

Fuzzy-rendszerek



Fuzzy rendszerekről általában



Fuzzy-rendszerek

Témák

1. Fuzzy rendszerekről általában
2. Fuzzy halmazelmélet
 - Fuzzy halmazműveletek
3. Fuzzy logika
 - Fuzzy logika műveletek
4. Fuzzy közelítő következtetés



1.Fuzzy rendszerekről általában

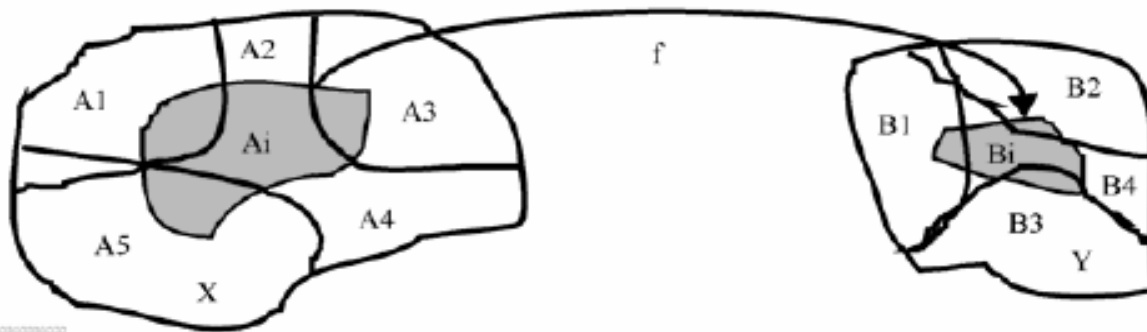
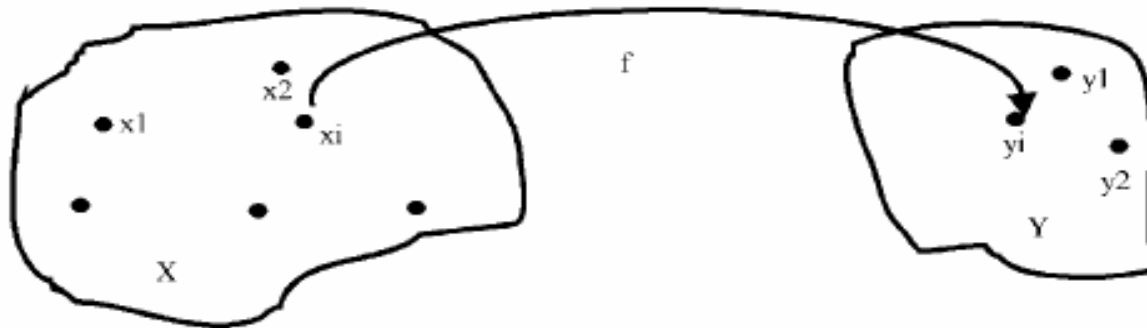
- Fuzzy információ
 - nem precíz, pontatlan kifejezések (idős ember, magas nyereség)
 - bizonytalan információkon alapuló kif. (hitelképes vállalat)
 - éleetlen relációk (körülbelül egyenlő)
- Előfordulás
 - mat. modellek, döntések, adatok analízise



1.Fuzzy rendszerekről általában

- A bizonytalanság jellege:
 - determinisztikus bizonytalanság (többértékű)
- Fuzzy technika (Zadeh 65)
 - „hozzátartozás foka”: elem halmazhoz tartozása
 - ~Igazságfok (predikátummal)
 - A technika alapja:
 - fuzzy halmazelmélet, fuzzy logika
 - fuzzy rendszer: halmaz- halmaz leképezés
 $A_i \rightarrow B_i$ formájú leképezéseket aggregál

1. Fuzzy rendszerekről általában



Neurális háló és fuzzy rendszer leképezés

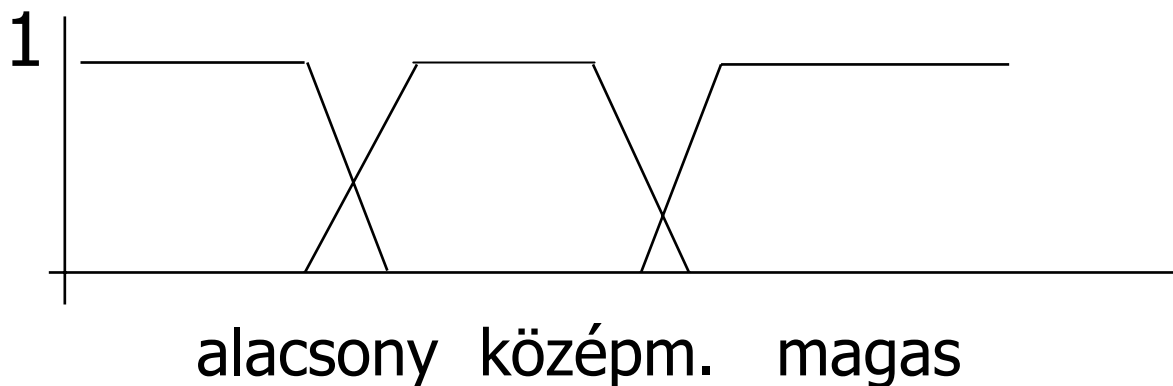
2.Fuzzy halmazelmélet

Fuzzy halmazelmélet

- klasszikus halmazelmélet $A = \{x \mid x > 120\}$
 - Tagsági/ karakterisztikus függvénye

$$\mu(x) = \begin{cases} 1 & \text{ha } x > 120 \\ 0 & \text{különben} \end{cases}$$

- élelten, fuzzy halmazok: hozzátartozás foka



2. Fuzzy halmazelmélet

- Tagsági / tartalmazási függvény:

$$\mu : X \rightarrow [0,1]$$

- Megadási formák:

$$A = \{ x, \mu_A(x) \mid x \in X \}$$

$$A = \mu_A(x_1)/x_1 + \mu_A(x_2)/x_2 + \dots + \mu_A(x_n)/x_n = \sum \mu_A(x_i)/x_i$$

$$A = \int_x \mu_A(x)/x .$$



2. Fuzzy halmazelmélet

Műveletek

Legyen A, B fuzzy halmaz, $\mu_A(x), \mu_B(x)$ tart. függv.

- Aritmetikai műveletek:

$$C = A \circ B \quad (\circ \text{ lehet } + - / *)$$

$$\mu_C(z) = \mu_{A \circ B}(z) = \sup_{z = x \circ y} \min(\mu_A(x), \mu_B(y)).$$

- Halmazműveletek:

- metszet (minimum operátor)

$$\mu_{A \wedge B}(x) = \min(\mu_A(x), \mu_B(x))$$



2. Fuzzy halmazelmélet

Műveletek

Legyen A, B fuzzy halmaz, $\mu_A(x), \mu_B(x)$ tart. függv.

- Halmazműv.:

- egyesítés (maximum operátor)

$$\mu_{A \vee B}(x) = \max(\mu_A(x), \mu_B(x))$$

- szorzat

$$\mu_{A \wedge B}(x) = \mu_A(x) * \mu_B(x)$$

- Komplement

$$\mu_{\neg A}(x) = 1 - \mu_A(x)$$

- T-norma S-norma művelettel definiálás

Metszet: $\mu_{A \wedge B}(x) = T(\mu_A(x), \mu_B(x))$



2.Fuzzy halmazelmélet

T-norma S-norma műveletek

Egy $T: [0,1]^2 \rightarrow [0,1]$ függvényt *T-normának* nevezünk, ha teljesül:

1. $T(a, 1) = a$ (1 az egységelem)
2. $a \leq b$ és $c \leq d \Rightarrow T(a,c) \leq T(b,c)$ (monoton növény)
3. $T(a,b) = T(b,a)$ (kommutatív)
4. $T(a, T(b,c)) = T(T(a,b),c)$ (asszociatív)

Egy $S: [0,1]^2 \rightarrow [0,1]$ függvényt *T-konormának*, vagy *S-normának* nevezünk, ha

1. $S(a, 0) = a$ (0 az egységelem)
2. $a \leq b$ és $c \leq d \Rightarrow S(a,c) \leq S(b,c)$ (monoton növény)
3. $S(a,b) = S(b,a)$ (kommutatív)
4. $S(a, S(b,c)) = S(S(a,b),c)$ (asszociatív).

2. Fuzzy halmazelmélet

- **Fuzzy reláció** (kartézi szorzat részhalmaza)

$$R = \{((x_1, x_2, \dots, x_n), \mu_R(x_1, x_2, \dots, x_n)) \mid (x_1, x_2, \dots, x_n) \in X_1 \times X_2 \times \dots \times X_n\}$$
$$\mu_R : X_1, X_2, \dots, X_n \rightarrow [0, 1]$$

- Példa:

$$X_1 = \{a, b, c, d, e, f\},$$

$$X_2 = \{1, 2, 3, 4, 5\},$$

$$A_1 = \{0.1/a, 0.4/b, 0.9/d\},$$

$$A_2 = \{0.9/1, 0.7/2, 0.3/3\}.$$

μ_R	1	2	3
a	0.1	0.1	0.1
b	0.4	0.4	0.3
d	0.9	0.7	0.3



2.Fuzzy halmazelmélet

■ Reláció műveletek

- projekció,
- cilindrikus kiterjesztés
- max-min kompozíció

R1 és R2 max-min kompozíciója X,Y felett egy R fuzzy reláció:

$$R = R1 \circ R2 = \{(x,z), \sup_y \min(\mu_{R1}(x,y), \mu_{R2}(y,z)) \mid x \in X, y \in Y, z \in Z\}.$$

2. Fuzzy halmazelmélet

- Kiterjesztési elv

Az $f: X_1 \times X_2 \times \dots \times X_n \rightarrow Y$ leképezés egy

$$f^*: A_1 \times A_2 \times \dots \times A_n \rightarrow B$$

kiterjesztése egy fuzzy halmaz Y felett:

$$B = f^*(A_1, A_2, \dots, A_n) = \int \mu_B(y) / y \quad \text{ahol}$$

$$\mu_B(y) = \sup_{(x_1, x_2, \dots, x_n) \in f^{-1}(y)} \min(\mu_{A_1}(x_1), \mu_{A_2}(x_2), \dots, \mu_{A_n}(x_n))$$

$$(x_1, x_2, \dots, x_n) \in f^{-1}(y)$$

Köv: a matematika módszerek kiterjeszthetők fuzzy –halmazokra.

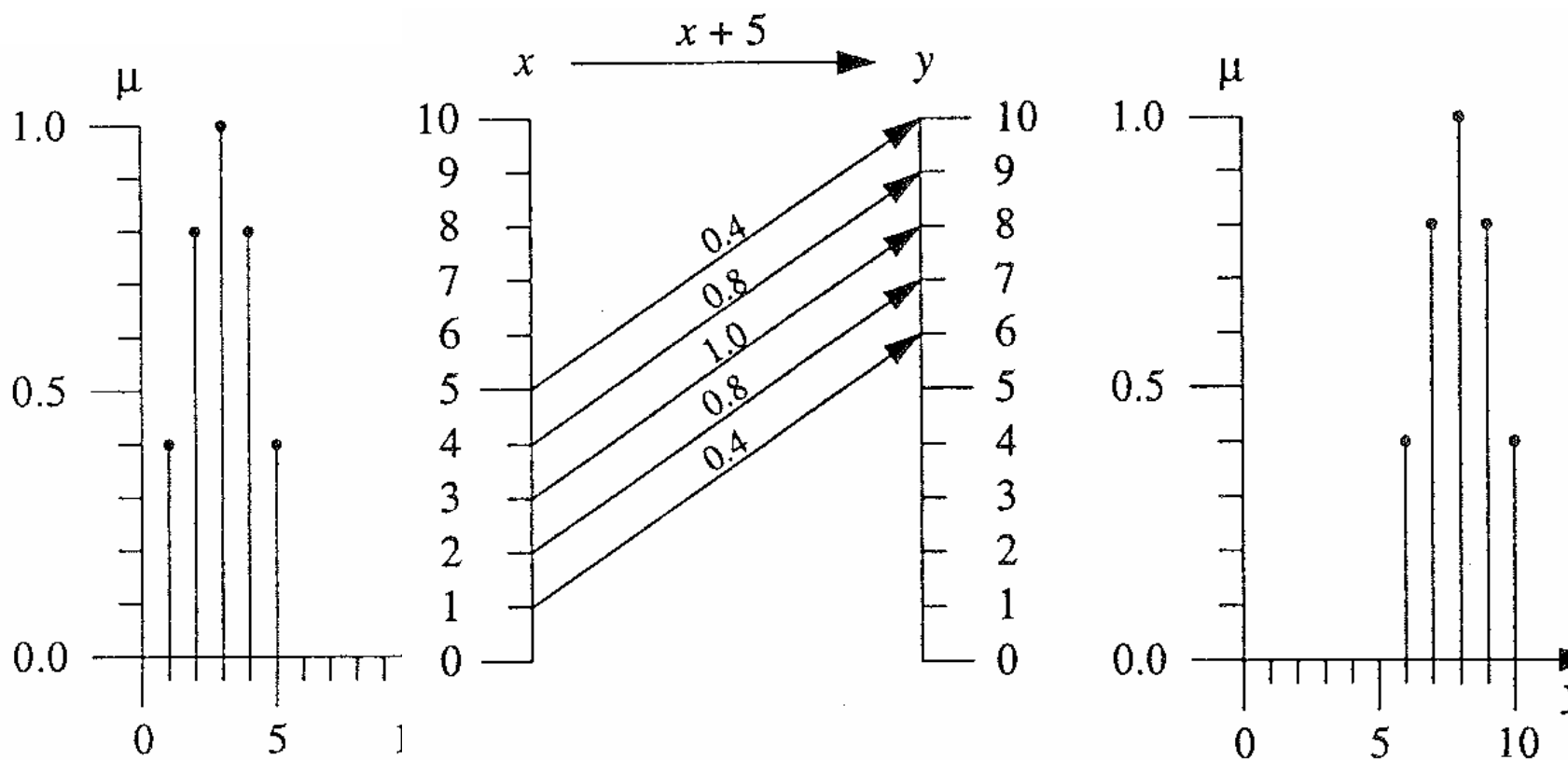
2.Fuzzy halmazelmélet

- Kiterjesztési elv ---- fuzzy aritmetika
pl. egész számok körében $f(x) = x+5$
fuzzy: $f^*(x)$: „Körülbelül 3” + 5
(= „körülberül 8”)

Eredményhalmaz:

$$5 + \{ 0.4/1 + 0.8/2 + 1.0/3 + 0.8/4 + 0.4/5 \}$$
$$= \{ 0.4/6 + 0.8/7 + 1.0/8 + 0.8/9 + 0.4/10 \}$$

2. Fuzzy halmazelmélet



2.Fuzzy halmazelmélet

Néhány definíció

- tartóhalmaz, $\text{supp}(A) = \{ x \mid (\mu_A(x) > 0) \}$
- α -nívóhalmaz (szelet), $A_\alpha = \{ x \mid (\mu_A(x) \geq \alpha) \}$
- normalizált, $\sup \mu_A(x) = 1 \quad (x \in X)$
- Konvex $\mu_A(\lambda x_1 + (1-\lambda)x_2) \geq \min(\mu_A(x_1), \mu_A(x_2))$
- fuzzy szám
 - (konvex, normalizált, egy helyen 1 max., szakszonként folyt)
- fuzzy intervallum
 - (fuzzy szám, egy intervallumon max. értékeket vesz fel és szak. folytonos)

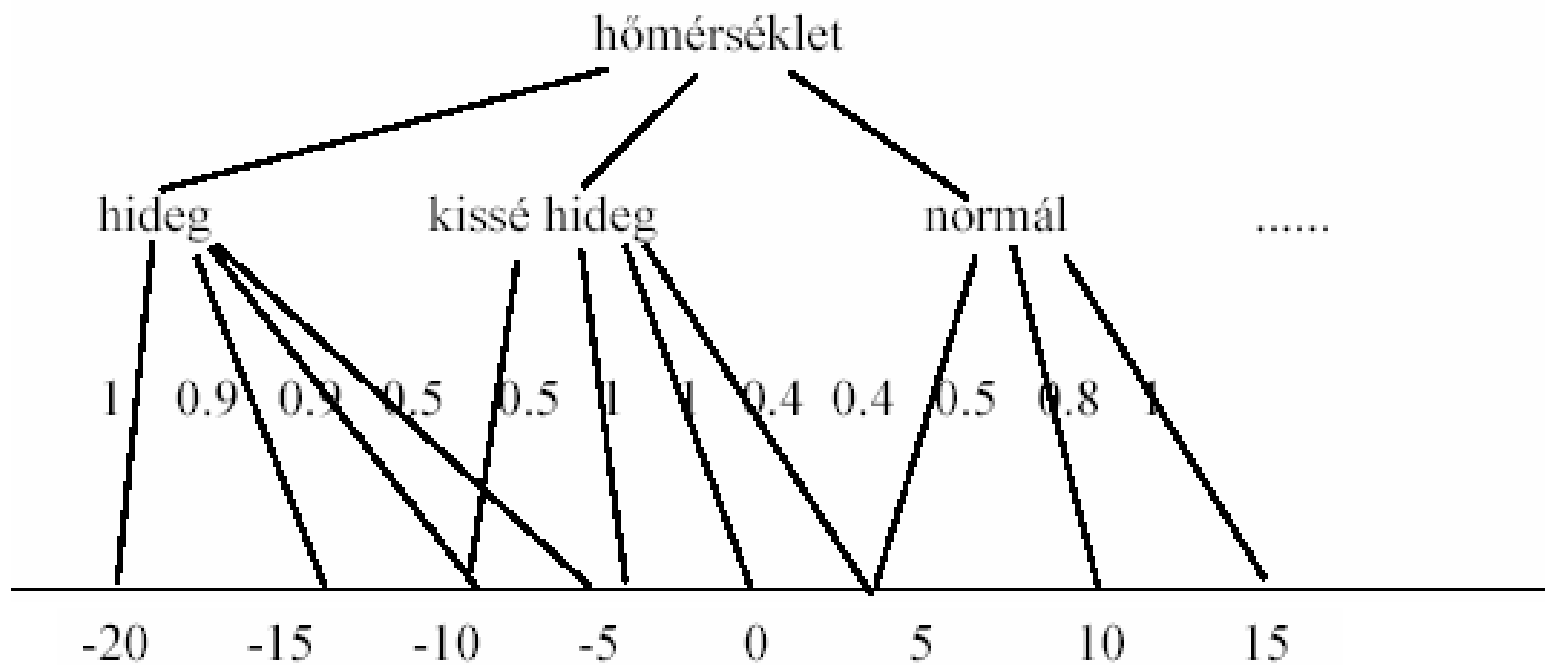
3.Fuzzy logika



- Fuzzy logika
 - **cél:** emberi gondolkodás modellezése összetett, bizonytalanságot tart. felt. esetén
- Jellemzők pl.
 - fuzzy halmazokon megfogalmazott kif.: =predikátum
 - Igazságtérben a log. értékek fuzzy halmazok
 - nyelvi változó
 - számtalan kvantor
 - logikai értékek: fuzzy halmazok (igazságtér)
 - fuzzy következtetés: közelítő, nem precíz

3.Fuzzy logika

- Nyelvi változó pl.



