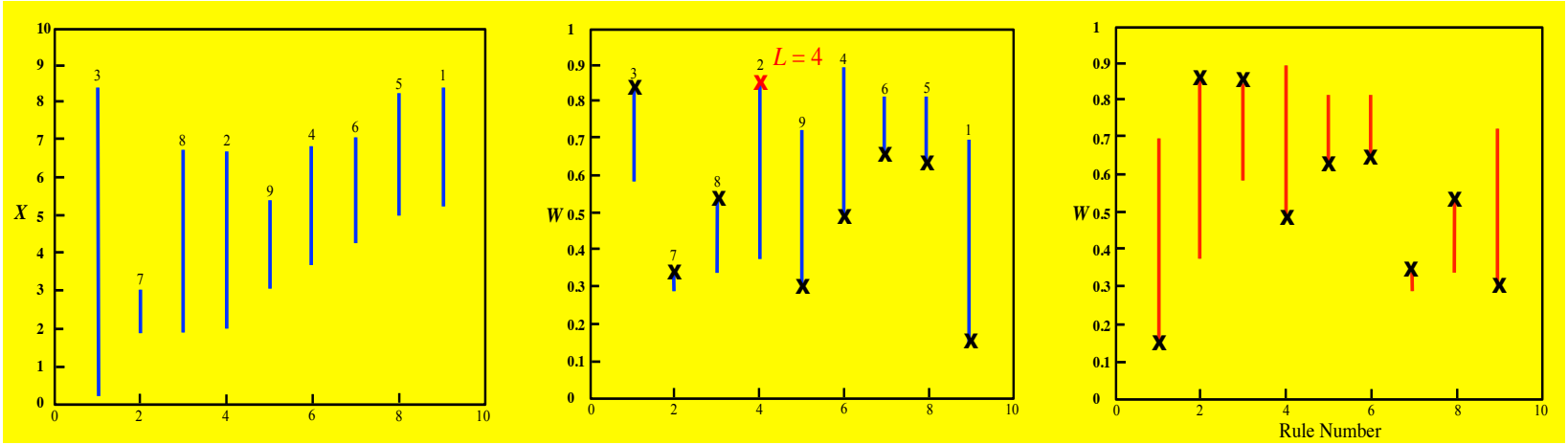
Example



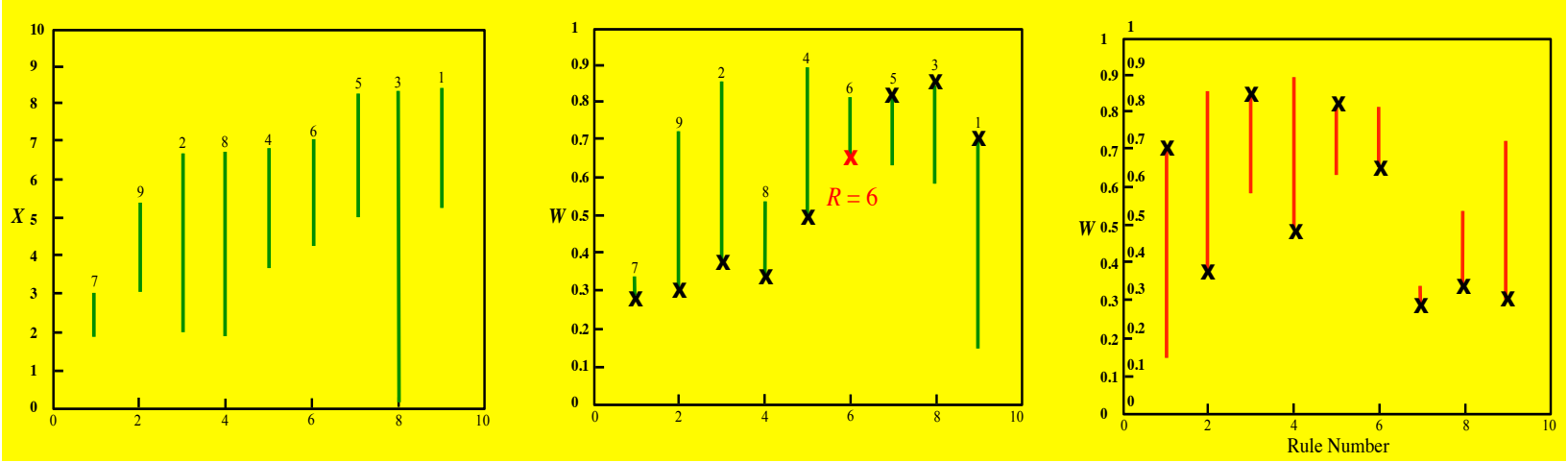
(a) Unordered X_i (b) Unordered W_i

KM Algorithms: 3

Example (Continued)



(a) Ordered X_i for $y_l(L)$ (b) Ordered W_i for $y_l(L)$ (c) Unordered W_i for $y_l(L)$



(a) Ordered X_i for $y_r(R)$ (b) Ordered W_i for $y_r(R)$ (c) Unordered W_i for $y_r(R)$

Enhanced KM Algorithms: 1

$$a_1 \le a_2 \le \cdots \le a_n \qquad b_1 \le b_2 \le \cdots \le b_n$$

Step	EKM Algorithm for $y_l(L) = \min_{\forall w_i \in [c_i, d_i]} \left(\sum_{i=1}^{n} a_i w_i \Big/ \sum_{i=1}^{n} w_i \right)$	EKM Algorithm for $y_r(R) = \max_{\forall w_i \in [c_i, d_i]} \left(\sum_{i=1}^{n} b_i w_i \Big/ \sum_{i=1}^{n} w_i \right)$
1	Set $k = [n/2.4]$ (the nearest integer to $n/2.4$) and compute: $a = \sum_{i=1}^{k} a_i d_i + \sum_{i=k+1}^{n} a_i c_i$ $b = \sum_{i=1}^{k} d_i + \sum_{i=k+1}^{n} c_i$	Set $k = [n/1.7]$ (the nearest integer to $n/1.7$) and compute $a = \sum_{i=1}^{k} b_i c_i + \sum_{i=k+1}^{n} b_i d_i$ $b = \sum_{i=1}^{k} c_i + \sum_{i=k+1}^{n} d_i$
	Compute $y' = a/b$	
2	Find $k' \in \{1,2,\ldots,n-1\}$ such that $a_{k'} \le y' \le a_{k'+1}$	Find $k' \in \{1,2,\ldots,n-1\}$ such that $b_{k'} \le y' \le b_{k'+1}$
3	Check if $k' = k$. If yes, stop and set $y' = y_l(L)$, and $k = L$. If no, go to Step 4.	Check if $k' = k$. If yes, stop and set $y' = y_r(R)$, and $k = R$. If no, go to Step 4.
4	Compute $s = sign(k' - k)$ and: $a' = a + s \sum_{i=\min(k,k')+1}^{\max(k,k')} a_i (d_i - c_i)$ $b' = b + s \sum_{i=\min(k,k')+1}^{\max(k,k')} (d_i - c_i)$	Compute $s = sign(k' - k)$ and: $a' = a - s \sum_{i=\min(k,k')+1}^{\max(k,k')} b_i (d_i - c_i)$ $b' = b - s \sum_{i=\min(k,k')+1}^{\max(k,k')} (d_i - c_i)$
	Compute $y''(k') = a'/b'$	
5	Set $y' = y''(k')$, $a = a'$, $b = b'$ and $k = k'$ and go to Step 2.	

Enhanced KM Algorithms: 2

- See Example 8.3 in the textbook for illustrations of numerical details

EIASC Algorithms: 1

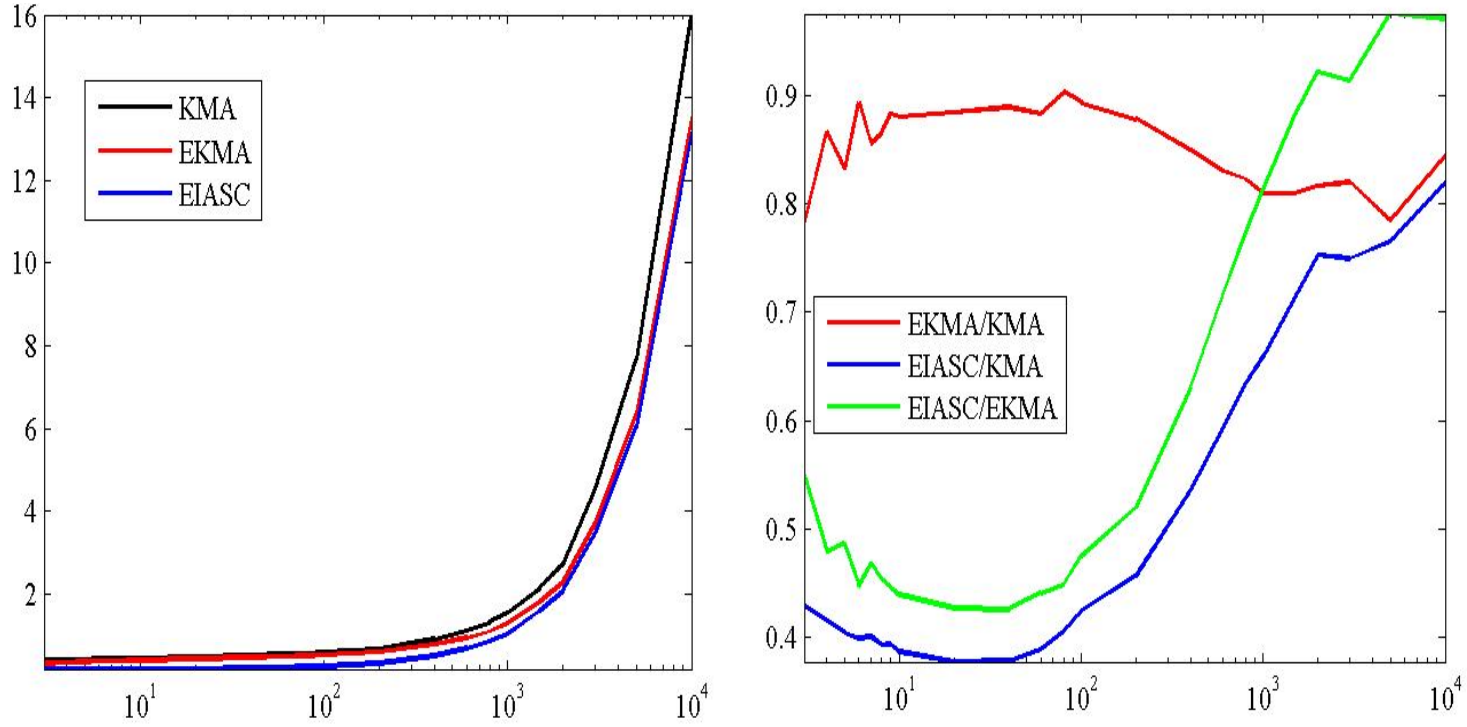
$$a_1 \le a_2 \le \cdots \le a_n \qquad b_1 \le b_2 \le \cdots \le b_n$$