3.Fuzzy logika



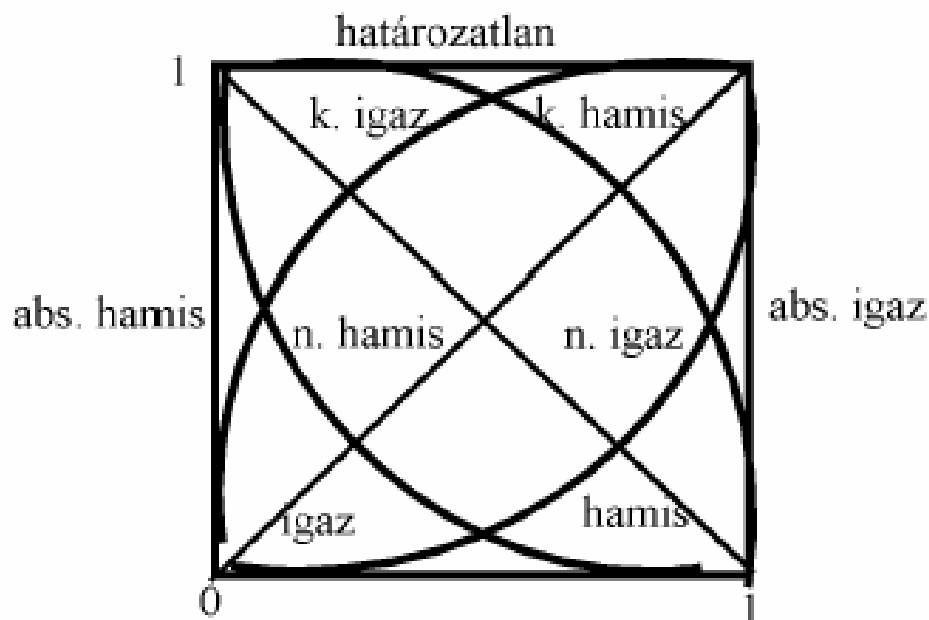
- nyelvi változók
- definíció: (név, $T(N)$ terms halmaz, X,S,M)
 - $T(N)$ terms halmaz (értékek elnevezése)
 - X alaphalmaz,
 - S szintaktikai szabály
 - képzési szab.
 - M szemantikai szababály
 - fuzzy halmaz def.
Kiterjesztési elv alapján generálás

3.Fuzzy logika

$$\mu_{\text{igaz}}(x) = x \quad x \in [0,1]$$

$$\mu_{\text{hamis}}(x) = 1 - x \quad x \in [0,1]$$

$$\mu_{\text{nagyon igaz}}(x) = x^2 \quad x \in [0,1]$$



3.Fuzzy logika

- Szokásos műveletek:

negáció:	$\mu_{\neg A}(x) = 1 - \mu_A(x)$
konjunkció	$\mu_C(z) = \sup_{z=x \wedge y} \min(\mu_A(x), \mu_B(y))$
diszjunció	$\mu_C(z) = \sup_{z=x \vee y} \min(\mu_A(x), \mu_B(y))$
implikáció	$\mu_C(z) = \sup_{z=f(x, y)} \min(\mu_A(x), \mu_B(y))$

- gyakorlat: igazságtér pontértékű
(kevesebb számolás)

pl.: $\mu_C(x) = \min(\mu_A(x), \mu_B(x))$

3.Fuzzy logika

Fuzzy logika művelet csoportok

elemi műveletek továbbfejlesztései

- T-norma műv. (minimum, algebrai szorzat, Einstein-szorzat, ...) „konjunkció”

minimum - művelet

$$\mu_C(x) = \min(\mu_A(x), \mu_B(x)).$$

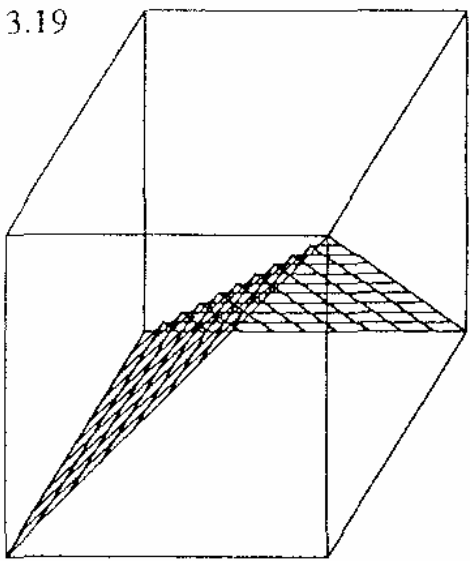
algebrai szorzat

$$\mu_C(x) = \mu_A(x) * \mu_B(x).$$

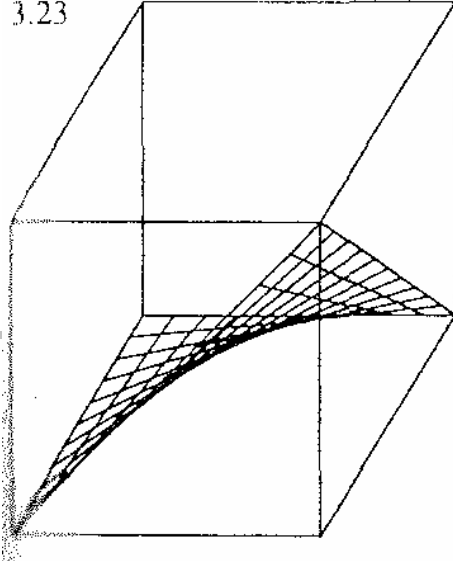
bounded-differencia

$$\mu_C(x) = \max(0, \mu_A(x) + \mu_B(x) - 1).$$

3.19

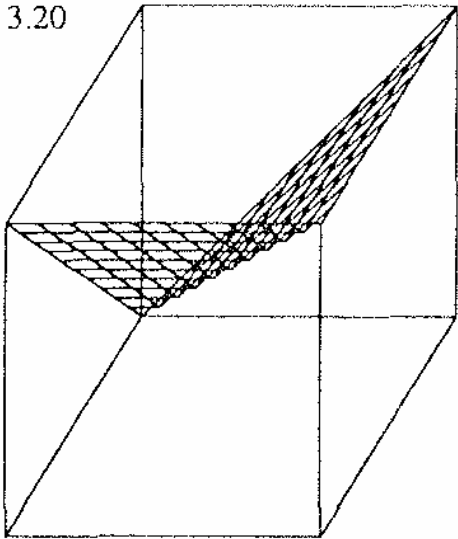


3.23



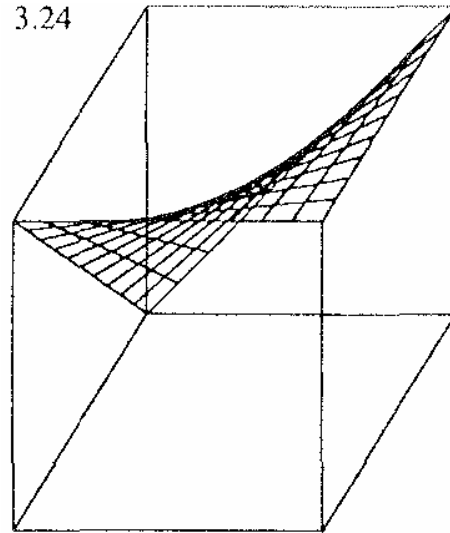
Minimum

3.20



Maximum

3.24



Algebraische Summe

Algebraisches Produkt

3.Fuzzy logika

S-norma műv. (maximum , algebrai összeg, ..)
„diszjunkció” műveletek

maximum művelet

$$\mu_C(x) = \max(\mu_A(x), \mu_B(x)).$$

algebrai összeg

$$\mu_C(x) = 1 - (1 - \mu_A(x)) * (1 - \mu_B(x)).$$

bounded összeg

$$\mu_C(x) = \min(1, \mu_A(x) + \mu_B(x)).$$

Einstein összeg

$$\mu_C(x) = \frac{\mu_A(x) + \mu_B(x)}{1 + \mu_A(x) * \mu_B(x)}$$

3.Fuzzy logika

- Paraméteres műveletek: és, vagy művelethez hasonlók
 - Paraméteres T-norma műv.: pl.

Fuzzy-és művelet $0 \leq \gamma \leq 1$

$$\mu_C(x) = \gamma * \min(\mu_A(x), \mu_B(x)) + 0.5 * (1-\gamma) * (\mu_A(x) + \mu_B(x))$$

- Paraméteres S-norma műv.: pl.

Yager egyesítés művelete $\gamma \geq 1$

$$\mu_C(x) = \min(1, (\mu_A(x)^\gamma + \mu_B(x)^\gamma)^{1/\gamma})$$

3.Fuzzy logika

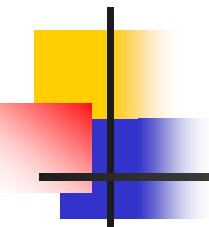
- Kompenzációs műv. (és -vagy közti műv.)
kritériumok egymás hatásait kompenzálhatják,
szituáció modellezés: T és S norma közti műveletek.

Pl.

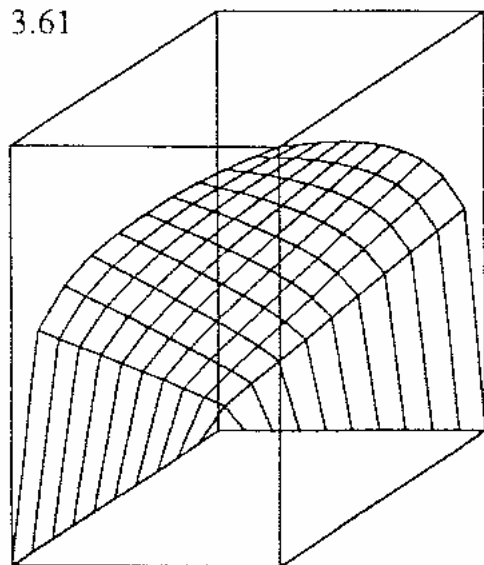
γ -művelet (compensatory-and) $0 \leq \gamma \leq 1$

$$\mu_C(x) = [\mu_A(x) * \mu_B(x)]^{1-\gamma} * [1 - (1 - \mu_A(x)) * (1 - \mu_B(x))]^\gamma$$

$\gamma=0$ -ra az algebrai szorzat, $\gamma=1$ -re az algebrai összeg műveletet adja.



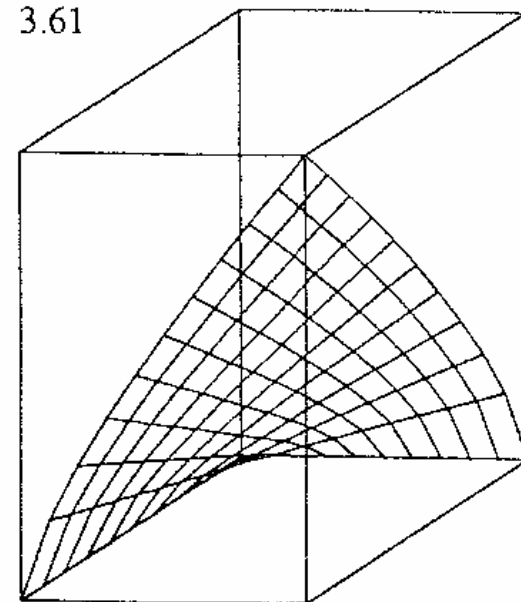
3.61



γ -Operator, $\gamma = .75$

	1	9	8	7	.6	5	4	3	2	.1	0	
0	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	.00	
1	.56	.51	.46	.41	.35	.30	.25	.20	.14	.09	.00	0
2	.67	.61	.55	.50	.44	.38	.33	.27	.21	.02	.00	1
3	.74	.68	.63	.57	.51	.45	.39	.33	.07	.04	.00	2
4	.80	.74	.68	.63	.57	.51	.45	.14	.10	.06	.00	3
5	.84	.79	.73	.68	.63	.57	.23	.18	.13	.07	.00	4
6	.88	.83	.78	.73	.68	.33	.27	.22	.16	.09	.00	5
7	.91	.87	.83	.78	.44	.38	.32	.25	.19	.11	.00	6
8	.95	.91	.87	.57	.51	.44	.37	.29	.21	.13	.00	7
9	.97	.94	.71	.64	.56	.49	.41	.33	.24	.14	.00	8
1	1.0	.85	.78	.70	.62	.54	.46	.37	.27	.16	.00	9
	1.0	.92	.85	.77	.68	.59	.50	.41	.30	.18	.00	1
	1	9	8	7	6	5	4	3	2	1	0	

3.61



γ -Operator, $\gamma = .25$

3.Fuzzy logika

Fuzzy közelítő következtetés

az általánosított modus ponens szimbolikus alakja:

$$\frac{A \rightarrow B, \quad A'}{B' = A' \circ (A \rightarrow B)}$$

ahol \circ a max-min a komp.

$$\mu_{B'}(v) = \max \min (\mu_{A'}(u), \mu_{A \rightarrow B}(u, v))$$

- Változatok: implikációra, kompozícióra

max-min köv.: implikáció: minimum és
max-min kompozíció

max-szorzat köv.: implikáció: algebrai szorzat és
max-min kompozíció

3.Fuzzy logika

- Fuzzy közelítő következtetés példa:

$$A = (0/100, 0.5 /125, 1/150, 0.5/175, 0/200)$$

$$B = (0/10, 0.6/20, 1/30, 0.6/40, 0/50).$$

Az $A \rightarrow B$ reláció M mátrixa:

$$M = \begin{vmatrix} 0 & 0 & 0 & 0 & 0 \\ 0 & 0.5 & 0.5 & 0.5 & 0 \\ 0 & 0.6 & 1 & 0.6 & 0 \\ 0 & 0.5 & 0.5 & 0.5 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{vmatrix}$$

ahol $m_{ij} = \min (a_i, b_j)$.

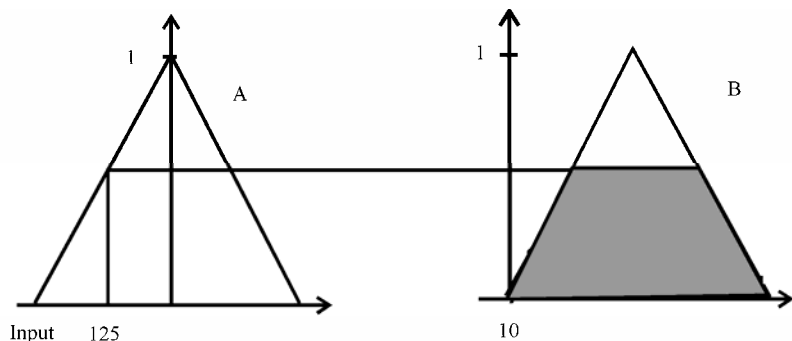
$$A^* = (0/100, 0.5/125, 0.5/150, 0/175, 0/200)$$

Max-min köv.: $B^* = (0/10, 0.5/20, 0.5/30, 0.5/40, 0/50).$

Max-szorzat: $B^* = (0/10, 0.3/20, 0.5/30, 0.3/40, 0/50)$

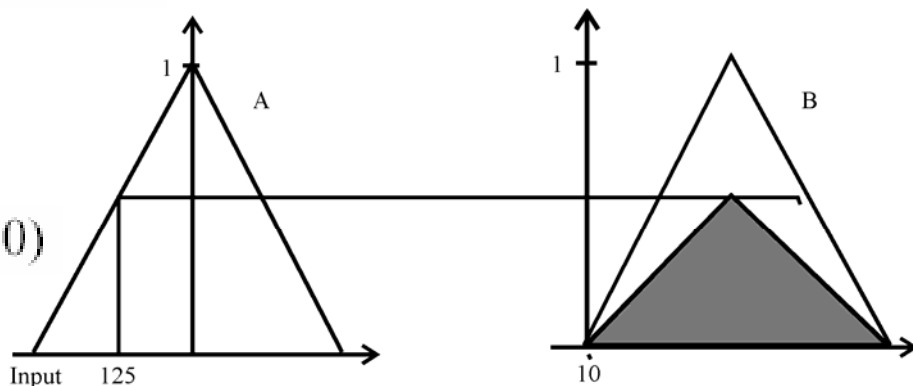
3.Fuzzy logika

Max-min és max-szorzat következtetés



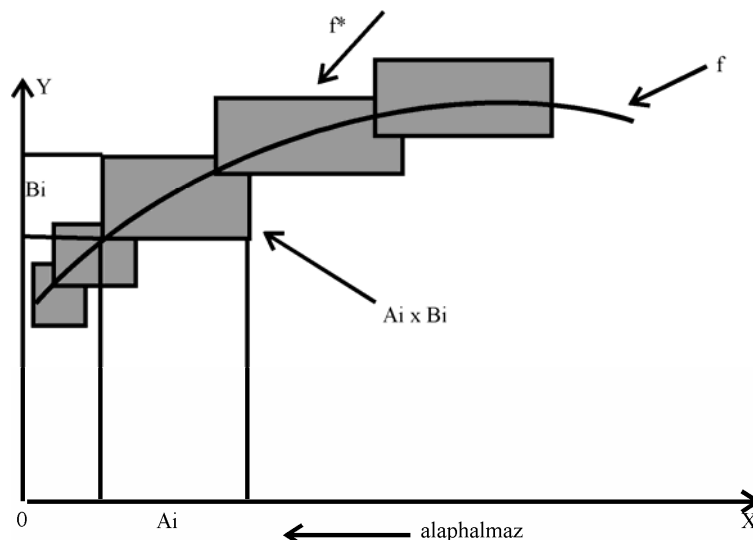
$$B' = (0/10, 0.5/20, 0.5/30, 0.5/40, 0/50).$$

$$B' = (0/10, 0.3/20, 0.5/30, 0.3/40, 0/50)$$



3.Fuzzy logika

- Fuzzy szabály, diagram
 f^* : IF X is A_1 THEN Y is B_1
.....
IF X is A_n THEN Y is B_n
Szabályok együtt -- diagram:



Fuzzy-rendszerek



Gyakori fuzzy-rendszer modellek



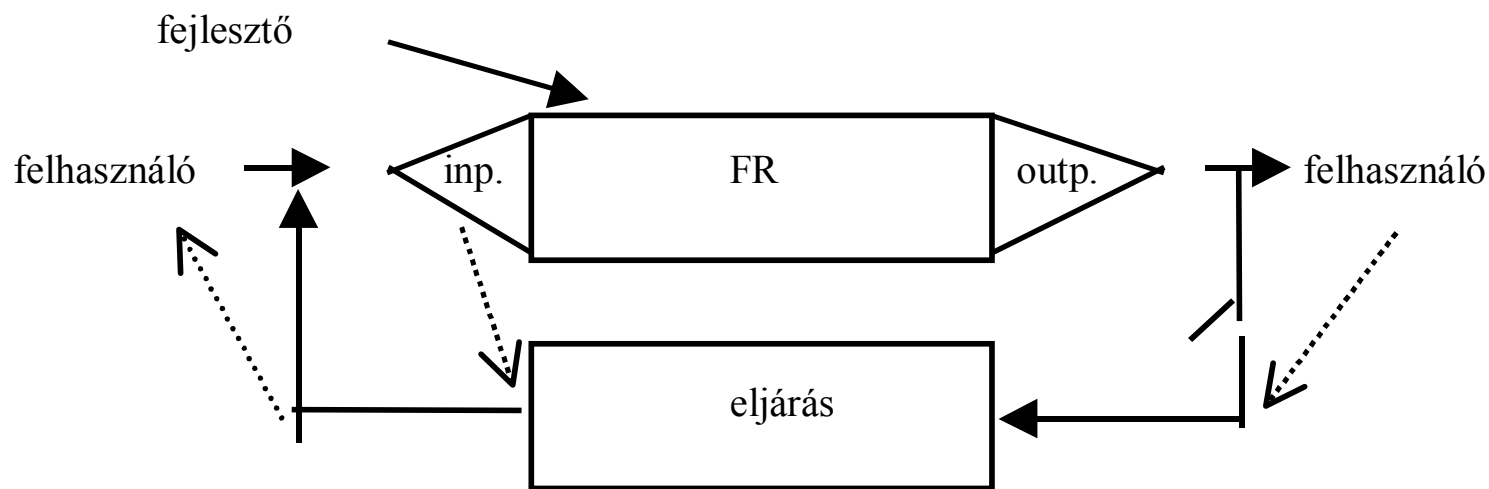
Fuzzy-rendszerek

Témakörök

- Fuzzy rendszer típusok
- Fuzzy szabályozó
 - Mamdani, Sugeno típus
- Fuzzy reláció egyenletrendszer
- Produkciós rendszerek
- Hibrid rendszerek

Fuzzy rendszerek

1. Az általános fuzzy rendszer



Fuzzy rendszerek

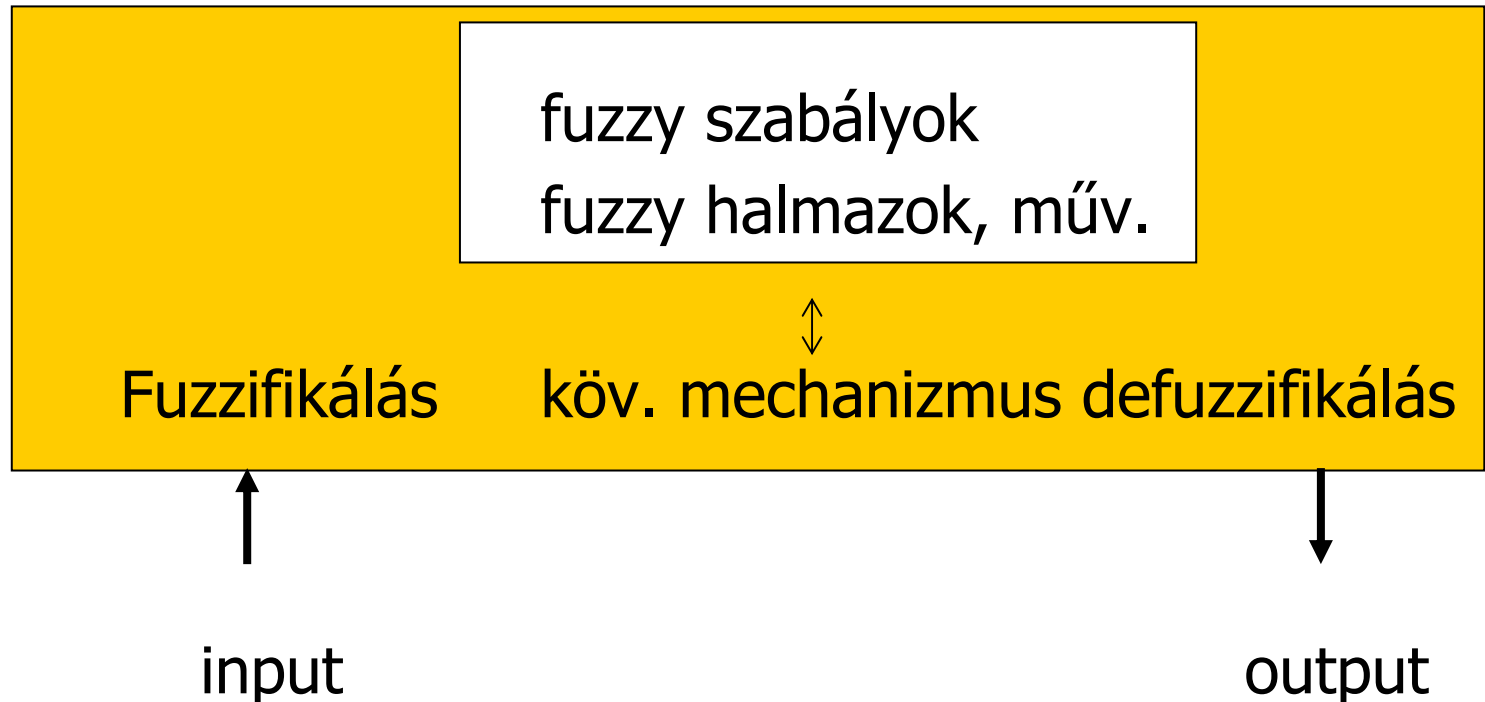
1. Az általános fuzzy rendszer

Gyakori fuzzy-rendszer modellek

- Szabályalapú rendszerek
 - fuzzy szabályozó,
 - reláció egyenletrendszer,
 - produkciós rendszerek
- Hibrid rendszerek
 - neurofuzzy: kooperatív, hibrid;
 - fuzzy neurális hálózat
 - fuzzy SZR

2. Fuzzy szabályozók

Felépítés:



2. Fuzzy szabályozók

Fuzzy szabályozók

- fuzzifikálás
- tudásbázis
 - Fuzzy szabályok : - két típus
 - Mamdani modell (diagram)

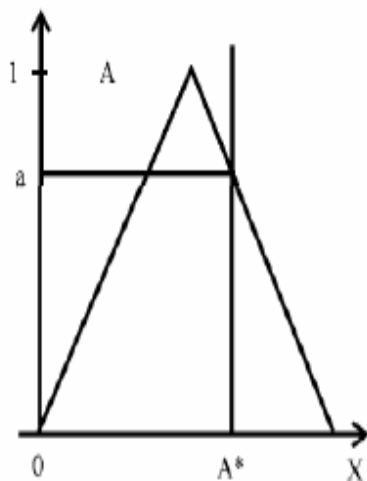
(w_i) IF x_1 is A_1 AND ... x_n is A_n THEN y is B_j CF

- Sugano (TSK-) modell

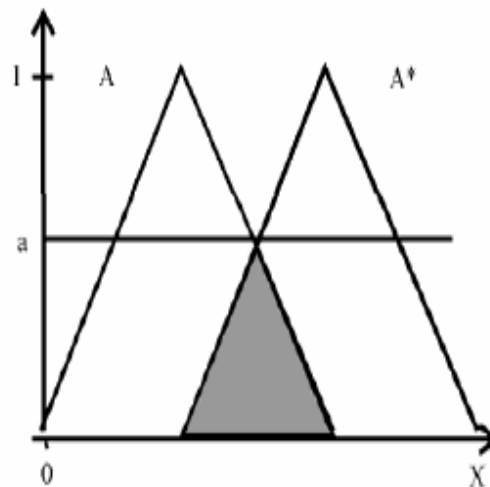
r : IF x_1 is A_1 AND... x_n is A_n THEN $y = f_r(x_1, x_2, \dots, x_n)$
ált.: $y = a_0^r + a_1^r x_1 + \dots + a_n^r x_n$

2. Fuzzy szabályozók

Fuzzifikálás



a.)

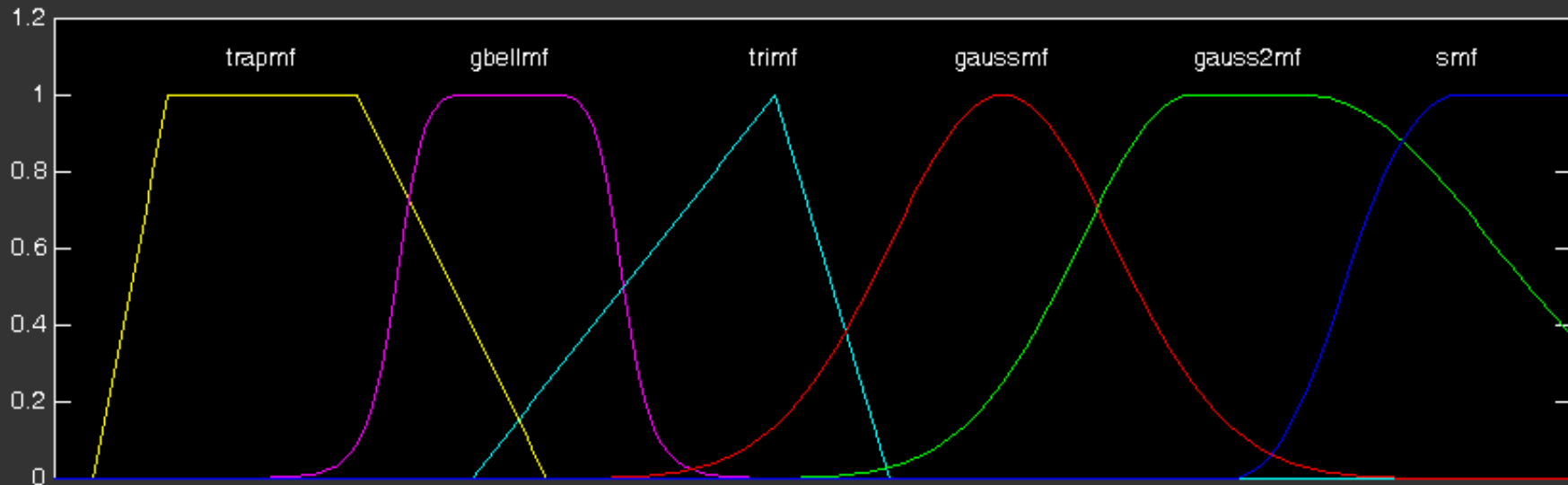


b.)

2. Fuzzy szabályozók

Fuzzy szabályozók

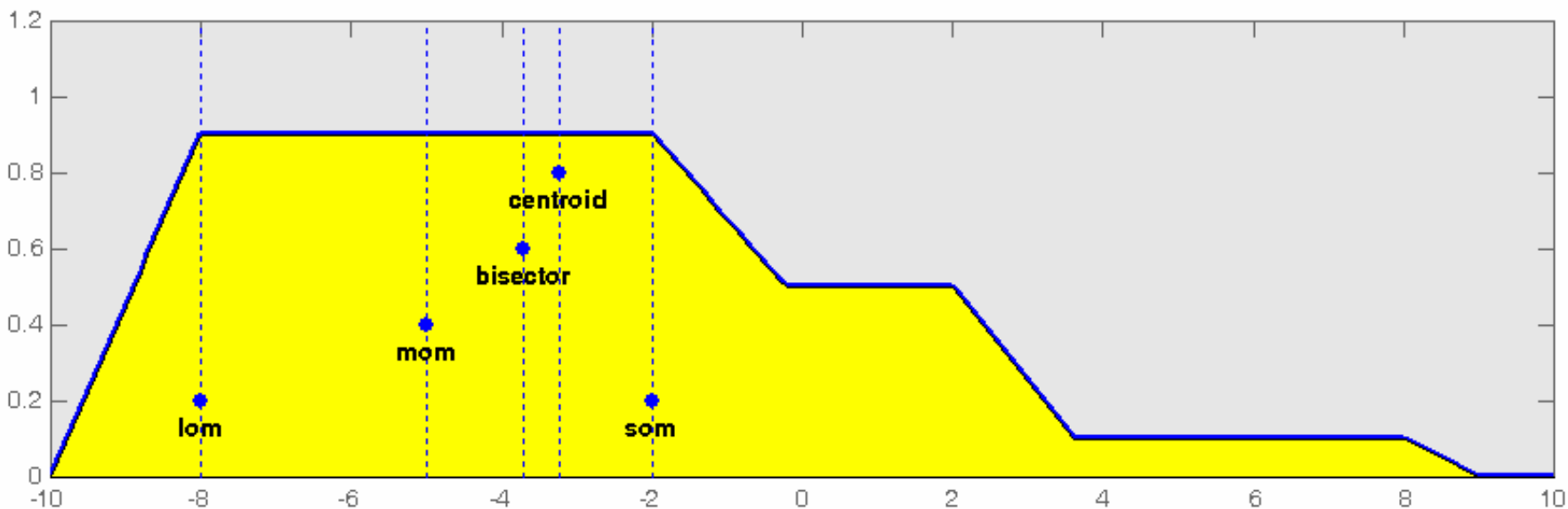
- nyelvi változók (tartalmazási függv.: háromszög, Gauss, trapéz,...)



2. Fuzzy szabályozók

Fuzzy szabályozók

- defuzifikálás



2. Fuzzy szabályozók

Következtetés Mamdani modellnél

- Szabályok párhuzamos kiértékelése
- Eredményhalmazok uniója (vagy, $w_i *$)
- A B output változó $B_1, B_2, \dots B_n$ értékeinél

$$B' = B'_1 + B'_2 + \dots + B'_n$$

- Defuzzifikálás: egy valós szám

Pl.