Step	EIASC for $y_l(L)$ $\quad y_l(L) = \min_{\forall w_i \in [c_i, d_i]} \left(\sum_{i=1}^{n} a_i w_i \Big/ \sum_{i=1}^{n} w_i \right)$	EIASC for $y_r(R)$ $\quad y_r(R) = \max_{\forall w_i \in [c_i, d_i]} \left(\sum_{i=1}^{n} b_i w_i \Big/ \sum_{i=1}^{n} w_i \right)$
1	Initialize $a = \sum_{i=1}^{n} a_i c_i \quad b = \sum_{i=1}^{n} c_i$ $L = 0$	Initialize $a = \sum_{i=1}^{n} b_i d_i \quad b = \sum_{i=1}^{n} d_i$ $R = n$
2	Compute $L = L + 1$ $a = a + a_L (d_L - c_L)$ $b = b + (d_L - c_L)$ $y_l(L) = a / b$	Compute $a = a + b_R (d_R - c_R)$ $b = b + (d_R - c_R)$ $y_r(R) = a / b$ $R = R - 1$
3	If $y_l(L) \le a_{L+1}$, stop, otherwise go to Step 2	If $y_r(R+1) \ge b_R$, stop, otherwise go to Step 2

EIASC Algorithms: 2

- See Example 8.4 in the textbook for illustrations of numerical details
- Example 8.5 compares the computation times for KM, EKM and EIASC algorithms using a massive Monte-Carlo simulation study
- The EIASC algorithms are the fastest for $n < 1300$ (see the next slide)

EIASC Algorithms: 3



(a) Total computation time (seconds) for the 5000 Monte Carlo simulations for different values of n, and (b) ratio of the computation time of EKM to KM, EIASC to KM and EIASC to EKM.

TR for IT2 FSs and Fuzzy Systems

- Regardless of the kind of type-reduction, the following is always true: ***type-reduction is an extension of a type-1 defuzzification procedure, obtained by using the Extension Principle, in which the t-norm is either the minimum or the product***.

- For an IT2 FS, it does not matter which one of these t-norms is used because all of the secondary grades are equal to 1 (min(1,1) = 1 and 1X1 = 1).

Centroid TR for IT2 FSs: 1

Definition 8.2 The *centroid* $C_{\tilde{A}}(x)$ of $\tilde{A}$ is the union of the centroids, $c(A_e)$, of all its embedded type-1 fuzzy sets A_e, and associated with each of these numbers is a membership grade of 1, because the secondary grades of an IT2 FS are all equal to 1. This means:

$$C_{\tilde{A}}(x) = 1 \Big/ \bigcup_{\forall A_e} c_{\tilde{A}}(A_e) = 1 \Big/ \bigcup_{\forall A_e} \frac{\sum_{i=1}^{N} x_i \mu_{A_e}(x_i)}{\sum_{i=1}^{N} \mu_{A_e}(x_i)}$$

$$= 1 \big/ \{c_l(\tilde{A}), ..., c_r(\tilde{A})\} \equiv 1 \big/ [c_l(\tilde{A}), c_r(\tilde{A})]$$

where N is the number of samples of the span of the UMF of $\tilde{A}$, and

$$c_l(\tilde{A}) = \min_{\forall A_e} c_{\tilde{A}}(A_e) = \min_{\forall w_i \in \left[\underline{\mu}_{\tilde{A}}(x_i), \overline{\mu}_{\tilde{A}}(x_i)\right]} \frac{\sum_{i=1}^{N} x_i w_i}{\sum_{i=1}^{N} w_i}$$

$$c_r(\tilde{A}) = \max_{\forall A_e} c_{\tilde{A}}(A_e) = \max_{\forall w_i \in \left[\underline{\mu}_{\tilde{A}}(x_i), \overline{\mu}_{\tilde{A}}(x_i)\right]} \frac{\sum_{i=1}^{N} x_i w_i}{\sum_{i=1}^{N} w_i}$$

Centroid TR for IT2 FSs: 2

Theorem 8.1 $[c_l(\tilde{A}), c_r(\tilde{A})]$ is an IWA in which $y_l = c_l(\tilde{A})$, $y_r = c_r(\tilde{A})$, $n = N$, $a_i = b_i = x_i$, $c_i = \underline{\mu}_{\tilde{A}}(x_i)$ and $d_i = \overline{\mu}_{\tilde{A}}(x_i)$, so that:

$$c_l(\tilde{A}) = \frac{\sum_{i=1}^{L} x_i \overline{\mu}_{\tilde{A}}(x_i) + \sum_{i=L+1}^{N} x_i \underline{\mu}_{\tilde{A}}(x_i)}{\sum_{i=1}^{L} \overline{\mu}_{\tilde{A}}(x_i) + \sum_{i=L+1}^{N} \underline{\mu}_{\tilde{A}}(x_i)}$$

$$c_r(\tilde{A}) = \frac{\sum_{i=1}^{R} x_i \underline{\mu}_{\tilde{A}}(x_i) + \sum_{i=R+1}^{N} x_i \overline{\mu}_{\tilde{A}}(x_i)}{\sum_{i=1}^{R} \underline{\mu}_{\tilde{A}}(x_i) + \sum_{i=R+1}^{N} \overline{\mu}_{\tilde{A}}(x_i)}$$

Centroid TR for IT2 FSs: 3

Example 1: Gaussian primary MF with uncertain mean

$[m_1, m_2]$	$m_2 - m_1$	$[c_l, c_r]$	$c_r - c_l$	$(c_r + c_l)/2$
[5, 5]	0	[5, 5]	0	5
[4.87, 5.12]	0.25	[4.87, 5.12]	0.25	5
[4.75, 5.25]	0.5	[4.74, 5.25]	0.51	5
[4.62, 5.37]	0.75	[4.62, 5.37]	0.75	5
[4.50, 5.50]	1	[4.49, 5.50]	1.01	5
[4.25,5.75]	1.5	[4.21, 5.78]	1.57	5
[4, 6]	2	[3.90, 6.09]	2.19	5
[3.75, 6.25]	2.5	[3.55, 6.44]	2.89	5
[3.50, 6.50]	3	[3.15, 6.84]	3.69	5

- As the uncertainty about the mean increases, $c_r - c_l$ increases.
- $[c_l, c_r]$ is always symmetrical about the type-1 mean $m = 5$, and the average value of c_l and c_r is always equal to 5, regardless of the amount of uncertainty there is in m.

Centroid TR for IT2 FSs: 4

Example 2: Gaussian primary MF with uncertain SD

$[\sigma_1,\sigma_2]$	$\sigma_2-\sigma_1$	$[c_l,c_r]$	c_r-c_l	$(c_r+c_l)/2$
[1, 1]	0	[5, 5]	0	5
[0.88, 1.13]	0.25	[4.80, 5.20]	0.40	5
[0.75, 1.25]	0.5	[4.60, 5.40]	0.80	5
[0.63, 1.38]	0.75	[4.40, 5.60]	1.20	5
[0.50, 1.50]	1	[4.18, 5.81]	1.62	5
[0.38, 1.63]	1.25	[3.93, 6.07]	2.14	5
[0.25, 1.75]	1.5	[3.59, 6.41]	2.82	5

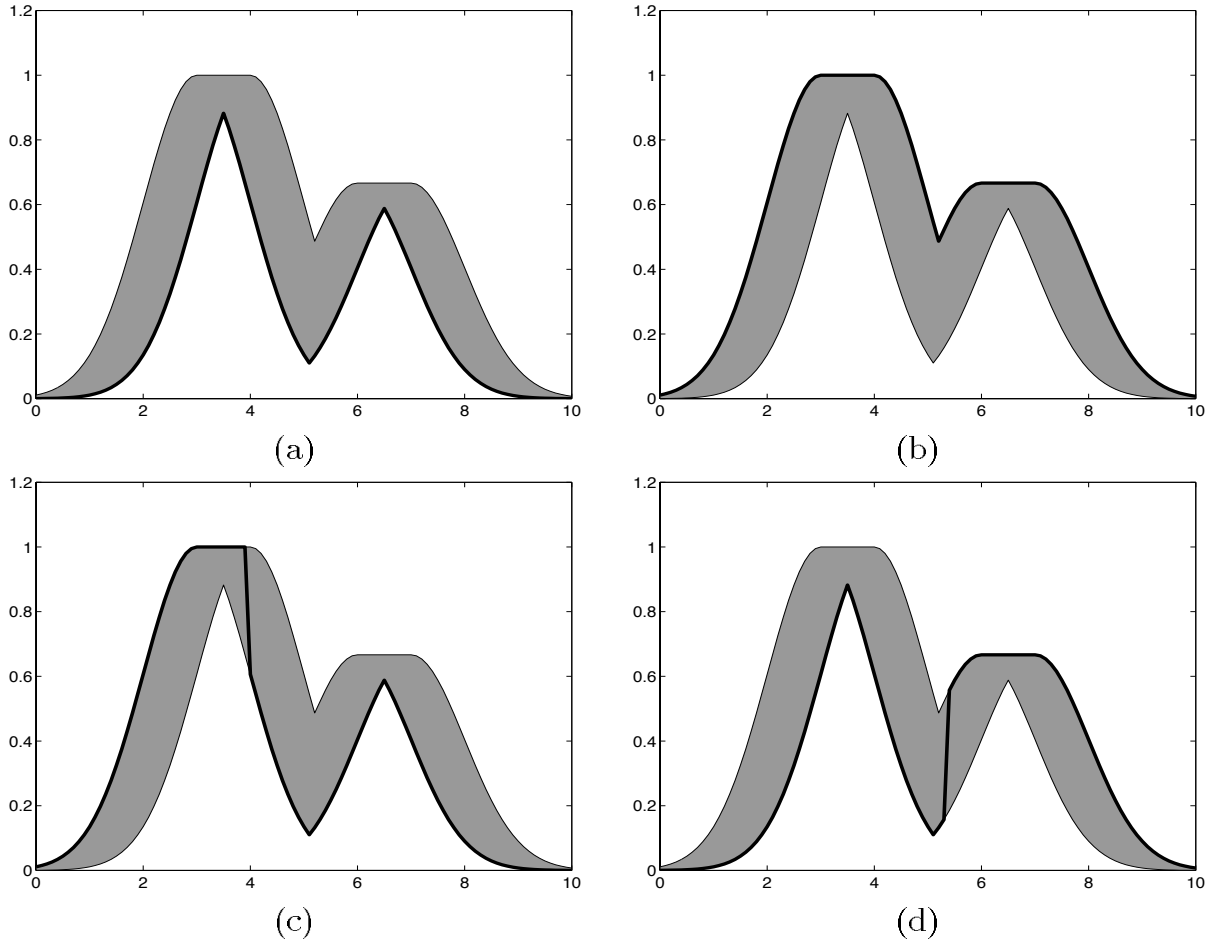
- As the uncertainty about the standard deviation increases, c_r-c_l increases.
- $[c_l,c_r]$ is again always symmetrical about the known mean $m=5$, and the average value of c_l and c_r is always equal to 5, regardless of the amount of uncertainty there is in σ.

Centroid TR for IT2 FSs: 5

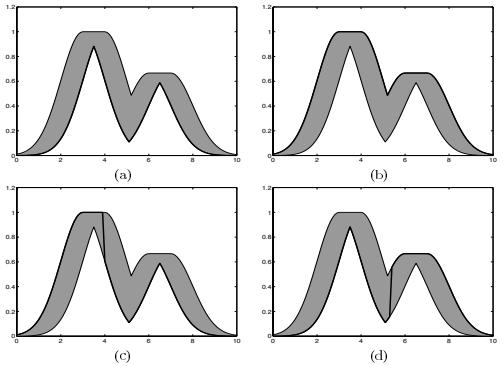
1. Given the FOU of an IT2 FS, one that is *symmetrical* about the primary variable x at $x = m$, then the centroid of such a type-2 fuzzy set is symmetrical about $x = m$, and the average value (i.e., the defuzzified value) of all the elements in the centroid equals m.
2. Given the FOU of an IT2 FS, one that is *symmetrical* about the primary variable x at $x = m$, $[c_l, c_r]$ can be computed by using an algorithm such as EKM or EIASC to compute c_l, after which c_r can be computed as $c_r = 2m - c_l$, leading to a 50% reduction in centroid computations.
3. If all that is desired is a crisp number after performing operations on IT2 FSs, then for the use of such sets to make a difference to not using them (e.g., to using T1 FSs or just crisp numbers) the operations that are applied to them must lead to an IT2 FS that has a non-symmetrical FOU.

Centroid TR for IT2 FSs: 6

Example 3: IMPORTANT



Centroid TR for IT2 FSs: 6



Example 3 (Continued): IMPORTANT

- The center of gravity for each of the embedded T1 FSs is (to 2 significant figures): $c_{(a)} = 4.65$, $c_{(b)} = 4.69$, $c_{(c)} = 3.99$, and $c_{(d)} = 5.37$.
- By using Theorem 8.1, it is established that $c_l = c_{(c)}$ and $c_r = c_{(d)}$, so that $C_{\tilde{A}}(x) = [3.99, 5.37]$
- Hence, this example should dispel any mistaken belief that the end-points of the centroid of an IT2 FS are associated with the centroids of its lower- and upper-MFs, $c_{(a)}$ and $c_{(b)}$.
- They are associated with embedded T1 FSs that involve segments from both the lower- and upper-MFs.

Properties About the Centroid of an IT2 FS: 1

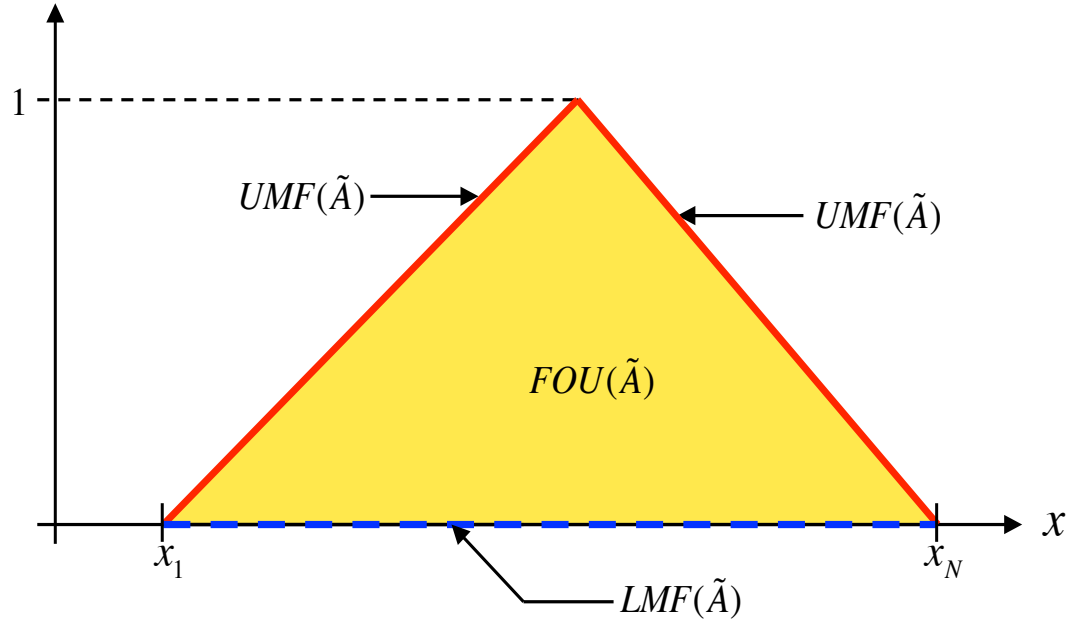
Property 8.1 $C_{\tilde{A}}(x)$ provides an *uncertainty measure* for IT2 FS $\tilde{A}$.

Property 8.2 Let $\tilde{A}$ be an IT2 FS defined on X, and $\tilde{A}'$ be $\tilde{A}$ shifted by Δm along X, i.e. $\underline{\mu}_{\tilde{A}'}(x) = \underline{\mu}_{\tilde{A}}(x - \Delta m)$ and $\overline{\mu}_{\tilde{A}'}(x) = \overline{\mu}_{\tilde{A}'}(x - \Delta m)$. Then the centroid of $\tilde{A}'$, $C_{\tilde{A}'}(x) = [c_l(\tilde{A}'), c_r(\tilde{A}')]$, is the same as the centroid of $\tilde{A}$, $C_{\tilde{A}}(x) = [c_l(\tilde{A}), c_r(\tilde{A})]$, shifted by Δm, i.e. $c_l(\tilde{A}') = c_l(\tilde{A}) + \Delta m$ and $c_r(\tilde{A}') = c_r(\tilde{A}) + \Delta m$.

Property 8.3 If the primary variable x is bounded, i.e. $x \in [x_1, x_N]$, then $c_l(\tilde{A}) \geq x_1$ and $c_r(\tilde{A}) \leq x_N$.

Properties About the Centroid of an IT2 FS: 2

Property 8.4 If $LMF(\tilde{A})$ is entirely on the primary-variable (x) axis, and $x \in [x_1, x_N]$, then the centroid does not depend upon the shape of $FOU(\tilde{A})$ and, as long as x_1 and x_N are included in the sampling points, the centroid equals $[x_1, x_N]$.



Completely filled-in FOU

Properties About the Centroid of an IT2 FS: 3

Property 8.5 When $\tilde{A} = \tilde{A}(t)$, $t = t_1, t_2, \ldots$, and $\tilde{A}(t_{k+1})$ does not change much from $\tilde{A}(t_k)$, then the EKM algorithms for computing $C_{\tilde{A}(t_{k+1})}(x)$ should be initialized by using $c_l(\tilde{A}(t_k))$ and $c_r(\tilde{A}(t_k))$. Doing this is called "EKMANI" (Enhanced KM algorithm with new initialization) and a fewer number of iterations will be required of the EKM algorithms when it is used then if it is not used.

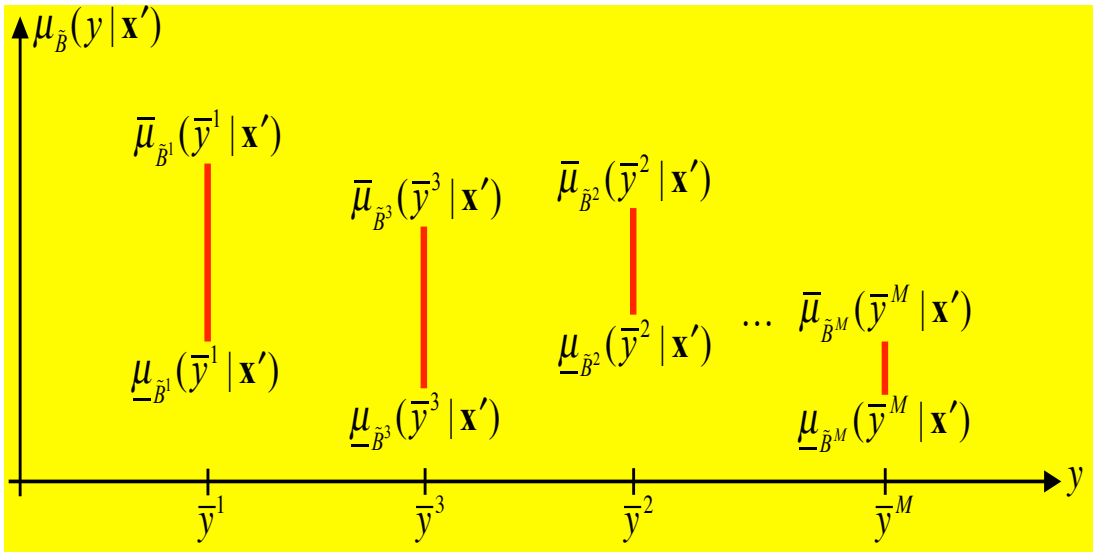
It should be adopted for all real-time IT2 fuzzy system

Centroid TR in an IT2 Fuzzy System

- Chapter 9 will demonstrate that, when an input $\mathbf{x} = \mathbf{x}'$ is applied to an IT2 rule (i.e., a rule that looks just like a type-1 rule except that some or all of its fuzzy sets are IT2 FSs) it leads to a *firing interval* $[\underline{f}^l(\mathbf{x}'), \overline{f}^l(\mathbf{x}')]$ which may then be combined with the entire consequent of that rule, $\tilde{G}^l$, by means of the meet operation, leading to an IT2 *fired-rule output FS*, $\tilde{B}^l$.
- Then, all of the IT2 fired rule output FSs may be combined by means of the join operation, producing one *combined IT2 fired rule output fuzzy set*, $\tilde{B}$. $\tilde{B}$ is then *type-reduced* by computing its centroid to give $C_{\tilde{B}}$.
- So, Theorem 8-1 applies directly to centroid type-reduction in a type-2 fuzzy system.