2. Fuzzy szabályozók

Következtetés Mamdani modellnél

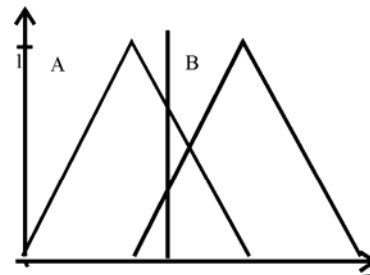
If X is A then Y is A

If X is B then Y is B.

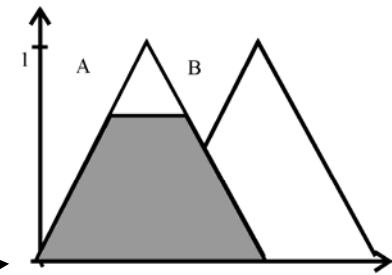
Input: $A^* = 1.3$

1.Szabály eredmény

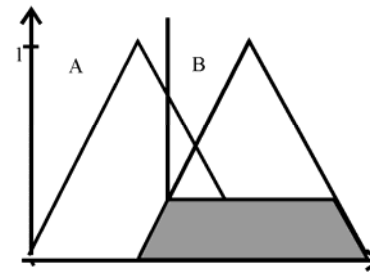
2.Szabály eredmény



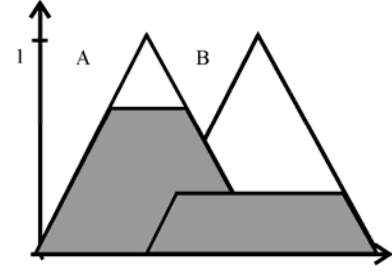
a.)



b.)



c.)



d.)

defuzzifikálás: pl. COG

2. Fuzzy szabályozók

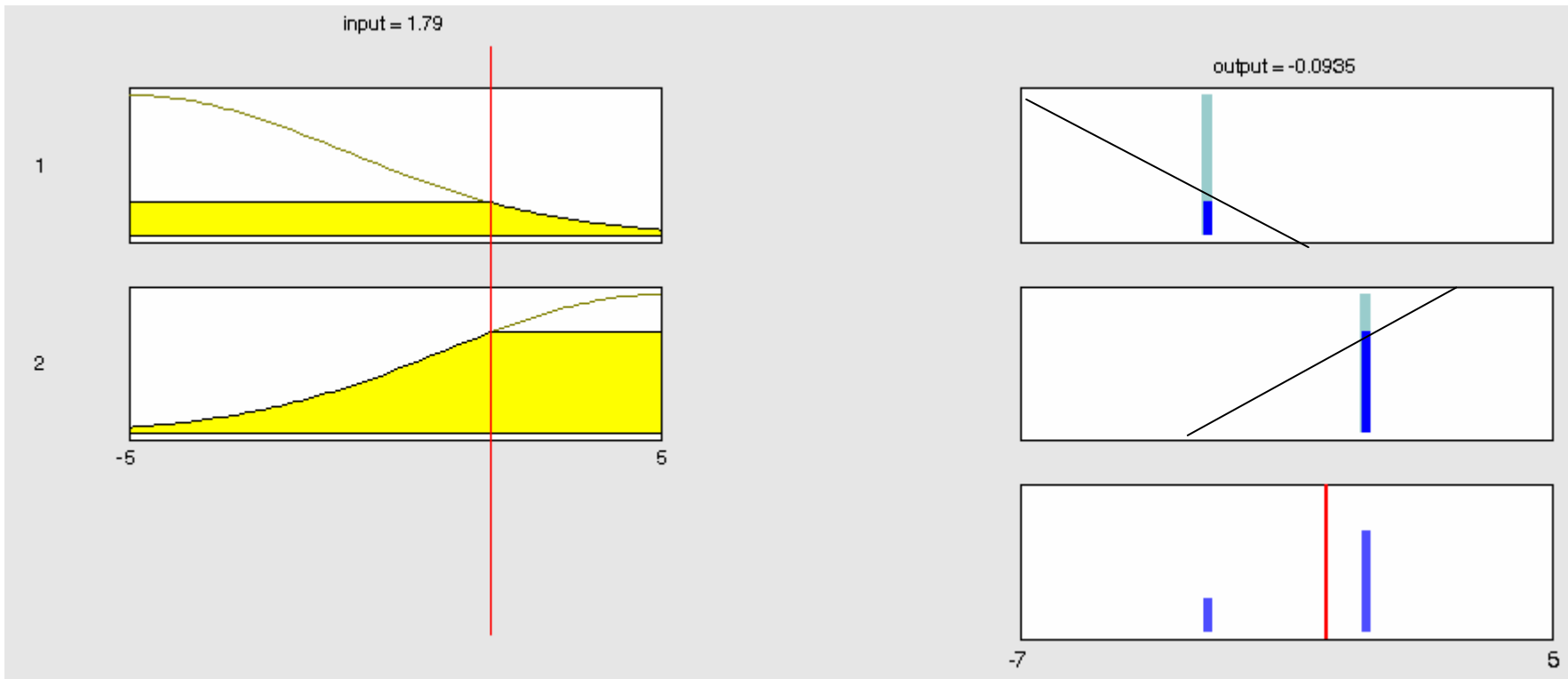
Következtetés a Sugeno modellnél

- Szabályok párhuzamos kiértékelése
 - r-edik szabálynál: $f_r(x_1, x_2, \dots, x_n)$ és a „súlya”:
$$\mu_{Rr}(x_1, x_2, \dots, x_n) = \mu_{r1}(x_1) \wedge \dots \wedge \mu_{rn}(x_n)$$
- Output: „defuzzifikálással”

$$y = f(x) = \frac{\sum_{r=1}^n \mu_{Rr}(x_1, x_2, \dots, x_n) * f_r(x_1, x_2, \dots, x_n)}{\sum_{i=1}^n \mu_{Rr}(x_1, x_2, \dots, x_n)}$$

2. Fuzzy szabályozók

- Sugeno példa: if input is low then output is line1
if input is high then output is line2



3. Fuzzy reláció egyenletrendszer

Reláció egyenletrendszer

- $A \rightarrow B$ szabály (implikáció) értelmezhető mint-- $B = A \circ R$ reláció
- a szabályrendszer: $B_i = A_i \circ R \quad (i=1,2,\dots,n)$.
- Akkor oldható meg, ha az

$$R = \bigcap_{i=1}^n A_i \rightarrow B_i$$

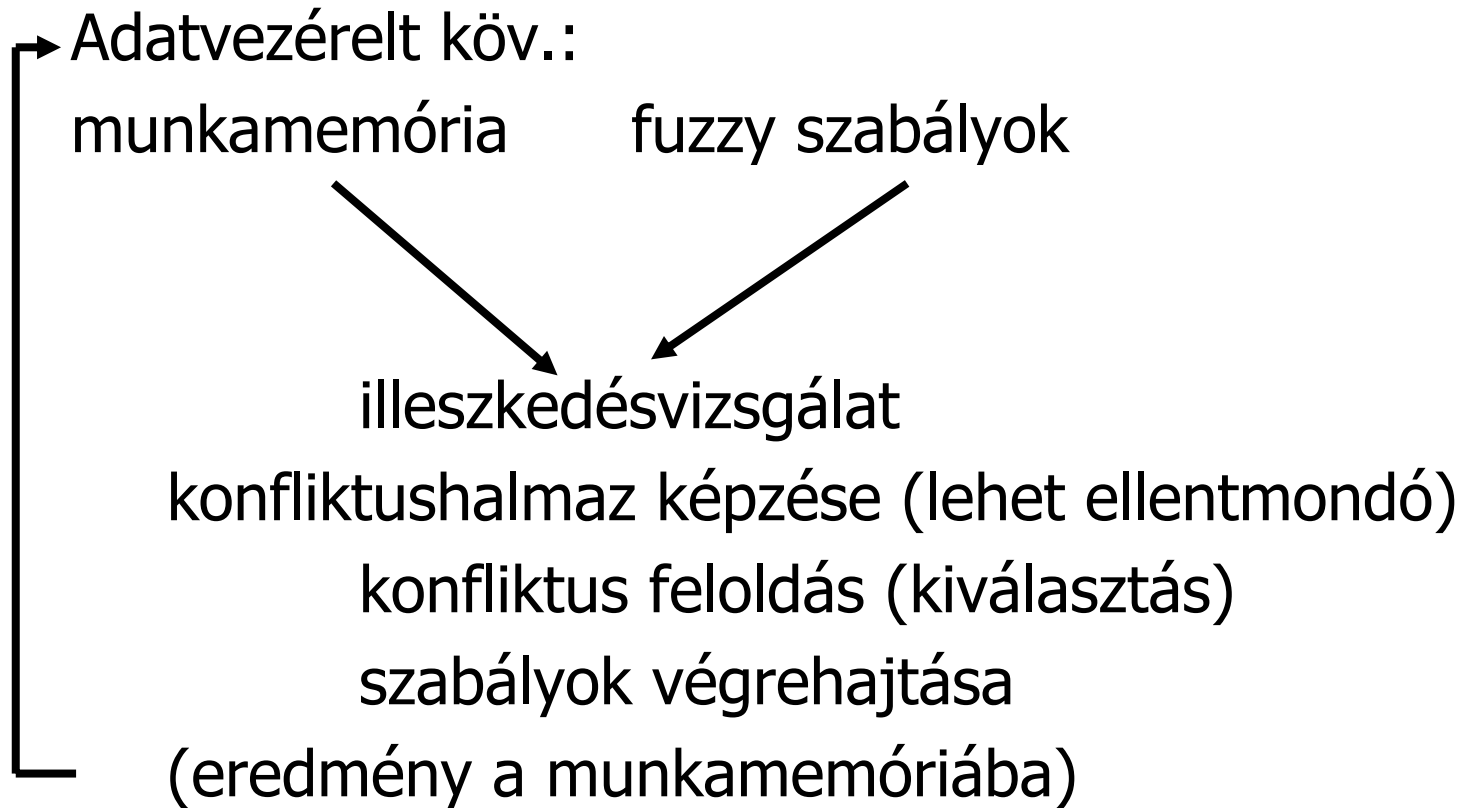
megoldása a rendszernek

4. Fuzzy produkciós rendszerek

Fuzzy produkciós rendszerek

- Szabályrendszer, következtetési mechanizmus + bizonytalan adatok
(fuzzy halmazok /lehetőségességi eloszlások)
- Az FPR felépítése: tudásbázis, ismeretszerző modul, munkamemória, következtető mech., magyarázó modul, felh. interfész
(SZR szabályformalizmus)
- Következtetés: adatvezérelt (soros, párhuzamos, hasonlóságon alapuló), célvezérelt, kombinált

4. Fuzzy produkciós rendszerek



4. Fuzzy produkciós rendszerek

- Pl. Fuzzy Toolbox (Turksen, Zwing) adatvezélet következtetése hasonlósági mértékkel:
 - Szabályok: n db. If A is A_j Then B is B_j ; Input: $A^*_1, A^*_2, \dots, A^*_n$
 - lépések
 - (1) Minden A_j, A_i^* párra meghatározza az $s_j = s(A_j, A_i^*)$ hasonlósági értéket.
 - (2) Meghatározza az aktivizálható szabályok R^* halmazát (konfliktus halmaz).
 - (3) R^* minden R_j szabályánál az s_j függvényében kiszámolja a B_j^* konklúziót.
 - (4) A B_j^* konklúziók kombinációjaként előállítja a B^* konklúziót:

$$\mu_{B^*}(x) = f(\mu_{B_1^*}(x), \mu_{B_2^*}(x), \dots, \mu_{B_m^*}(x)) \quad (x \in B)$$

ahol f a fuzzy halmazokon értelmezett T-, vagy S-norma művelet.

4. Fuzzy produkciós rendszerek

- Pl. Fuzzy Toolbox (Turksen, Zwing) adatvezélet következtetése hasonlósági mértékkel:
 - Hasonlóság képlet: $s(A,B) = 1 / (1 + d(A,B))$
 - R^* meghatározása: legnagyobb $S(A,B)$ értékű szabály néhány legnagyobb s - értékű sz., $s(A,B) >$ limitet teljesítők
 - Pontértékű tagsági függvények
 - Következmény számolás:

$$\mu_{B_j^*}(x) = \min(1, \mu_{B_j}(x) / s_j), \quad \text{vagy}$$
$$\mu_{B_j^*}(x) = \max(\mu_{B_j}(x), s_j) \quad \text{ahol } x \in Z.$$

4. Fuzzy produkciós rendszerek

- Pl. SYTEM Z-11 célvezérelt következtetése:
- Lehet: éles-, fuzzy tény, CF faktor
- Következtető séma:
számítása:

$$\frac{\text{IF A THEN B } CF_R}{A^* CF_{A^*}} \\ B^* CF_{B^*}$$

feltétel (A)	következmény (B)	konklúzió (B*)	CF _{B*}
precíz	precíz	$B^* = B$	$CF_{B^*} = CF_A, CF_R$
precíz	fuzzy	$B^* = B$	$CF_{B^*} = CF_A, CF_R$
fuzzy	precíz	$B^* = B$	$CF_{B^*} = CF_A, CF_R S(A, A^*)$
fuzzy	fuzzy	$B^* = (A, B) \circ A^*$	$CF_{B^*} = CF_A, CF_R$

4. Fuzzy produkciós rendszerek

- Pl. SYTEM Z-11 célvezérelt következtetése:
- Összetett következtető séma:

$$\begin{array}{l} \text{IF } A1 \text{ AND } A2 \text{ THEN } B \text{ } CF_R \\ A1^* \\ A2^* \quad CF_{A2^*} \\ \hline B^* \end{array}$$

modus ponens kiértékelésének folyamata:

$$\begin{array}{l} \text{IF } A1 \text{ THEN } B \\ A1^* \\ \hline B1^* \end{array}$$
$$\begin{array}{l} \text{IF } A2 \text{ THEN } B \\ A2^* \\ \hline B2^* \end{array}$$

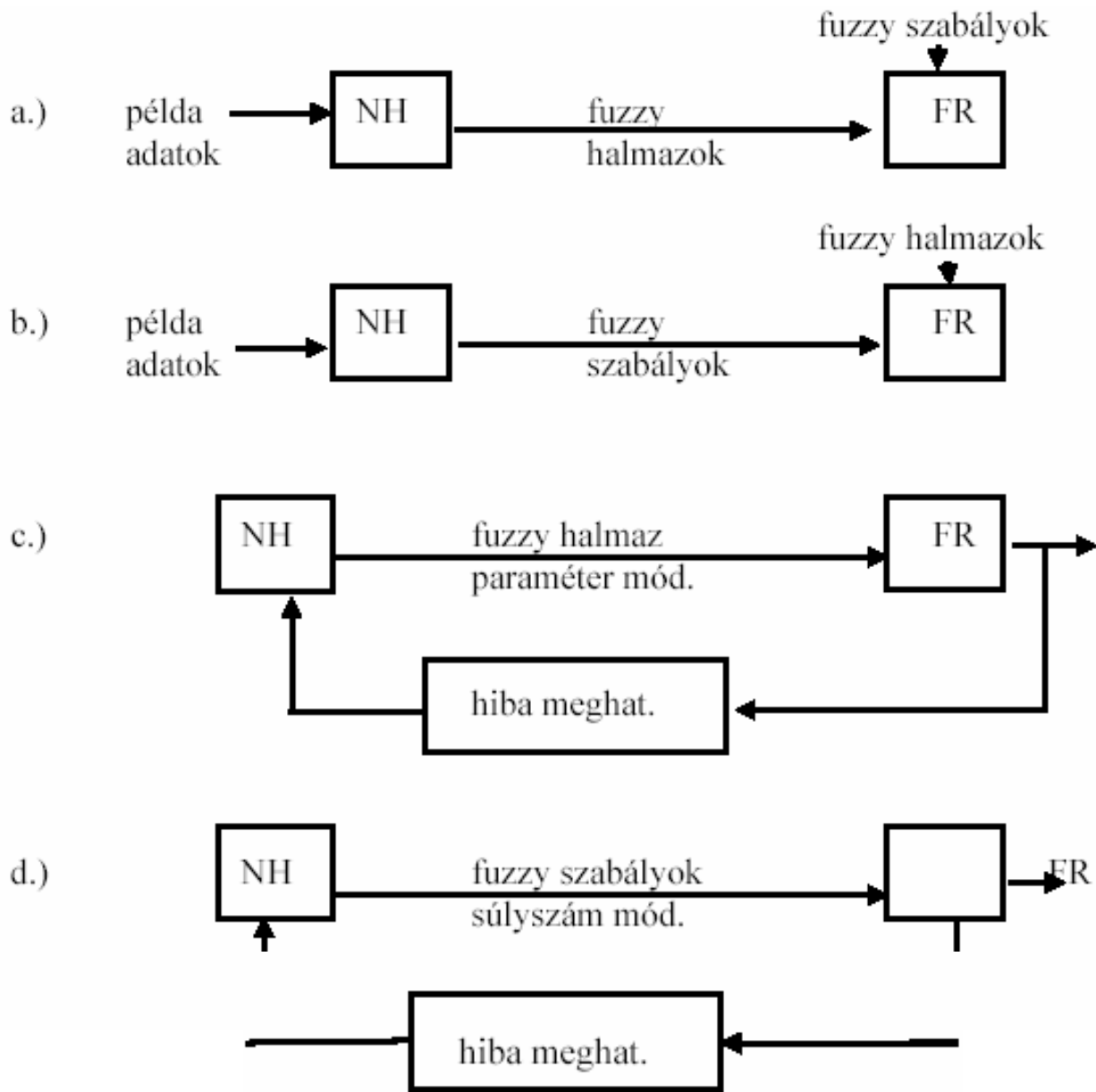
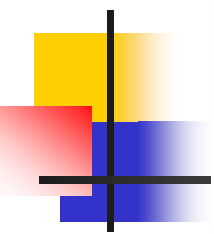
és $B^* = B1^* \cup B2^*$, $CF_{B^*} = \min(CF_{A1^*}, CF_{A2^*}) CF_R$ lesz.