Height TR in an IT2 Fuzzy System: 1

- The *height type-reducer* replaces each fired rule IT2 output fuzzy set $\tilde{B}^l$ by an IT2 FS whose y-domain consists of a single point ($\overline{y}^l$), and whose secondary MF is the type-1 interval fuzzy number $\mu_{\tilde{B}^l}(\overline{y}^l \mid \mathbf{x}') = 1/[\underline{\mu}_{\tilde{B}^l}(\overline{y}^l \mid \mathbf{x}'), \overline{\mu}_{\tilde{B}^l}(\overline{y}^l \mid \mathbf{x}')]$.
- $\overline{y}^l$ can be chosen to be the point having the highest primary membership in the principal MF of the output set $\tilde{B}^l$.
- Formulas for $\underline{\mu}_{\tilde{B}^l}(\overline{y}^l \mid \mathbf{x}')$ and $\overline{\mu}_{\tilde{B}^l}(\overline{y}^l \mid \mathbf{x}')$ are given in Chapter 9.
- This procedure is summarized in the figure below.



Height TR in an IT2 Fuzzy System: 2

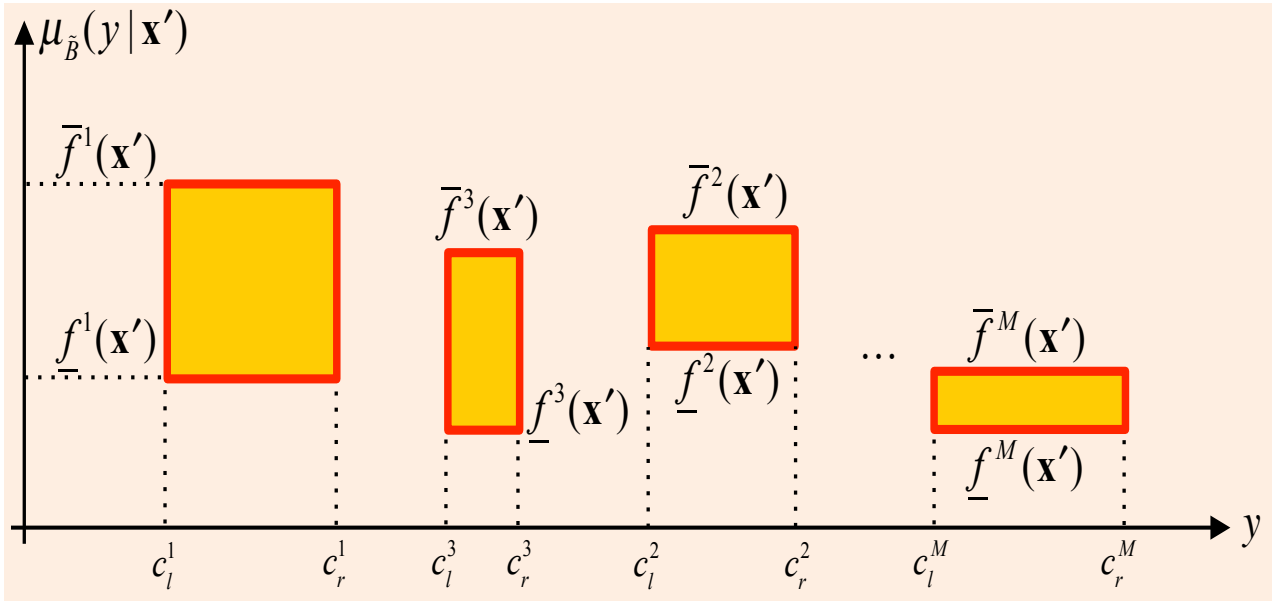
Theorem 8.2 $[y_l^h(\mathbf{x}'), y_r^h(\mathbf{x}')]$ is an IWA in which $y_l = y_l^h(\mathbf{x}')$ $y_r = y_r^h(\mathbf{x}')$, $n = M$, $a_i = b_i = \overline{y}^i$, $c_i = \underline{\mu}_{\tilde{B}^i}(\overline{y}^i \mid \mathbf{x}')$ and $d_i = \overline{\mu}_{\tilde{B}^i}(\overline{y}^i \mid \mathbf{x}')$ so that

$$y_l^h(\mathbf{x}') = \frac{\sum_{i=1}^{L} \overline{y}^i \overline{\mu}_{\tilde{B}^i}(\overline{y}^i \mid \mathbf{x}') + \sum_{i=L+1}^{M} \overline{y}^i \underline{\mu}_{\tilde{B}^i}(\overline{y}^i \mid \mathbf{x}')}{\sum_{i=1}^{L} \overline{\mu}_{\tilde{B}^i}(\overline{y}^i \mid \mathbf{x}') + \sum_{i=L+1}^{M} \underline{\mu}_{\tilde{B}^i}(\overline{y}^i \mid \mathbf{x}')}$$

$$y_r^h(\mathbf{x}') = \frac{\sum_{i=1}^{R} \overline{y}^i \underline{\mu}_{\tilde{B}^i}(\overline{y}^i \mid \mathbf{x}') + \sum_{i=R+1}^{M} \overline{y}^i \overline{\mu}_{\tilde{B}^i}(\overline{y}^i \mid \mathbf{x}')}{\sum_{i=1}^{R} \overline{\mu}_{\tilde{B}^i}(\overline{y}^i \mid \mathbf{x}') + \sum_{i=R+1}^{M} \overline{\mu}_{\tilde{B}^i}(\overline{y}^i \mid \mathbf{x}')}$$

COS TR in an IT2 Fuzzy System: 1

- The *COS type-reducer* replaces each rule consequent IT2 FS, $\tilde{G}^i$, by the domain of its centroid, $[c_l(\tilde{G}^i), c_r(\tilde{G}^i)]$, and assigns a secondary MF of $1/[\underline{f}^i(\mathbf{x}'), \overline{f}^i(\mathbf{x}')]$ to it where $[\underline{f}^i(\mathbf{x}'), \overline{f}^i(\mathbf{x}')]$ is the firing interval for the i^{th} rule.
- Formulas for $\underline{f}^i(\mathbf{x}')$ and $\overline{f}^i(\mathbf{x}')$ are given in Chapter 9.
- This procedure is summarized in the figure below.
- The shaded rectangles are information granules.



COS TR in an IT2 Fuzzy System: 2

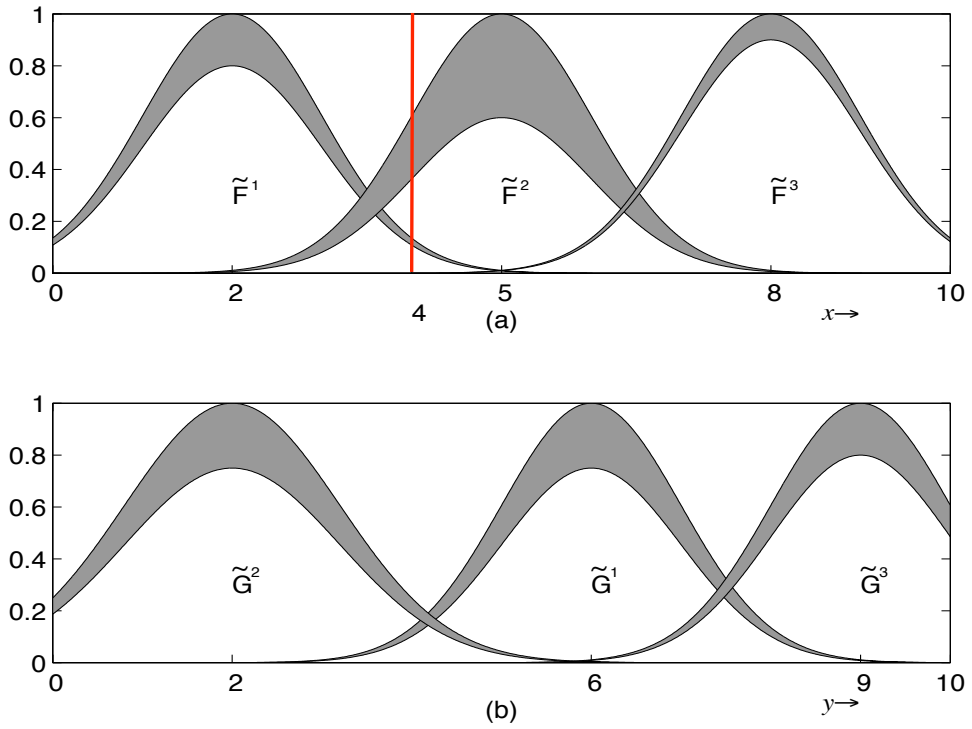
Theorem 8-3: $[y_l^{COS}(\mathbf{x}'), y_r^{COS}(\mathbf{x}')]$ is an IWA in which $y_l = y_l^{COS}(\mathbf{x}')$, $y_r = y_r^{COS}(\mathbf{x}')$, $n = M$, $a_i = c_l(\tilde{G}^i)$, $b_i = c_r(\tilde{G}^i)$, $c_i = \underline{f}^i(\mathbf{x}')$ and $d_i = \overline{f}^i(\mathbf{x}')$, so that

$$y_l^{COS}(\mathbf{x}') = \frac{\sum_{i=1}^{L} c_l(\tilde{G}^i)\overline{f}^i(\mathbf{x}') + \sum_{i=L+1}^{M} c_l(\tilde{G}^i)\underline{f}^i(\mathbf{x}')}{\sum_{i=1}^{L} \overline{f}^i(\mathbf{x}') + \sum_{i=L+1}^{M} \underline{f}^i(\mathbf{x}')}$$

$$y_r^{COS}(\mathbf{x}') = \frac{\sum_{i=1}^{R} c_r(\tilde{G}^i)\underline{f}^i(\mathbf{x}') + \sum_{i=R+1}^{M} c_r(\tilde{G}^i)\overline{f}^i(\mathbf{x}')}{\sum_{i=1}^{R} \underline{f}^i(\mathbf{x}') + \sum_{i=R+1}^{M} \overline{f}^i(\mathbf{x}')}$$

TR Example: 1

$$\tilde{R}_Z^l : \text{IF } x \text{ is } \tilde{F}^l, \text{THEN } y \text{ is } \tilde{G}^l \quad (l = 1,2,3)$$



- Antecedent and consequent FOUs.
- The applied input is $x = 4$.

TR Example: 2

Type-Reduced Set	Interval	Center	Spread
Centroid	[2.33,3.31]	2.82	0.49
Height	[2.47,3.33]	2.90	0.43
Center-of-sets	[2.58,3.32]	2.95	0.37

- There are some differences between the three type-reduced sets.
- Those differences are due mainly to the truncated natures of $\tilde{F}^1$ and $\tilde{G}^2$.

Metrics Used to Evaluate TR Algorithms: 1

- A meaningful metric is *computation time per iteration*, because number of iterations is independent of computer speed.
- This number will, of course, become smaller and smaller as computers become faster and faster, something that always seems to occur.
- When the type-reduction algorithms are implemented in *hardware*, then the faster they can be performed frees up the hardware to perform other computations, something that can be very important for real-world applications.
- Consequently, it is important to perform the type-reduction calculations as quickly as possible, which is why there has been extensive work on improving type-reduction algorithms.
- And so, evaluating new type-reduction algorithms in terms of computing time/iteration is a meaningful metric, even when this is done outside of the context of an application of a fuzzy system.

Metrics Used to Evaluate TR Algorithms: 2

- Studies that focus on "accuracy" as a metric for a centroid algorithm, and that evaluate accuracy by using the "exact" centroid, where the "exact" centroid is the infinite precision result, are of limited value, because if centroid type-reduction is used in a type-2 fuzzy system, it is not this kind of "accuracy" that is important.
- Instead, it is achieving acceptable application-related performance metrics that is important.

TR for GT2 FSs and Fuzzy Systems

- The approach to type-reduction that is taken in this book uses the horizontal-slice representation for a GT2 FS, namely:

$$\tilde{A} = \bigcup_{\alpha \in [0,1]} \alpha / \tilde{A}_\alpha = \sup_{\alpha \in [0,1]} \alpha / \tilde{A}_\alpha$$

where

$$\tilde{A}_\alpha = \int_{x \in X} \tilde{A}(x)_\alpha / x = \int_{x \in X} [a_\alpha(x), b_\alpha(x)] / x$$

- Regardless of the kind of type-reduction (centroid, height or center-of-sets), the following is always true: **type-reduction for a GT2 FS (or for more than one GT2 FS) can be viewed as a non-linear function of the primary variable (or variables) of the set (or sets), and so it can be computed as the fuzzy union of that non-linear function applied to α - *planes*.**

Centroid TR for GT2 FSs: 1

Theorem 8.4 The centroid of a closed GT2 FS $\tilde{A}$, $C_{\tilde{A}}(x)$, is a type-1 fuzzy set that can be computed using the horizontal-slice representation of $\tilde{A}$, as:

$$C_{\tilde{A}}(x) = \bigcup_{\alpha \in [0,1]} C_{R_{\tilde{A}_\alpha}}(x) = \bigcup_{\alpha \in [0,1]} \alpha / [c_l(R_{\tilde{A}_\alpha}), c_r(R_{\tilde{A}_\alpha})]$$

$$\equiv \bigcup_{\alpha \in [0,1]} \alpha / [c_l(\alpha), c_r(\alpha)]$$

where $C_{R_{\tilde{A}_\alpha}}(x)$ is the centroid of the horizontal slice at level α, $R_{\tilde{A}_\alpha}$.

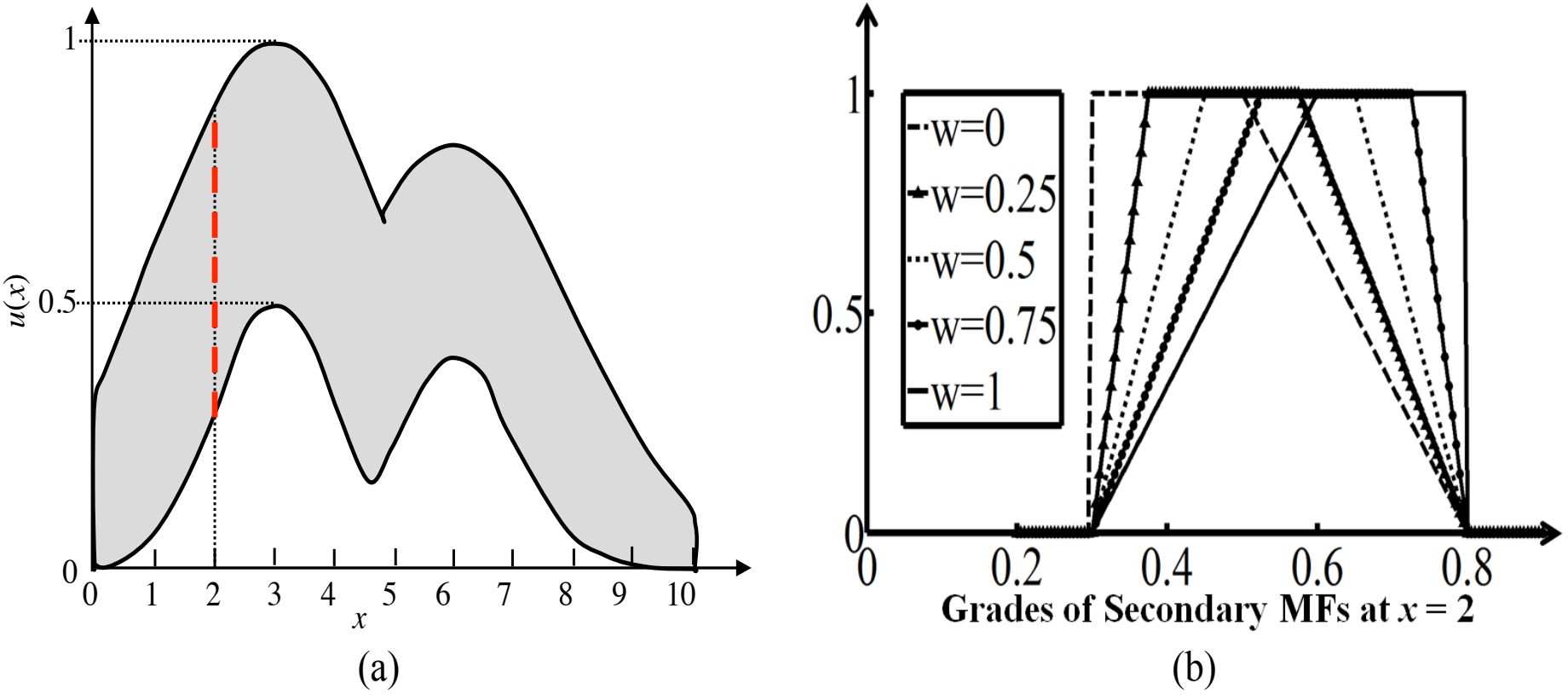
Centroid TR for GT2 FSs: 2

A procedure for computing $C_{\tilde{A}}(x)$ is:

1. Decide on how many α-planes will be used, where $\alpha \in [0,1]$. Call that number $k_{\max}$; its choice will depend on the accuracy that is required, if accuracy is important, or its choice will be made a design parameter during the design of a general type-2 fuzzy system.
2. For each α, compute $\tilde{A}_\alpha$.
3. Compute $c_l(\alpha)$ and $c_r(\alpha)$ using two EKM algorithms or EIASC algorithms, or even better by using the monotone centroid flow algorithms that are explained soon.
4. Repeat Steps 2 and 3 for the $k_{\max}$ values of α chosen in Step 1.
5. Bring all of the $k_{\max}$ $C_{R_{\tilde{A}_\alpha}}(x)$ together using the fuzzy union to obtain $C_{\tilde{A}}(x)$.

Centroid TR for GT2 FSs: 3

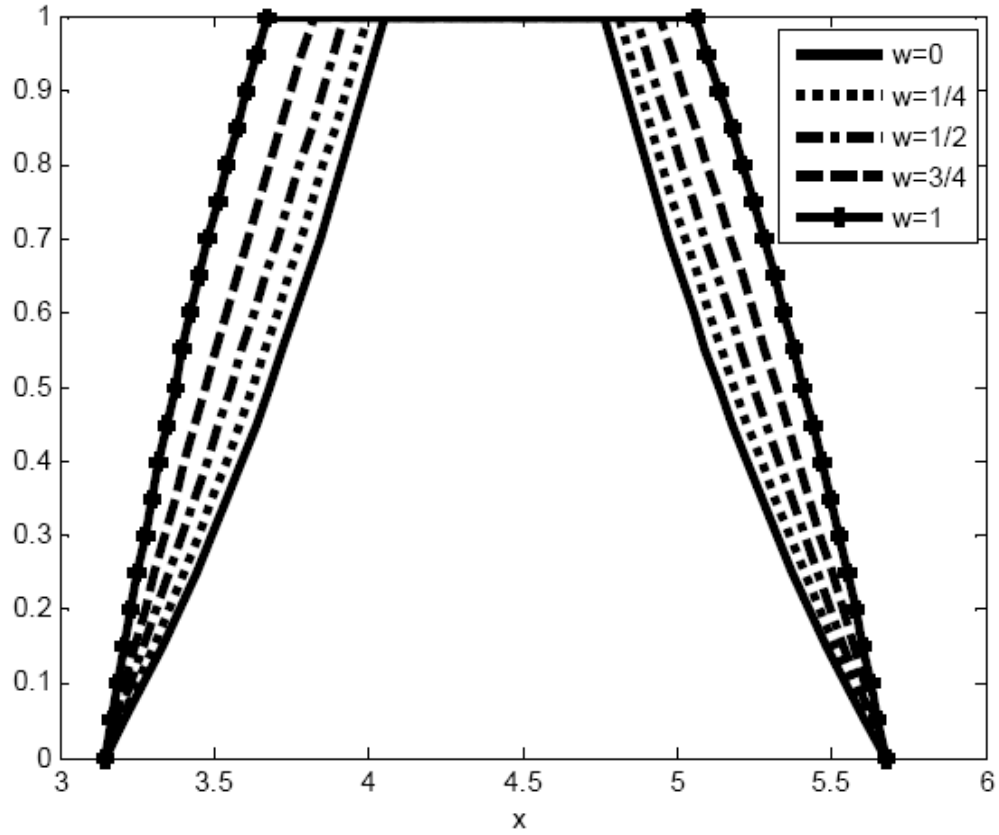
EXAMPLE 1: Gaussian LMF and UMF and trapezoidal secondary MFs



(a) $FOU(\tilde{G})$, and (b) Secondary MFs at $x = 2$ for $w = 0,\ 0.25,\ 0.5,\ 0.75$ and 1.