5. Hibrid fuzzy rendszerek



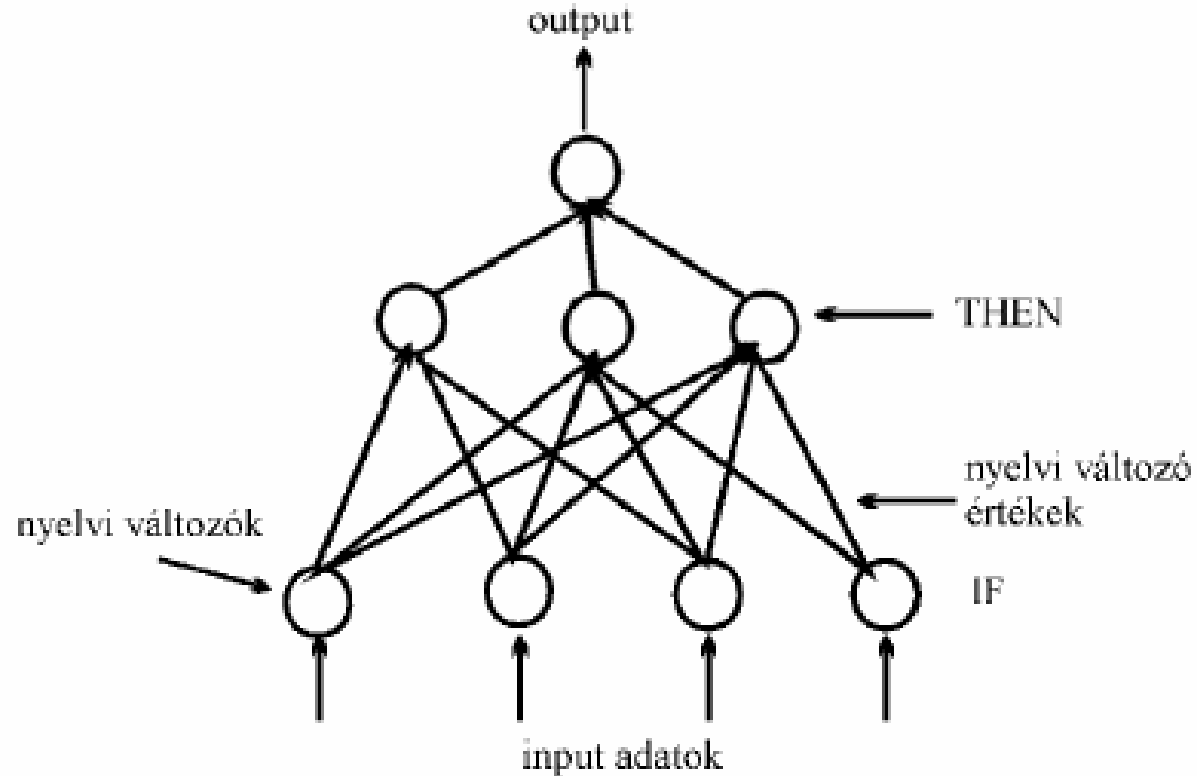
Hibrid rendszerek

- Fuzzy SZR (fuzzy technika adaptálása)
(REVEAL, CLIFS)
- Neurofuzzy rendszerek
 - kooperatív
 - offline kapcs. (f. szabályok, f. halmazok)
 - online kapcs. (f. halmaz paraméter, f.szabály súly módosítás)



5. Hibrid fuzzy rendszerek

- hibrid neurofuzzy r.



- fuzzy neurális hálózat (f. input, f. neuron, mat. művelet helyett: f. relációk, f. műveletek)



Fuzzy mintapéldák

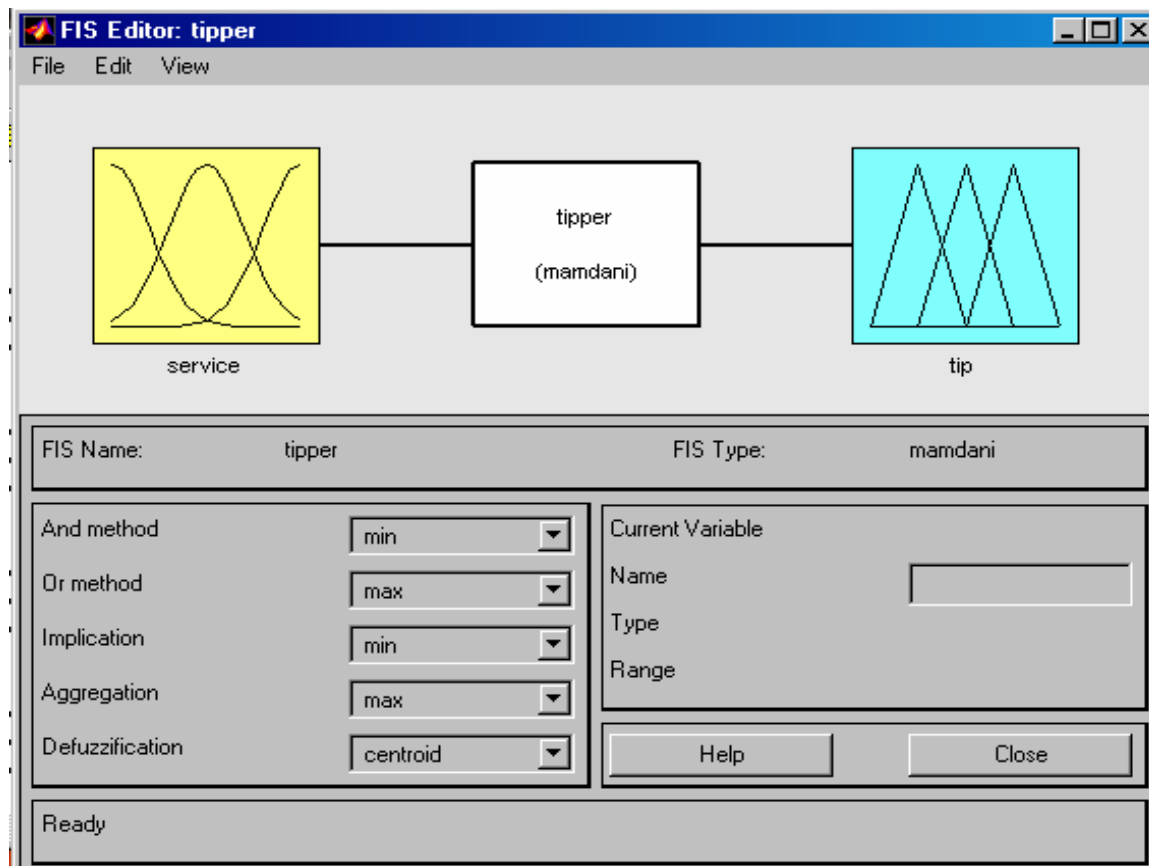
MATLAB fuzzy tools



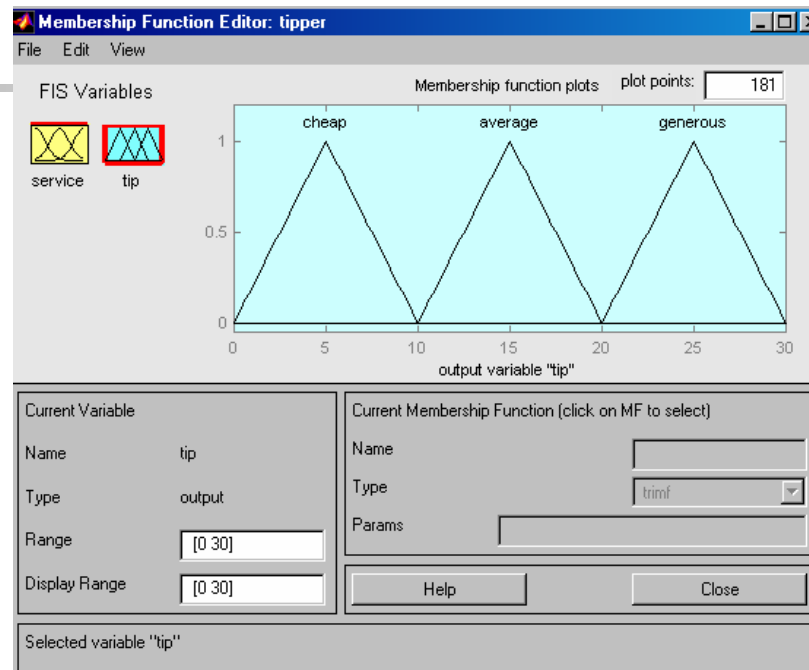
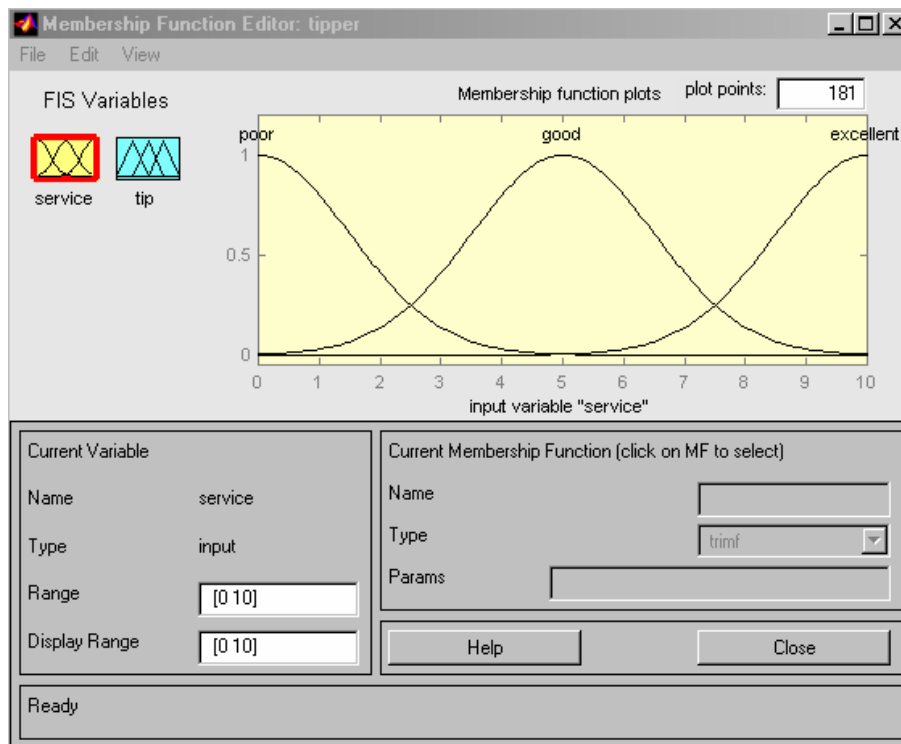
1. mintapélda

- Készítsünk egy Mamdani típusú fuzzy szabályozót, amely javaslatot tesz arra, hogy egy étteremben hány % borravalót adjunk a pincérnek.
 - Döntési szempontok:
 - a.) a borravaló csak a kiszolgálás minőségétől függ (0-10 pont)
 - b.) az étel minősége is befolyásolja (0-10 pont)