Centroid TR for GT2 FSs: 4

EXAMPLE 1 (Continued): Gaussian LMF and UMF and trapezoidal secondary MFs



$C_{\tilde{G}}(x)$ when $w = 0,\ 0.25,\ 0.5,\ 0.75$ and 1.

Centroid TR for GT2 FSs: 5

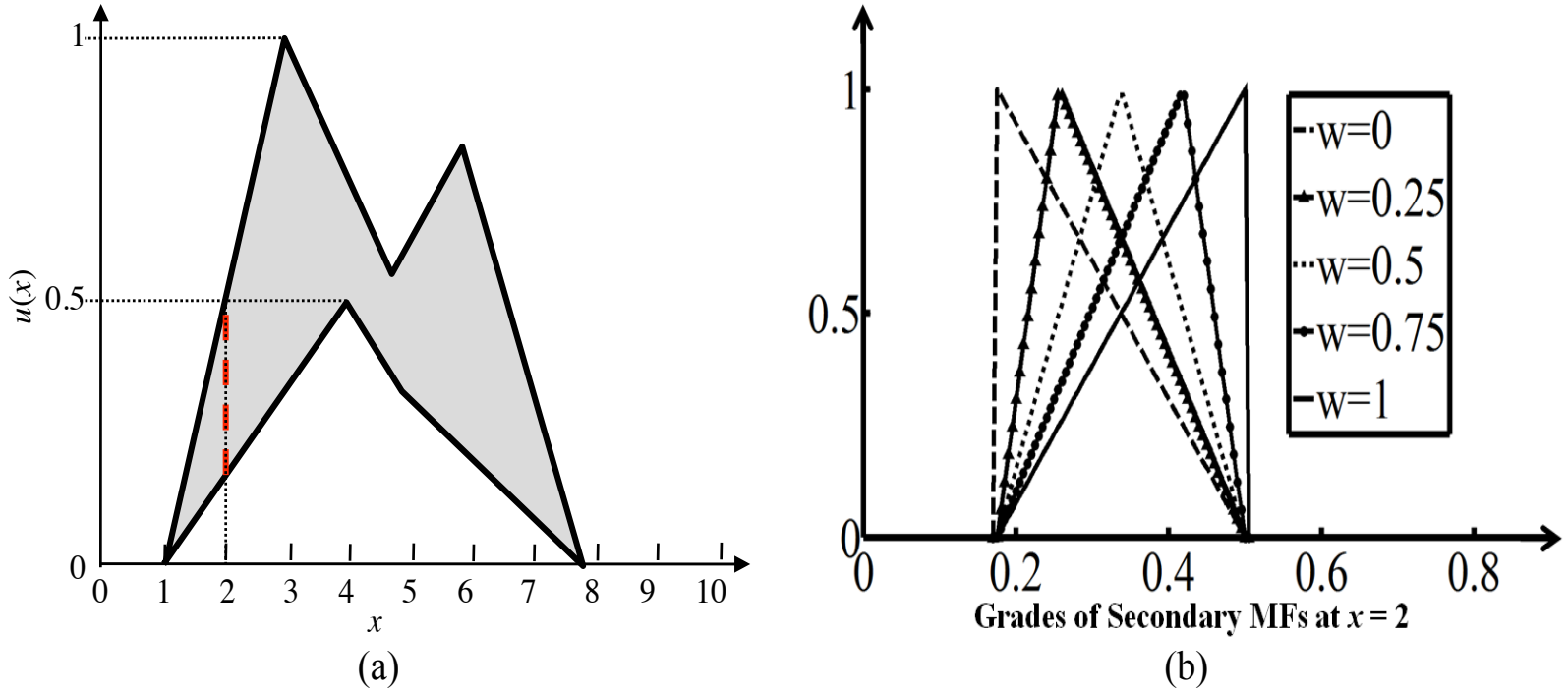
EXAMPLE 1 (Continued): Gaussian LMF and UMF and trapezoidal secondary MFs

w	$m(C_{\tilde{G}}(x))$	$m(C_{\tilde{G}_{\alpha=0}}(x)) - m(C_{\tilde{G}}(x))$	Left-end of $C_{\tilde{G}_{\alpha=1}}(x)$	Right-end of $C_{\tilde{G}_{\alpha=1}}(x)$	$m(\hat{C}_{\tilde{G}}(x))$	$m(\hat{C}_{\tilde{G}}(x)) - m(C_{\tilde{G}}(x))$
0	4.41	0.01 (0.23%)	4.06	4.77	4.41	0 (0%)
0.25	4.40	0.02 (0.46%)	4.00	4.81	4.41	0.01 (0.23%)
0.50	4.40	0.02 (0.46%)	3.93	4.87	4.41	0.01 (0.23%)
0.75	4.39	0.03 (0.69%)	3.83	4.95	4.40	0.01 (0.23%)
1	4.38	0.04 (0.92%)	3.68	5.06	4.39	0.01 (0.23%)

A very good approximation to the centroid can be obtained by computing centroids for only two α-planes — $\alpha = 0$ and $\alpha = 1$.

Centroid TR for GT2 FSs: 6

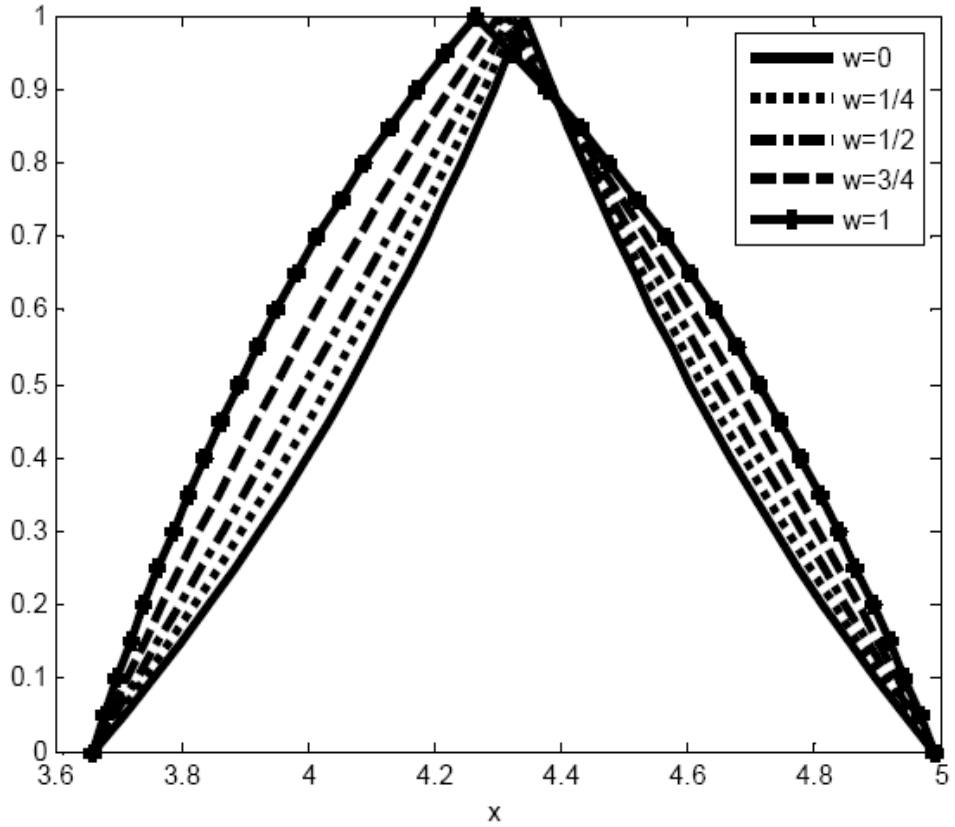
EXAMPLE 2: Piecewise Linear LMF and UMF and triangle secondary MFs



(a) $FOU(\tilde{F})$ and (b) Secondary MFs at $x = 2$ for $w = 0,\ 0.25,\ 0.5,\ 0.75$ and 1.

Centroid TR for GT2 FSs: 7

EXAMPLE 2 (Continued): Piecewise Linear LMF and UMF and triangle secondary MFs



$C_{\tilde{F}}(x)$ when $w = 0,\ 0.25,\ 0.5,\ 0.75$ and 1.

Centroid TR for GT2 FSs: 7

EXAMPLE 2 (Continued): Piecewise Linear LMF and UMF and triangle secondary MFs

w	$m(C_{\tilde{F}}(x))$	$m(C_{\tilde{F}_{\alpha=0}}(x)) - m(C_{\tilde{F}}(x))$	$C_{\tilde{F}_{\alpha=1}}(x)$	$m(\hat{C}_{\tilde{F}}(x))$	$m(\hat{C}_{\tilde{F}}(x)) - m(C_{\tilde{F}}(x))$
0	4.34	-0.01 (-0.23%)	4.34	4.33	-0.01 (-0.23%)
0.25	4.33	0.00 (0%)	4.33	4.33	0 (0%)
0.50	4.32	0.01 (0.23%)	4.32	4.32	0 (0 %)
0.75	4.31	0.02 (0.46%)	4.30	4.32	0.01 (0.23%)
1	4.29	0.04 (0.96%)	4.27	4.31	0.02 (0.47%)

This example again demonstrates that a very good approximation to the centroid can be obtained by computing centroids for only two α-planes — $\alpha = 0$ and $\alpha = 1$.

Centroid TR for GT2 FSs: 8

- The five-step procedure stated on Slide # 40 does not make use of the additional information that is available about the nature of the secondary MFs of a GT2 FS.
- A number of centroid flow algorithms have been developed that make use of this information.
- In them computations flow from one alpha level to the next
- They are very complicated even though they achieve a 70% reduction in computation time over the EKM algorithms

Centroid TR for GT2 FSs: 9

- Linda and Manic developed *monotone centroid flow* (*MCF*) *algorithms* that may not require any EIASC or EKM algorithms, and are to-date the fastest way to compute the centroid of a GT2 FS.
- The MCF algorithms start with the horizontal slice at level $\alpha = 1$ so as to use an exact COG calculation when all secondary MFs are either triangles (no EIASC or EKM algorithms are needed to do this) or trapezoids (EIASC or EKM algorithms are needed to do this, but only one time).
- They then move down to the horizontal slice at level $\alpha = 1 - \delta$, but do not use the EKM algorithms at that plane.
- Instead, they return to *fundamentals*, by focusing on the structure of the solutions for $c_l(\tilde{A})$ and $c_r(\tilde{A})$ that are given in Theorem 8.1, but for the horizontal slice at level $\alpha = 1 - \delta$.
- See pages 429-430 of the textbook for the details.

Centroid TR for GT2 FSs: 10

EXAMPLE

- Linda and Manic have many very interesting computational results in their paper.
- All are for the GT2 FSs $\tilde{G}$ and $\tilde{F}$ that have been described above, respectively, for $w = \{0.1, 0.9\}$.
- They concluded: "… the MCF and KM algorithms compute numerically identical solutions."
- They then studied *computation times* for many values of n (samples, ranging from 20 to 215) and $k_{\max}$ (number of horizontal slices, ranging from 2 to 100), and for $w = \{0.0.25.0.5, 0.75, 1\}$.
- Computation times were averaged over all n, $k_{\max}$ and w.
- The overall average speed-up of MCF over KM was approximately 85% for both $\tilde{G}$ and $\tilde{F}$, whereas it was approximately 55% to 58% for EKM over KM.

Centroid TR in a GT2 Fuzzy System

- When an input $\mathbf{x} = \mathbf{x}'$ is applied to a general type-2 rule (i.e., a rule that looks just like a type-1 rule except that some or all of its fuzzy sets are GT2 FSs) it leads to a type-1 fuzzy *firing set* $F(\mathbf{x}')$ which may then be combined with the entire GT2 consequent of that rule, $\tilde{G}^l$, by means of the meet operation, leading to a *fired-rule output GT2 FS*, $\tilde{B}^l$.

- Then, all of the fired rule output GT2 FSs may be combined by means of the join operation, producing one *combined fired rule GT2 output fuzzy set*, $\tilde{B}$. $\tilde{B}$ is then *type-reduced* by computing its centroid to give $Y_c(\mathbf{x}')$.

- Theorem 8.4 applies directly to centroid type-reduction in a GT2 fuzzy system.

COS TR in a GT2 Fuzzy System

- Center-of-sets (COS) type-reduction is explained in Chapter 11.
- It uses the horizontal-slice representation for a GT2 FS.
- Height type-reduction is left as an exercise, because as of 2017 COS type-reduction is being used in applications whereas height type-reduction is not.

A Wavy Slice Approach to Centroid TR: 1

$$C_{\tilde{B}}(y) = \int_{\theta_1 \in I_{y_1}} \cdots \int_{\theta_N \in I_{y_N}} \left[\min_{i=1,\ldots,N} \{ f_{y_i}(\theta_i) \} \right] \Bigg/ \frac{\sum_{i=1}^{N} y_i \theta_i}{\sum_{i=1}^{N} \theta_i}$$

1. Discretize Y into N points $y_1, \ldots, y_N$.
2. Discretize each I_{y_i} (the support of the secondary MF at y_i) into a suitable number of points, say M_i $(i = 1, \ldots, N)$. Let $\theta_i \in I_{y_i}$.
3. Enumerate all the embedded T1 FSs of $\tilde{B}$; there will be $\prod_{i=1}^{N} M_i$ of them.
4. Compute the centroid of each enumerated embedded T1 FS and assign it a membership grade equal to the minimum of the secondary grades corresponding to that enumerated embedded T1 FS.

A Wavy Slice Approach to Centroid TR: 2

$$C_{\tilde{B}}(y) = \left\{ \left[\zeta_k, \left(\min_{i=1,\ldots,N} \{ f_{y_i}(\theta_i) \} \right)_k \right] \right\}_{k=1}^{\prod_{i=1}^{N} M_i}$$

$$\zeta_k = \left(\frac{\sum_{i=1}^{N} y_i \theta_i}{\sum_{i=1}^{N} \theta_i} \right)_{k^{th} \text{ embedded T1 FS}}$$

This four-step procedure is not very practical because it requires the explicit enumeration of the $\prod_{i=1}^{N} M_i$ embedded T1 FSs of $\tilde{B}$, so that their centroids can be computed, and this number will in general be extremely large.

A Wavy Slice Approach to COS TR: 1

$$GC = \int_{z_1 \in Z_1} \cdots \int_{z_n \in Z_n} \int_{w_1 \in W_1} \cdots \int_{w_n \in W_n} \left[T_{i=1}^{n} \mu_{Z_i}(z_i) \bigstar T_{i=1}^{n} \mu_{W_i}(w_i) \right] \Big/ \frac{\sum_{i=1}^{n} z_i w_i}{\sum_{i=1}^{n} w_i}$$

1. Discretize the domain of each T1 FS Z_i into a suitable number of points, say $N_i \quad (i = 1,...,n)$.

2. Discretize the domain of each T1 FS W_i into a suitable number of points, say $M_i \quad (i = 1,...,n)$.

3. Enumerate all the possible combinations $\boldsymbol{\theta} = [z_1,...,z_n, w_1,...,w_n]^T$ such that $z_i \in Z_i$ and $w_i \in W_i$. The total number of combinations will be $\prod_{i=1}^{n} M_i N_i$.

4. Compute the centroid $\sum_{i=1}^{n} z_i w_i \Big/ \sum_{i=1}^{n} w_i$ of each of the enumerated combinations and assign it a membership grade equal to the t-norm $T_{i=1}^{n} \mu_{Z_i}(z_i) \bigstar T_{i=1}^{n} \mu_{W_i}(w_i)$.

A Wavy Slice Approach to COS TR: 2

$$GC = \left\{ \left[\xi_k, \left(T_{i=1}^{n} \mu_{Z_i}(z_i) \bigstar T_{i=1}^{n} \mu_{W_i}(w_i) \right)_k \right] \right\}_{k=1}^{\prod_{i=1}^{n} M_i N_i}$$

$$\xi_k = \left(\frac{\sum_{i=1}^{n} z_i w_i}{\sum_{i=1}^{n} w_i} \right)_{k^{th} \text{ combination}}$$

This four-step procedure is not very practical because it requires the explicit enumeration of $\prod_{i=1}^{n} M_i N_i$ combinations of $[z_1,...,z_n,w_1,...,w_n]^T$ so that their weighted average can be computed, and this number is also in general extremely large.

Properties of the IWA: 1

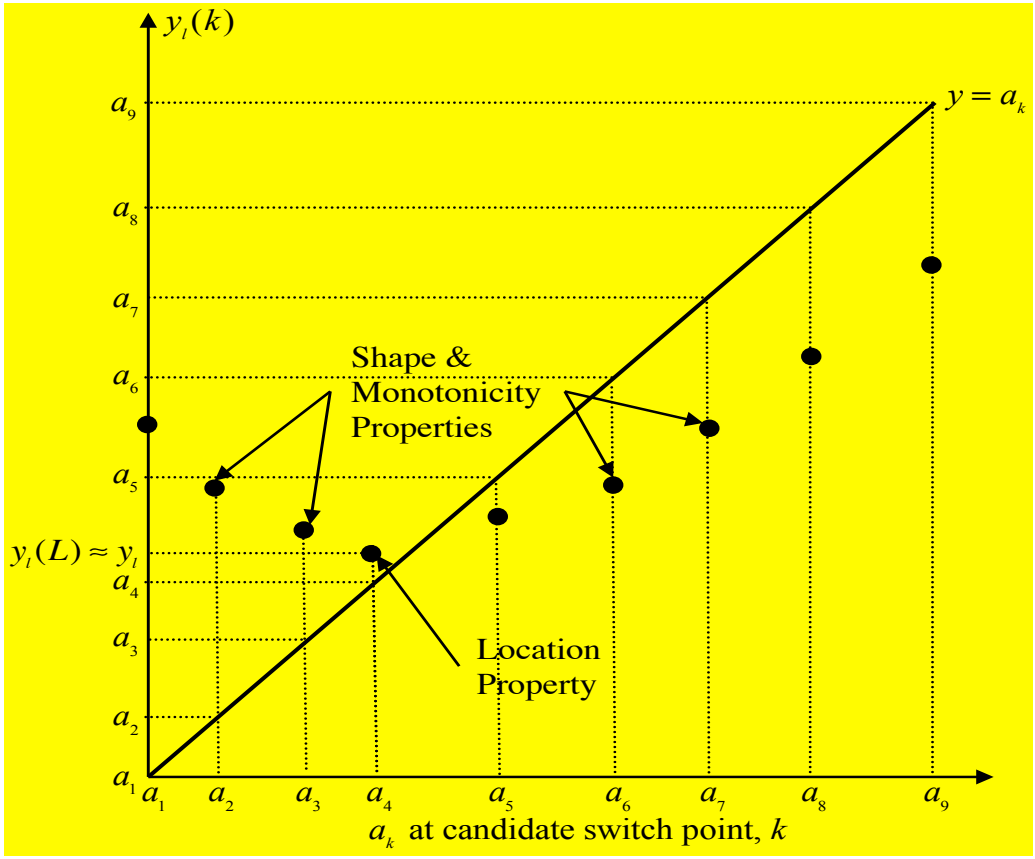
Property 8.6 [Location Property for $y_l(L)$] When $k = L$, then

$$a_L \leq y_l(L) = y_l < a_{L+1}$$

If $a_{L+1} = a_L$, then change the right-hand-side of this inequality to: "the first value of *i* such that $a_i \neq a_L$".

- This property locates $y_l(L)$ either between two specific adjacent values of a_i, or at the left end-point of these adjacent values

Properties of the IWA: 2



- Illustration of the three properties associated with finding $y_l(L)$. The solid line shown for $y = a_k$ only has values at $a_1,...,a_9$; and the large dots are $y_l(k)$ in (8-17) for $k = 1,...,9$. The 45^0 line $y = a_k$ is shown because of the computations in Step 2 of the KM (or EKM) algorithm for y_l .