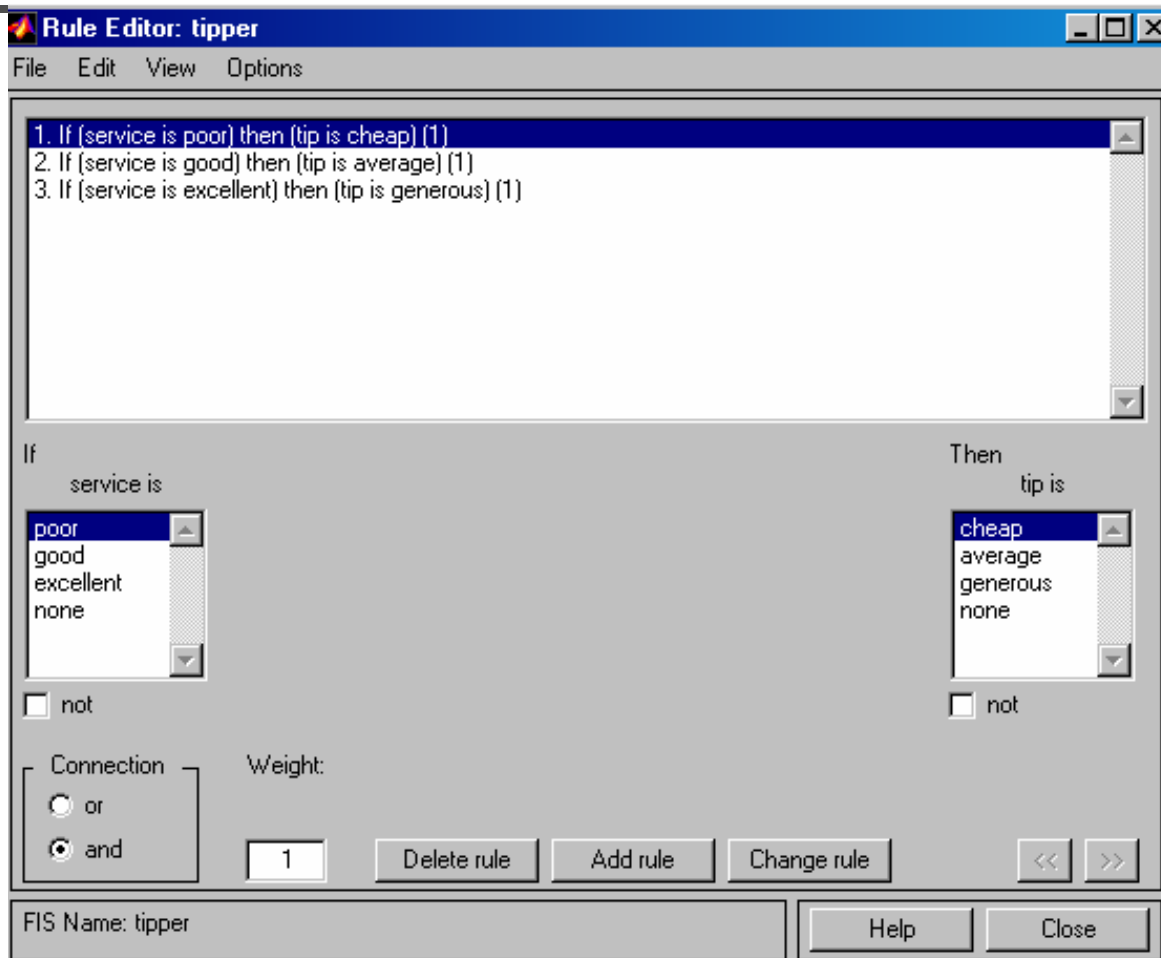
1. mintapélda



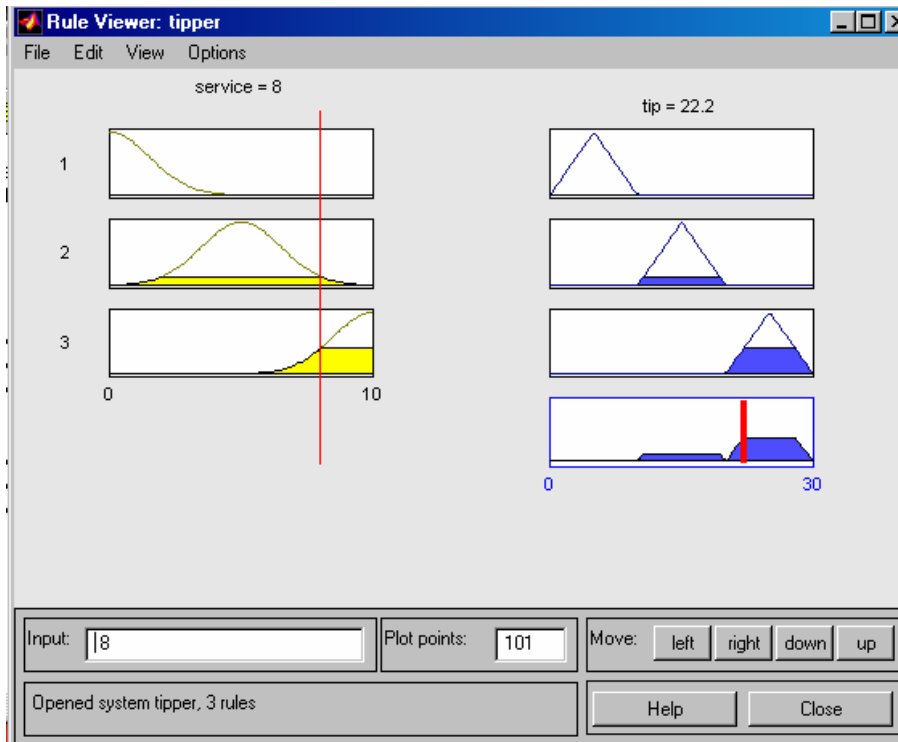
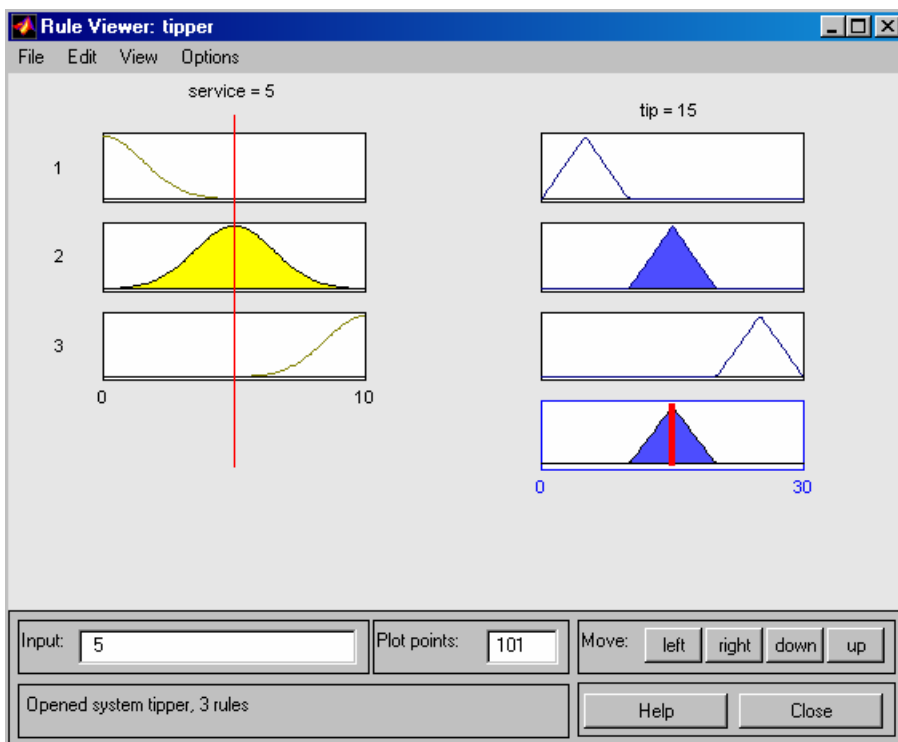
1. mintapélda



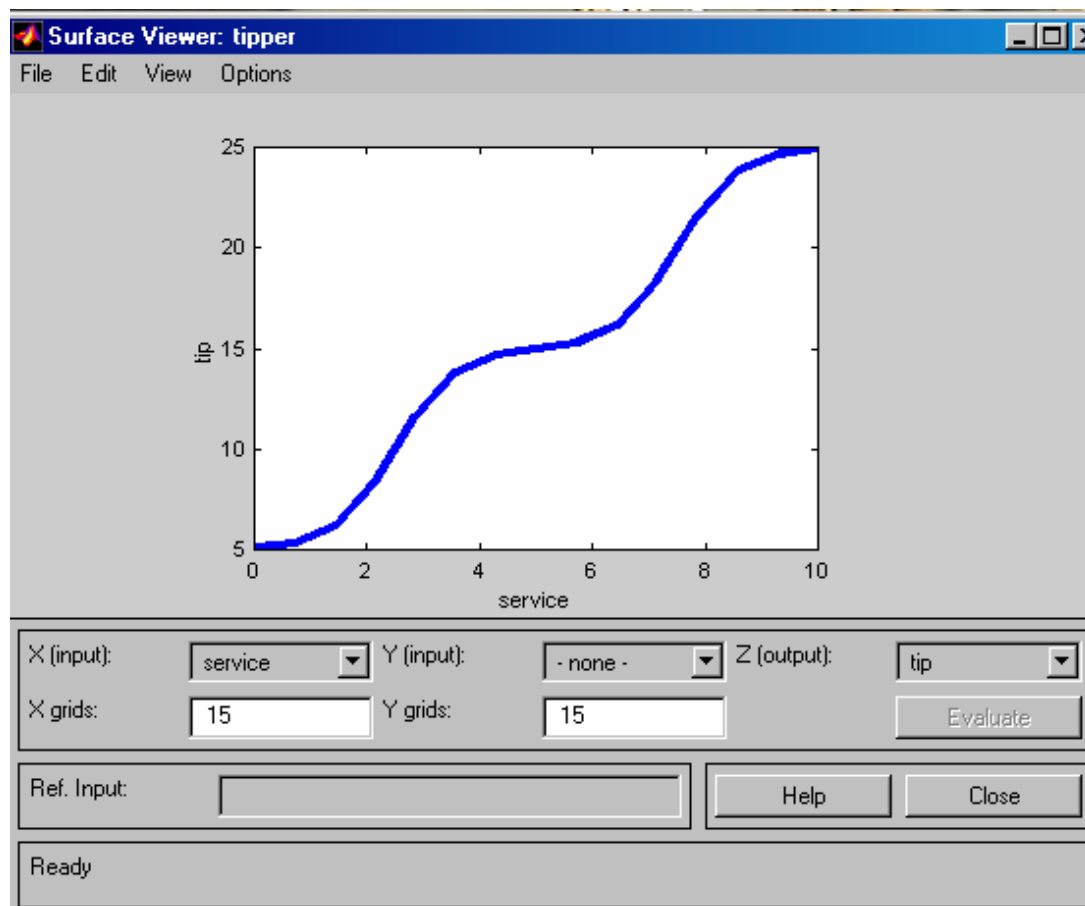
1. mintapélda



1. mintapélda

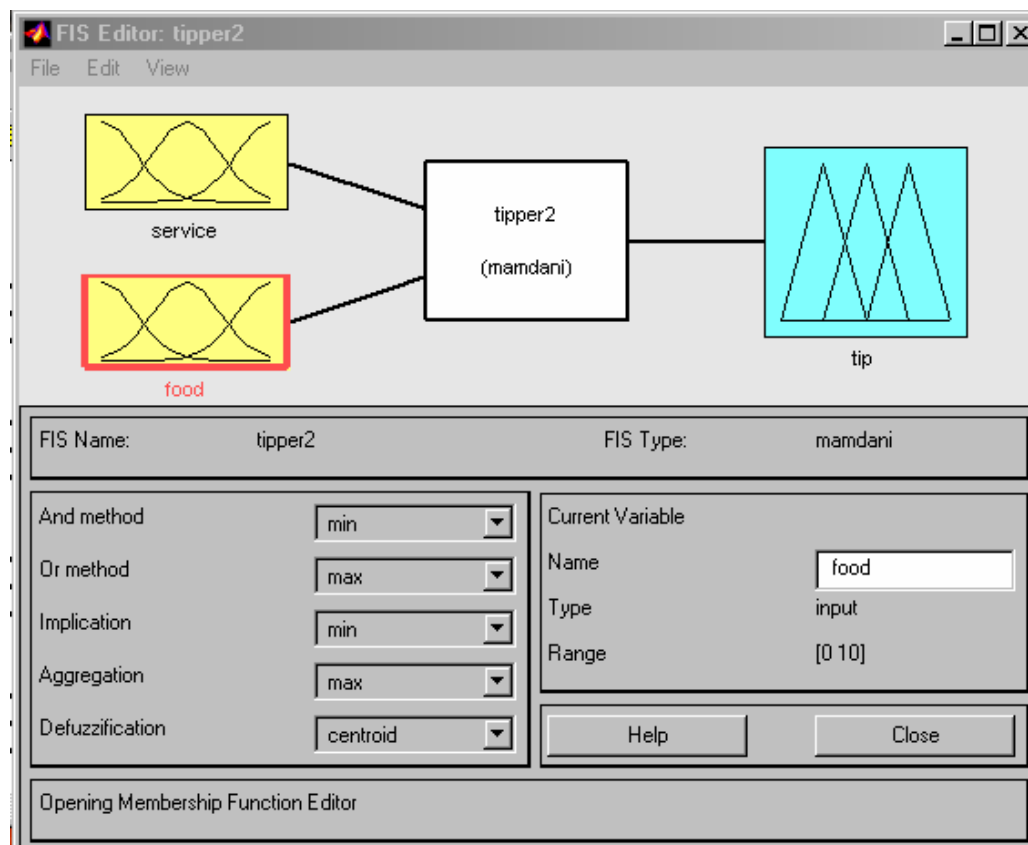


1. mintapélda



1. mintapélda

- Újabb szempont: az étel minősége



1. mintapélda

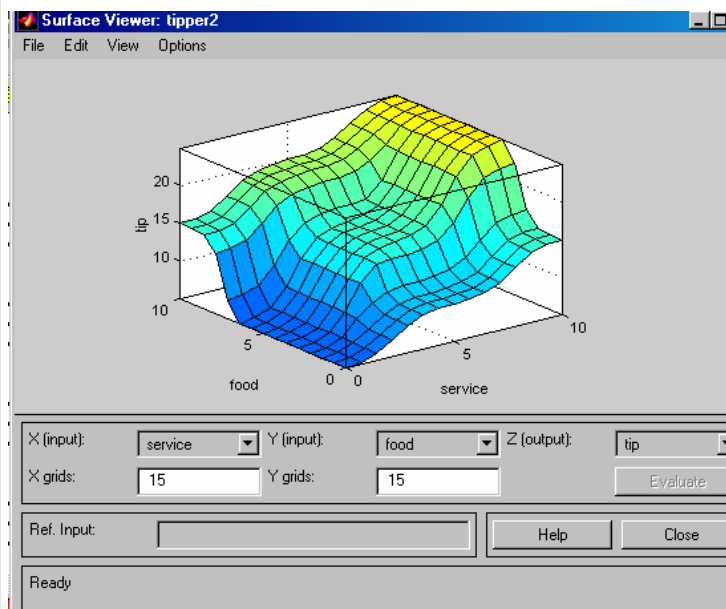
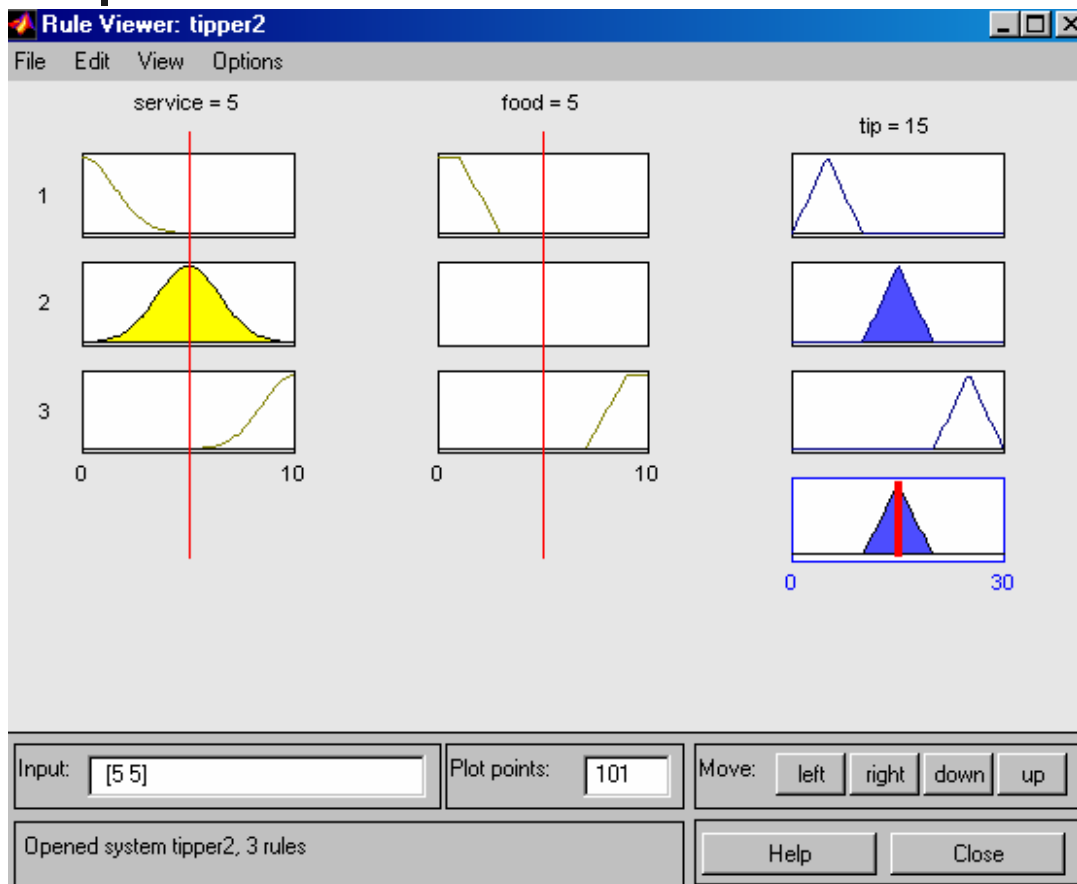
The screenshot shows a window titled "Rule Editor: tipper2" with a menu bar containing "File", "Edit", "View", and "Options". The main area contains three rules:

1. If (service is poor) or (food is rancid) then (tip is cheap) (1)
2. If (service is good) then (tip is average) (1)
3. If (service is excellent) or (food is delicious) then (tip is generous) (1)

Below the rules, there are three dropdown menus for the variables "service is", "food is", and "tip is". Each menu has a "not" checkbox below it.

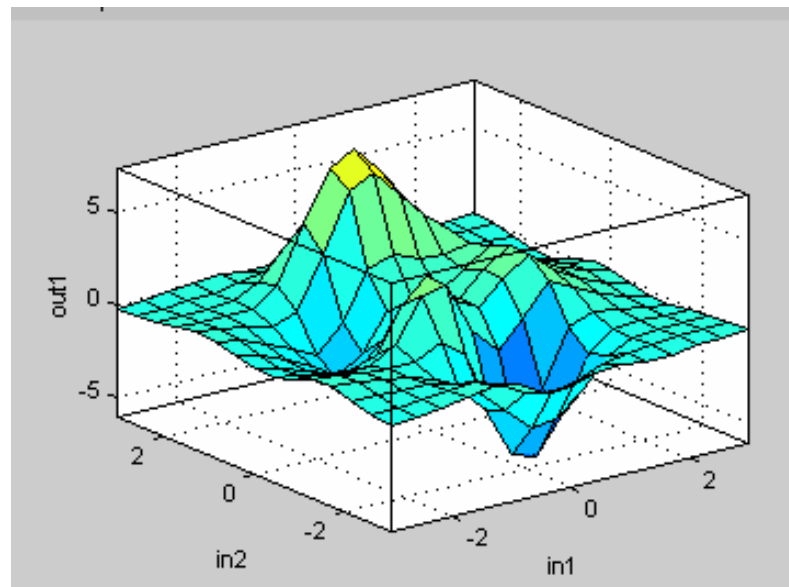
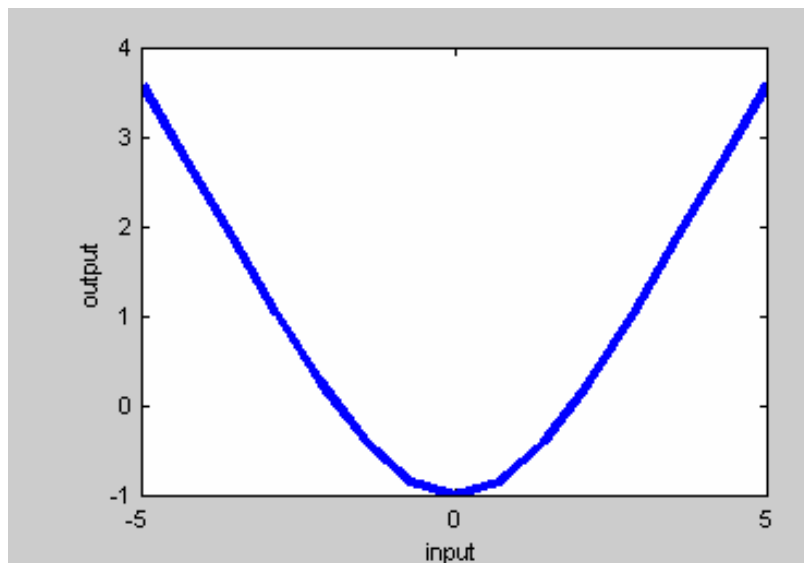
Variable	Selected	Other Options	Not
service is	poor	good, excellent, none	<input type="checkbox"/>
food is	rancid	delicious, none	<input type="checkbox"/>
tip is	cheap	average, generous, none	<input type="checkbox"/>

1. mintapélda



2. mintapélda

- Sugeno modellel függvények közelítése
 - Kétváltozós függvény (parabola)
 - Háromváltozós felület közelítés



2. mintapélda

The screenshot shows the FIS Editor interface for a Sugeno-type fuzzy inference system. At the top, a block diagram illustrates the system flow: an input variable (represented by a yellow box with two overlapping bell-shaped membership functions) feeds into a Sugeno inference block (white box labeled 'sugeno (sugeno)'), which then outputs to an output variable (represented by a cyan box labeled 'f(u)').