Properties of the IWA: 3

Property 8.7 [Shape Property for $y_l(k)$; location of $y_l(k)$ in relation to the line $y = a_k$] $y_l(k)$ lies above the line $y = a_k$ when a_k is below y_l and lies below the line $y = a_k$ when a_k is above y_l , i.e.

$$\begin{cases} y_l(k) > a_k & \text{when } a_k < y_l \\ y_l(k) < a_k & \text{when } a_k > y_l \end{cases}$$

- This property explains the shape of $y_l(k)$ both to the left and right of its minimum point (see the large dots on the figure on Slide # 57)

Properties of the IWA: 4

Property 8.8 [Monotonicity Property of $y_l(k)$] It is true that:

$$\begin{cases} y_l(k-1) \geq y_l(k) & \text{when} \quad a_k < y_l \\ y_l(k+1) \geq y_l(k) & \text{when} \quad a_k > y_l \end{cases}$$

- This property also helps us to understand the shape of $y_l(k)$. When $y_l(k)$ is going in the downward direction (see the figure on Slide #57) it cannot change that direction before $a_k = y_l$; and, after $a_k = y_l$, when it goes in the upward direction it cannot change that direction.
- Similar properties for y_r are given in the book, as Properties 8.9, 8.10 and 8.11.

Continuous KM Algorithms for the Centroid of an IT2 FS: 1

- Some interesting properties have been developed about the centroid of an IT2 FS $\tilde{A}$ and the convergence rates of the KM algorithms for the situation when one begins with mathematical formulas for the lower and upper MFs, so that the universe of discourse for $\tilde{A}$ can be treated as continuous.

- An alternative interpretation is to let the discretization size (sampling size) approach 0.

Continuous KM Algorithms for the Centroid of an IT2 FS: 2

Continuous KM (CKM) algorithms to compute the centroid end points of an IT2 FS, $\tilde{A}$

Step	CKM algorithm for c_l $c_l = \min_{\forall\theta(x)\in[\underline{\mu}_{\tilde{A}}(x),\overline{\mu}_{\tilde{A}}(x)]} \frac{\int_a^b x\theta(x)dx}{\int_a^b \theta(x)dx}$	CKM algorithm for $c_r = \max_{\forall\theta(x)\in[\underline{\mu}_{\tilde{A}}(x),\overline{\mu}_{\tilde{A}}(x)]} \frac{\int_a^b x\theta(x)dx}{\int_a^b \theta(x)dx}$
1	Let $\theta(x) = [\underline{\mu}_{\tilde{A}}(x) + \overline{\mu}_{\tilde{A}}(x)]/2$ and compute the initial value ξ, as $\xi = \int_a^b x\theta(x)dx \Big/ \int_a^b \theta(x)dx$	
2	Compute $\xi_l = \frac{\int_a^{\xi} x\overline{\mu}_{\tilde{A}}(x)dx + \int_{\xi}^b x\underline{\mu}_{\tilde{A}}(x)dx}{\int_a^{\xi} \overline{\mu}_{\tilde{A}}(x)dx + \int_{\xi}^b \underline{\mu}_{\tilde{A}}(x)dx}$	Compute $\xi_r = \frac{\int_a^{\xi} x\underline{\mu}_{\tilde{A}}(x)dx + \int_{\xi}^b x\overline{\mu}_{\tilde{A}}(x)dx}{\int_a^{\xi} \underline{\mu}_{\tilde{A}}(x)dx + \int_{\xi}^b \overline{\mu}_{\tilde{A}}(x)dx}$
3	Check if $\lvert\xi - \xi_l\rvert \le \varepsilon$ (ε is a given error bound of the algorithm). If yes, stop and set $c_l = \xi_l$. If no, go to Step 4.	Check if $\lvert\xi - \xi_r\rvert \le \varepsilon$ (ε is a given error bound of the algorithm). If yes, stop and set $c_r = \xi_r$. If no, go to Step 4.
4	Set $\xi = \xi_l$ and go to Step 2.	Set $\xi = \xi_r$ and go to Step 2.

Continuous KM Algorithms for the Centroid of an IT2 FS: 3

Property 8.12 The continuous versions of (8.17) and (8.19), for finding the centroid of an IT2 FS, are:

$$c_l = \min_{\xi \in [a,b]} c_l(\xi) = \min_{\xi \in [a,b]} \frac{\int_a^{\xi} x \overline{\mu}_{\tilde{A}}(x)\,dx + \int_{\xi}^{b} x \underline{\mu}_{\tilde{A}}(x)\,dx}{\int_a^{\xi} \overline{\mu}_{\tilde{A}}(x)\,dx + \int_{\xi}^{b} \underline{\mu}_{\tilde{A}}(x)\,dx}$$

$$c_r = \max_{\xi \in [a,b]} c_r(\xi) = \max_{\xi \in [a,b]} \frac{\int_a^{\xi} x \underline{\mu}_{\tilde{A}}(x)\,dx + \int_{\xi}^{b} x \overline{\mu}_{\tilde{A}}(x)\,dx}{\int_a^{\xi} \underline{\mu}_{\tilde{A}}(x)\,dx + \int_{\xi}^{b} \overline{\mu}_{\tilde{A}}(x)\,dx}$$

Continuous KM Algorithms for the Centroid of an IT2 FS: 4

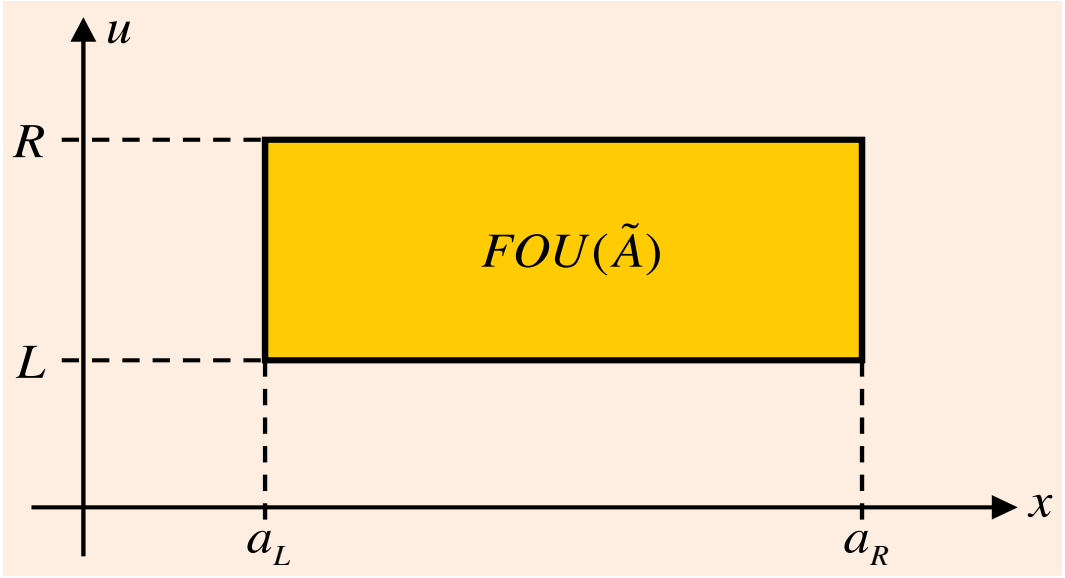
Property 8.13 c_l and c_r are the solutions to the following integral equations:

$$c_l = \frac{\int_a^{c_l} x\overline{\mu}_{\tilde{A}}(x)dx + \int_{c_l}^{b} x\underline{\mu}_{\tilde{A}}(x)dx}{\int_a^{c_l} \overline{\mu}_{\tilde{A}}(x)dx + \int_{c_l}^{b} \underline{\mu}_{\tilde{A}}(x)dx}$$

$$c_r = \frac{\int_a^{c_r} x\underline{\mu}_{\tilde{A}}(x)dx + \int_{c_r}^{b} x\overline{\mu}_{\tilde{A}}(x)dx}{\int_a^{c_r} \underline{\mu}_{\tilde{A}}(x)dx + \int_{c_r}^{b} \overline{\mu}_{\tilde{A}}(x)dx}$$

Continuous KM Algorithms for the Centroid of an IT2 FS: 5

EXAMPLE



$$c_l(\tilde{A}) = \frac{\sqrt{L}a_R + \sqrt{R}a_L}{\sqrt{R} + \sqrt{L}}$$

$$c_r(\tilde{A}) = \frac{\sqrt{R}a_R + \sqrt{L}a_L}{\sqrt{R} + \sqrt{L}}$$

These closed-form formulas were possible because $\underline{\mu}_{\tilde{A}}(x)$ and $\overline{\mu}_{\tilde{A}}(x)$ are constants, so that the equations for c_l and c_r are quadratic.

Continuous KM Algorithms for the Centroid of an IT2 FS: 6

Property 8.14: The optimal solutions of c_l and c_r (Slide #62) can be transformed into root-finding problems, i.e.: (a) $c_l = c_l(\xi^*)$ where ξ^* is the unique simple root of the monotonic increasing convex function $\varphi(\xi)$,

$$\varphi(\xi) = \int_a^{\xi} (\xi - x)\overline{\mu}_{\tilde{A}}(x)dx + \int_{\xi}^{b} (\xi - x)\underline{\mu}_{\tilde{A}}(x)dx$$

(b) $c_r = c_r(\xi^*)$ where ξ^* is the unique simple root of the monotonic decreasing convex function $\psi(\xi)$,

$$\psi(\xi) = -\int_a^{\xi} (\xi - x)\underline{\mu}_{\tilde{A}}(x)dx - \int_{\xi}^{b} (\xi - x)\overline{\mu}_{\tilde{A}}(x)dx$$

- As a result of this property, one can apply methods for solving nonlinear root-finding problems to $\varphi(\xi)$ and $\psi(\xi)$ e.g., the Newton-Raphson algorithm

Continuous KM Algorithms for the Centroid of an IT2 FS: 7

Property 8.15: The iteration processes in the CKM algorithms in the Table on Slide #61 are equivalent to the Newton-Raphson root-finding method for solving $\varphi(\xi)=0$ and $\psi(\xi)=0$ with the iteration formulae:

$$\xi_{k+1}=\xi_k-\frac{\varphi(\xi_k)}{[\partial\varphi(\xi)/\partial\xi]_{\xi=\xi_k}}$$

$$\xi_{k+1}=\xi_k-\frac{\psi(\xi_k)}{[\partial\psi(\xi)/\partial\xi]_{\xi=\xi_k}}$$

Property 8.16: convergence speed of the CKM algorithms is quadratic.

Property 8.17 CKM algorithms exhibit global convergence, i.e. for any initialization method the CKM algorithms will always converge to the sample optimal solutions c_l and c_r.

Continuous KM Algorithms for the Centroid of an IT2 FS: 8

- The results for the CKM algorithm only apply to the situation where an FOU exists, i.e. for centroid TR
- They do not apply to height or COS TR