Below the diagram, the configuration panel shows the following settings:

- FIS Name: sugeno
- FIS Type: sugeno
- And method: prod
- Or method: probor
- Implication: min
- Aggregation: max
- Defuzzification: wtaver
- Current Variable Name: (empty)
- Current Variable Type: (empty)
- Current Variable Range: (empty)

Buttons for 'Help' and 'Close' are visible at the bottom of the configuration panel. A status bar at the very bottom indicates: "System 'sugeno': 1 input, 1 output, and 2 rules".

The screenshot displays the Membership Function Editor for the 'sugeno' system. It features a plot area with a yellow background showing two fuzzy membership functions for the input variable 'input'. The x-axis ranges from -5 to 5, and the y-axis (membership degree) ranges from 0 to 1. One function, labeled 'low', starts at 1 at x=-5 and decreases to 0 at x=5. The other function, labeled 'high', starts at 0 at x=-5 and increases to 1 at x=5. They intersect at x=0 with a membership degree of 0.5.

Below the plot, the 'Current Variable' settings are:

- Name: input
- Type: input
- Range: [-5 5]
- Display Range: [-5 5]

The 'Current Membership Function' settings are:

- Name: (empty)
- Type: trimf
- Params: (empty)

Buttons for 'Help' and 'Close' are present. A status bar at the bottom indicates: "Selected variable 'input'".

2. mintapélda

Membership Function Editor: sugeno

File Edit View

FIS Variables

input output

Membership function plots plot points: 181

line2

line1

output variable "output"

Current Variable

Name	output
Type	output
Range	[0 1]
Display Range	

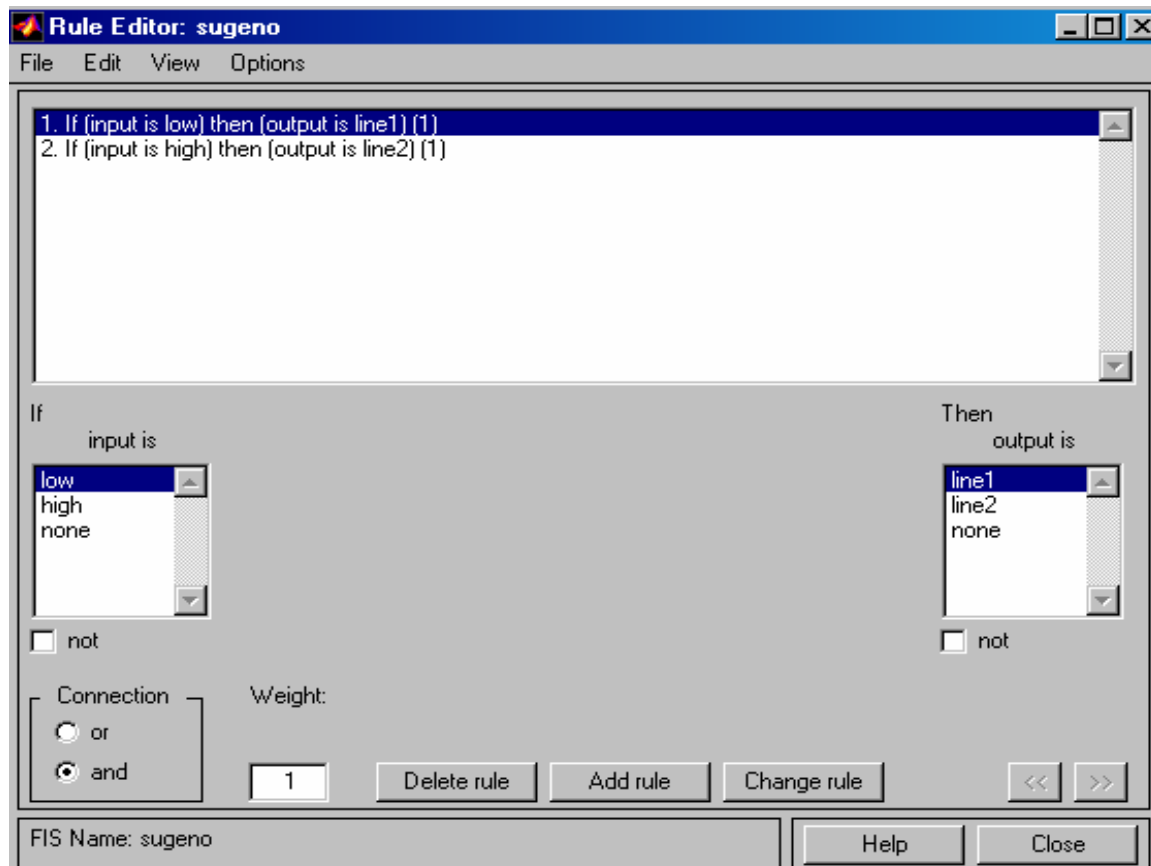
Current Membership Function (click on MF to select)

Name	line1
Type	linear
Params	[-1 -1]

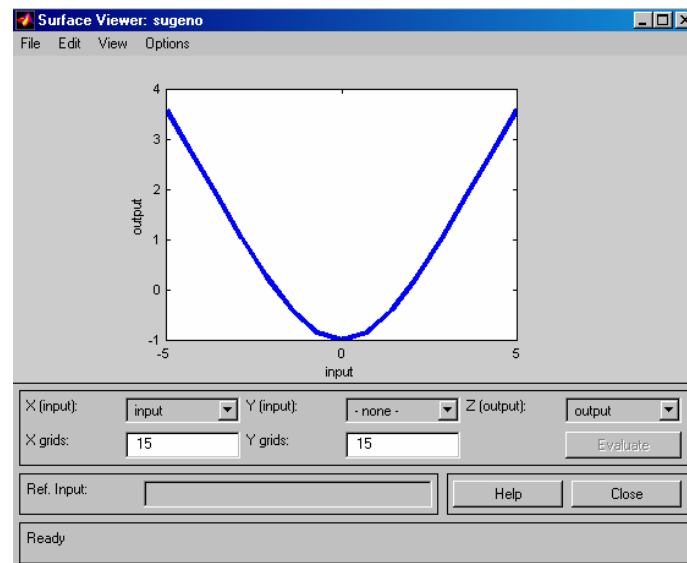
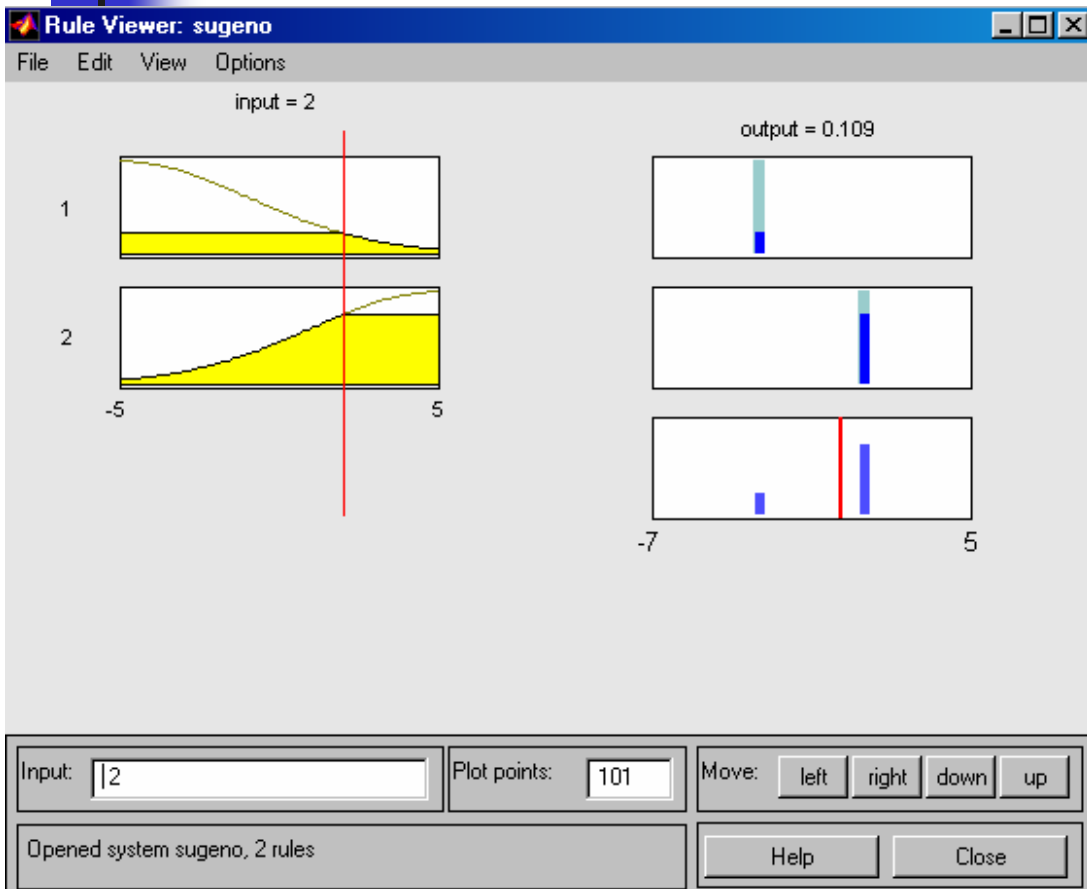
Help Close

Selected variable "output"

2. mintapélda

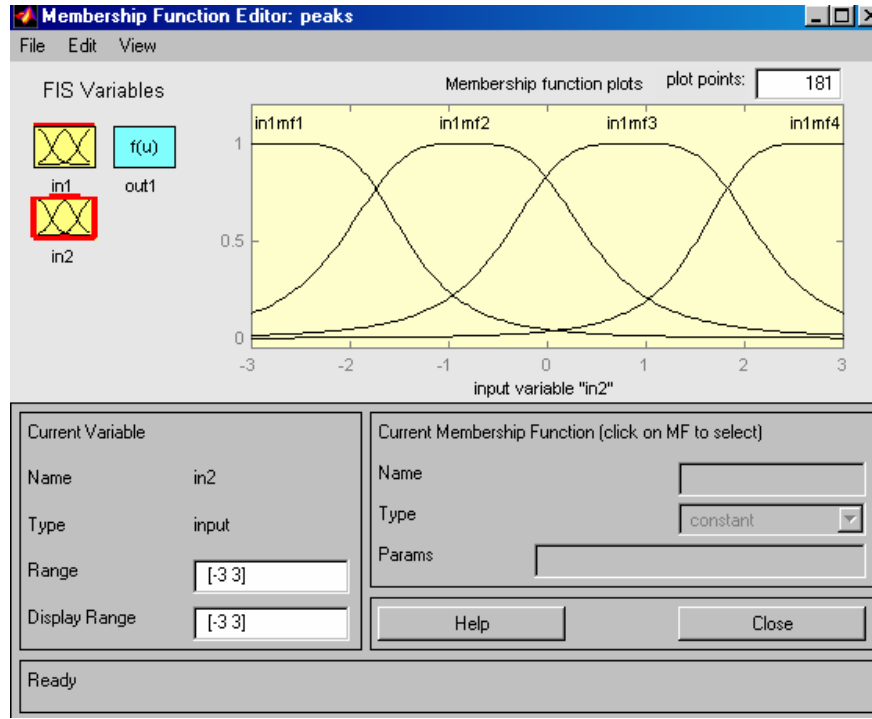
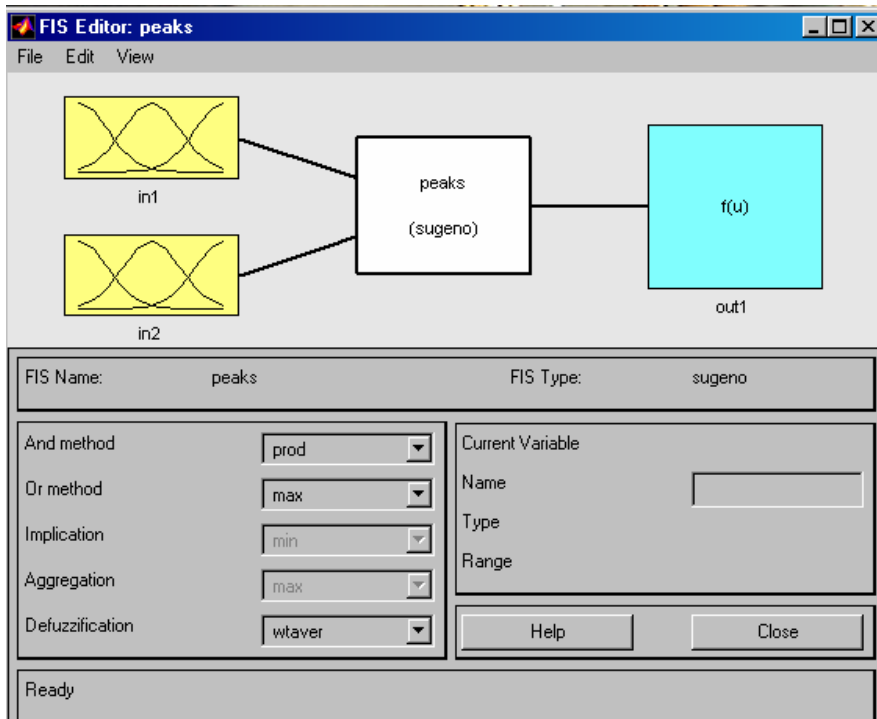


2. mintapélda



2. mintapélda

■ 3D felület:



2. mintapélda

- 3D felület:

Rule Editor: peaks

File Edit View Options

7. If [in1 is in1mf2] and [in2 is in1mf3] then [out1 is out1mf7] (1)
8. If (in1 is in1mf2) and (in2 is in1mf4) then (out1 is out1mf8) (1)
9. If (in1 is in1mf3) and (in2 is in1mf1) then (out1 is out1mf9) (1)
10. If (in1 is in1mf3) and (in2 is in1mf2) then (out1 is out1mf10) (1)
11. If (in1 is in1mf3) and (in2 is in1mf3) then (out1 is out1mf11) (1)
12. If (in1 is in1mf3) and (in2 is in1mf4) then (out1 is out1mf12) (1)
13. If (in1 is in1mf4) and (in2 is in1mf1) then (out1 is out1mf13) (1)
14. If (in1 is in1mf4) and (in2 is in1mf2) then (out1 is out1mf14) (1)
15. If (in1 is in1mf4) and (in2 is in1mf3) then (out1 is out1mf15) (1)
16. If (in1 is in1mf4) and (in2 is in1mf4) then (out1 is out1mf16) (1)

If and Then

in1 is in2 is out1 is

in1mf1
in1mf2
in1mf3
in1mf4
none

in1mf1
in1mf2
in1mf3
in1mf4
none

out1mf2
out1mf3
out1mf4
out1mf5
out1mf6
out1mf7

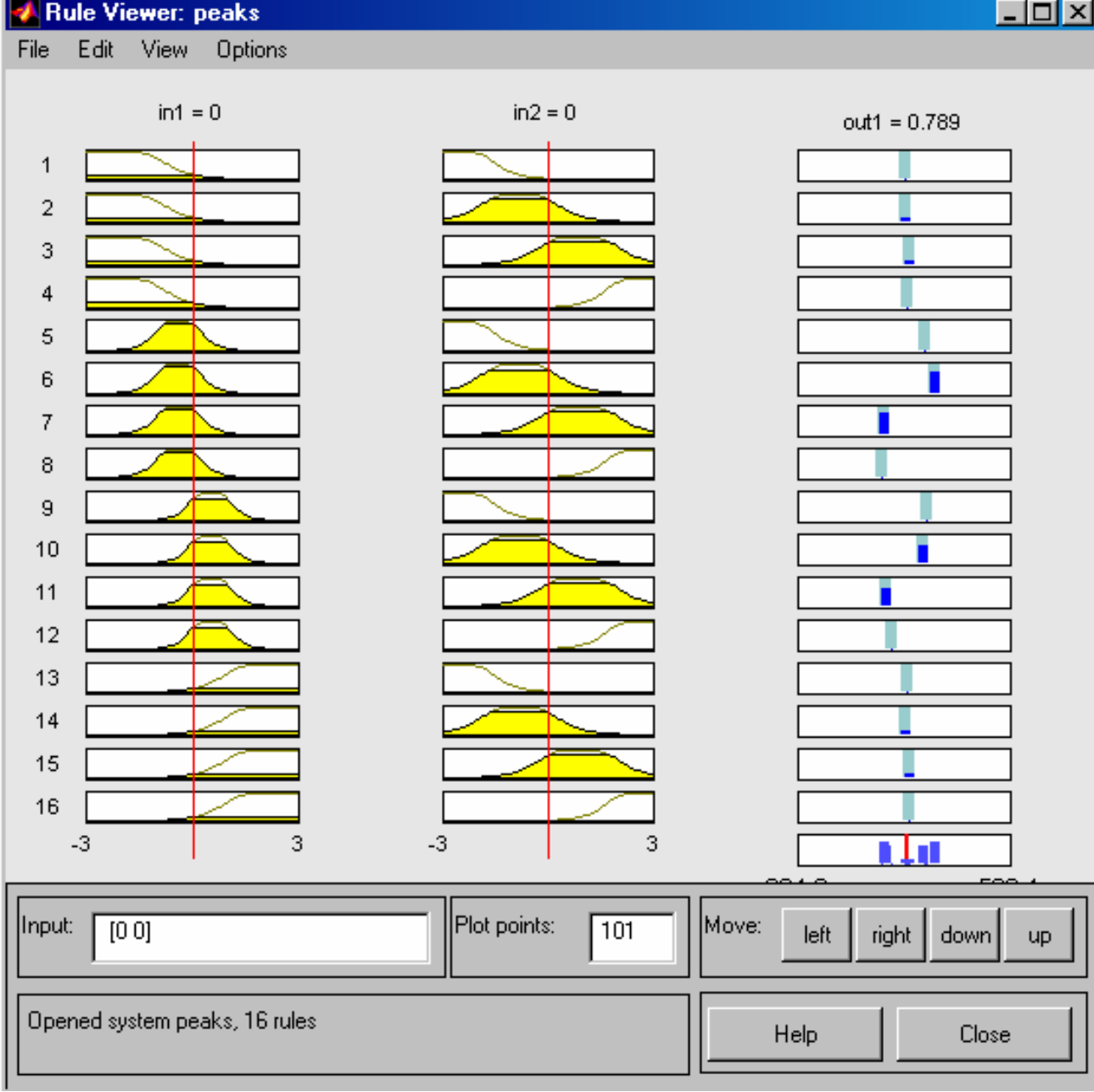
not not not

Connection
 or
 and

Weight: 1

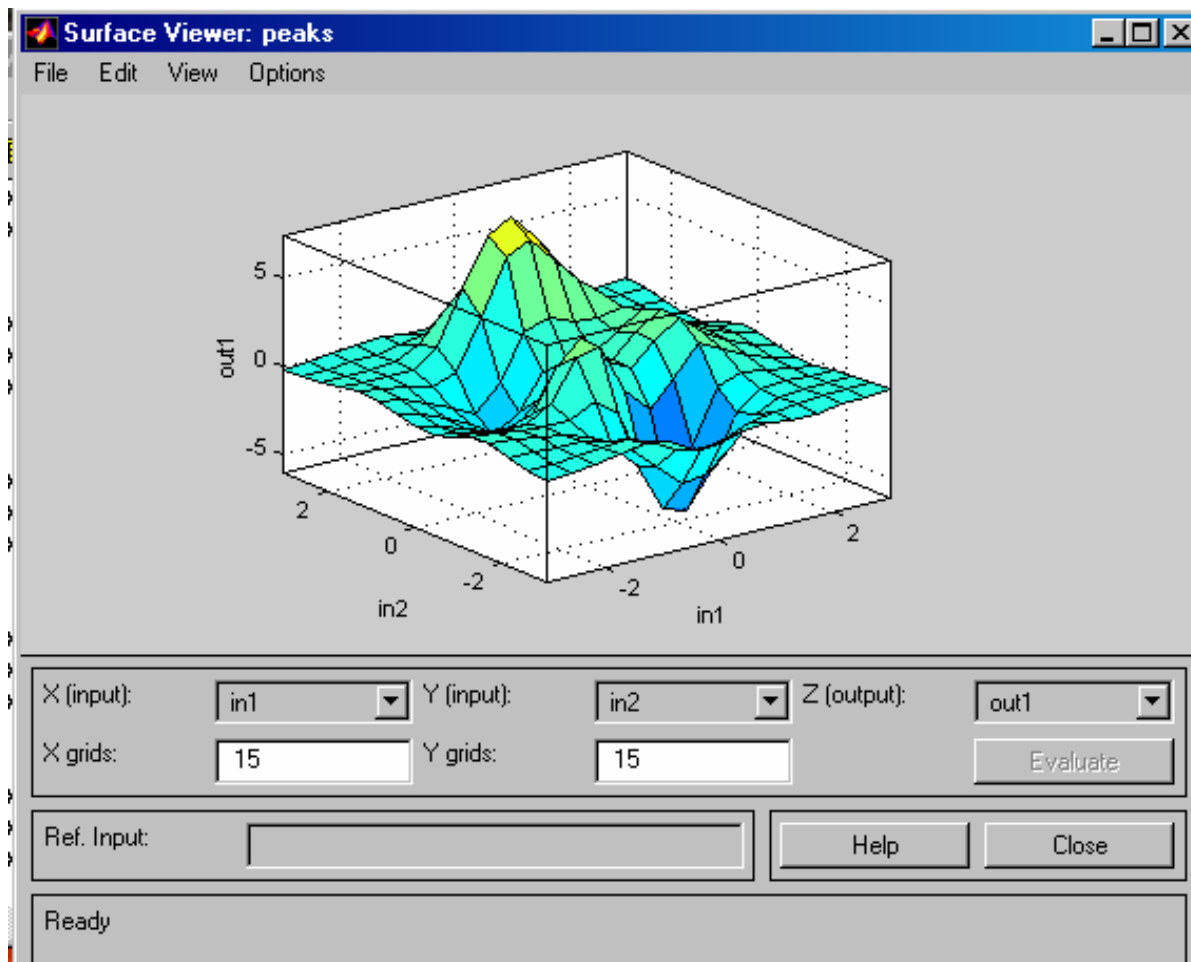
Delete rule Add rule Change rule << >>

FIS Name: peaks Help Close



2. mintapélda

- 3D felület:





Fuzzy szakértői rendszer fejlesztés

XpertRule Knowledge Builder

3-4. Gyakorló feladat (két fuzzy példa)



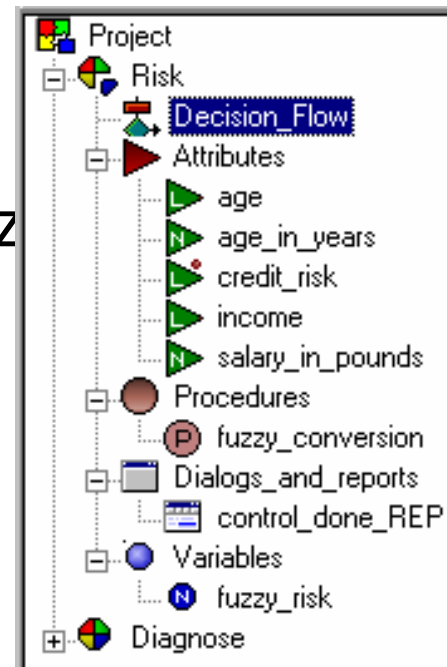
XpertRule, 3. gyakorló feladat

- A feladat: „Hitel kockázat”
 - Egy banki SZR egy személynél megállapítja a hitelezés kockázatát: alacsony, átlagos, vagy magas.
 - Döntési szempontok: életkor, jövedelem
 - Körülbelüli értékekkel dolgozik: fuzzy objektumok az életkor, jövedelem, eredmény (risk)

XpertRule, 3. gyakorló feladat

A megvalósítás lépései:

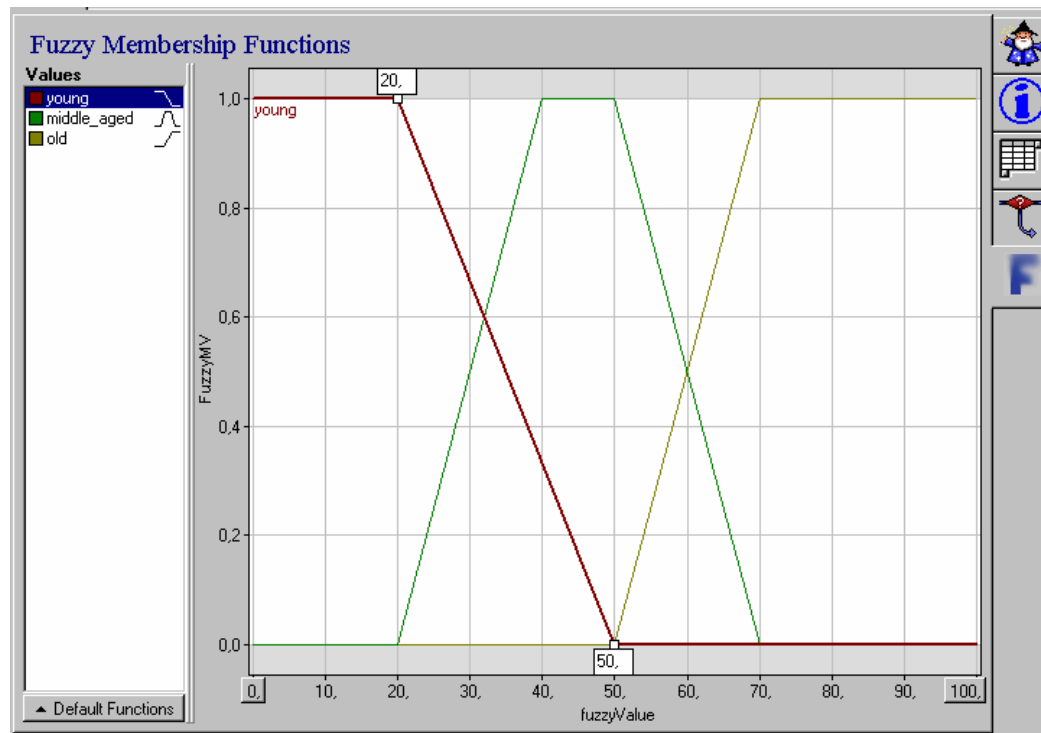
- Új projekt létrehozás: varázslóval
Access adatbázis hozzárendelés
(felhasználói névvel, jelszóval)
- Új tudás modul definiálás (itt: hitel kockáz
- Attribútumok definiálása:
életkor_év (numerikus), életkor (fuzzy),
jövedelem (fuzzy), jövedelem_Ft (num.),
hitel_kockázat (fuzzy).



XpertRule, 3. gyakorló feladat

A megvalósítás lépései:

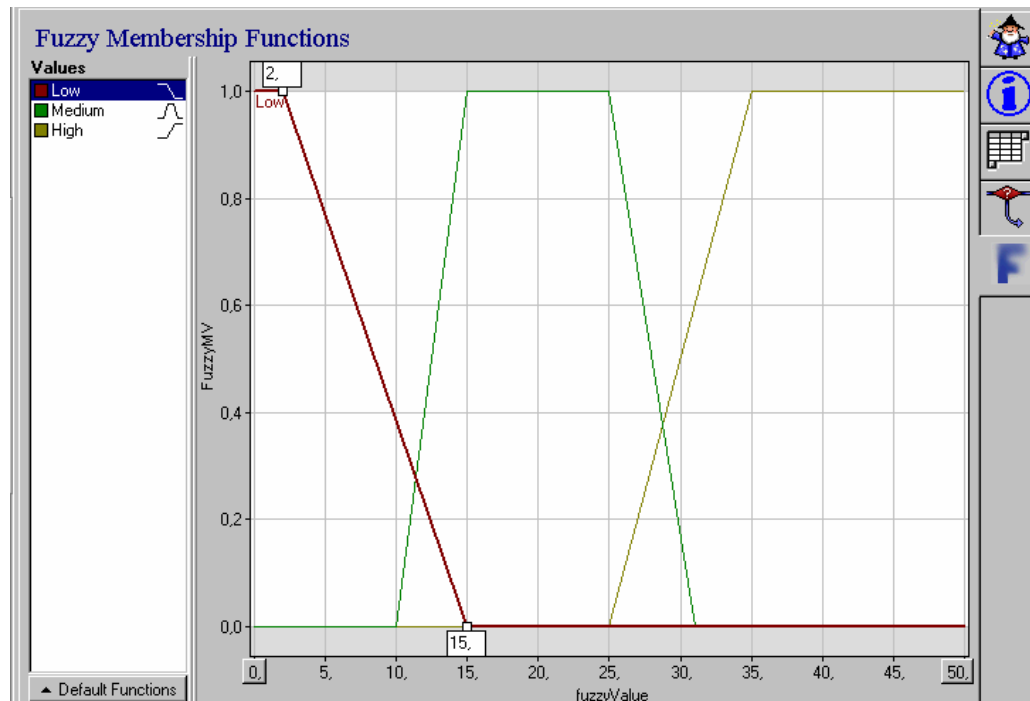
- Attribútum értékek definiálása (Ins. Property):
 - Életkor (fuzzy)



XpertRule, 3. gyakorló feladat

A megvalósítás lépései:

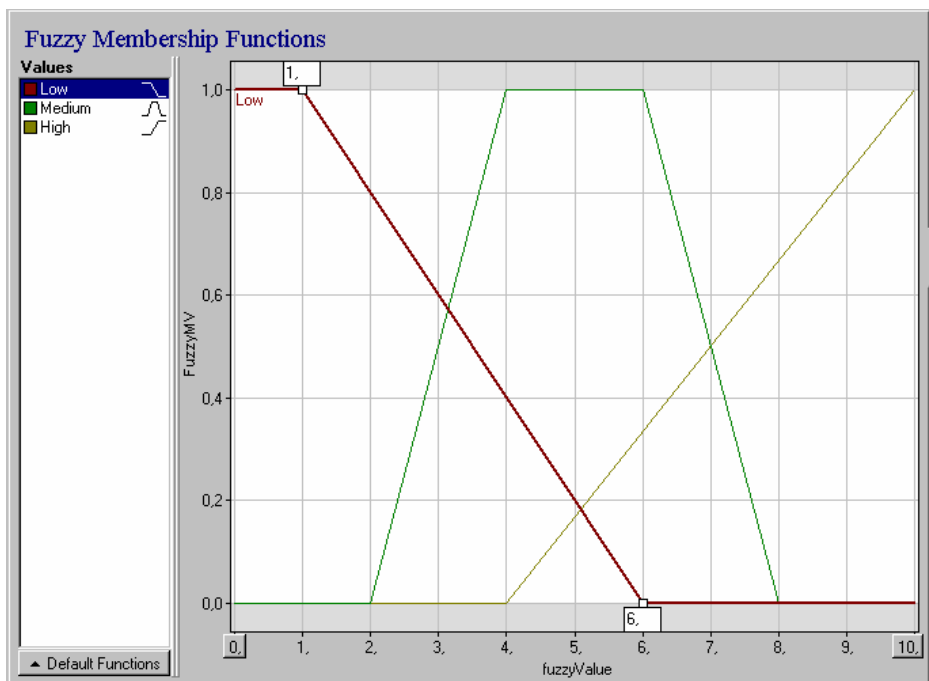
- Attribútum értékek definiálása (Ins. Property):
 - Jövedelem (fuzzy)



XpertRule, 3. gyakorló feladat

A megvalósítás lépései:

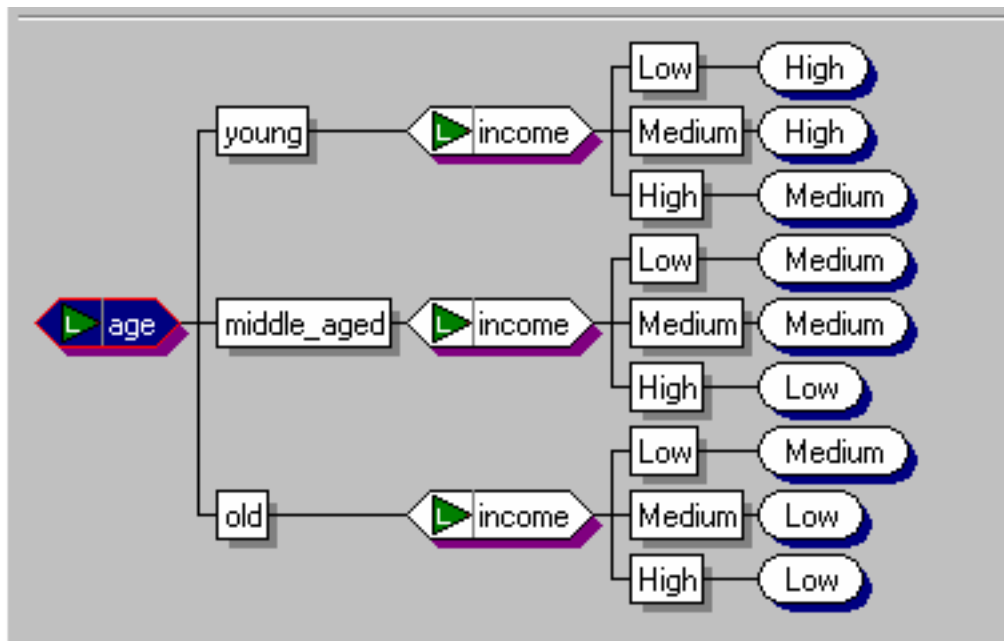
- Attribútum értékek definiálása (Ins. Property):
 - hitel_kockázat (fuzzy)



XpertRule, 3. gyakorló feladat

A megvalósítás lépései:

- Attribútum értékek definiálása (Ins. Property):
 - hitel_kockázat (fuzzy)+ „tudás” (döntési fa):



XpertRule, 3. gyakorló feladat

A megvalósítás lépései:

- A végrehajtás szabályozása (főprogram):

input kor - fuzzy értéke

input fizetés – fuzzy ért.

fuzzy hitel_kockázat ért.

és konvertálása numer.

+ riport

Main Agenda

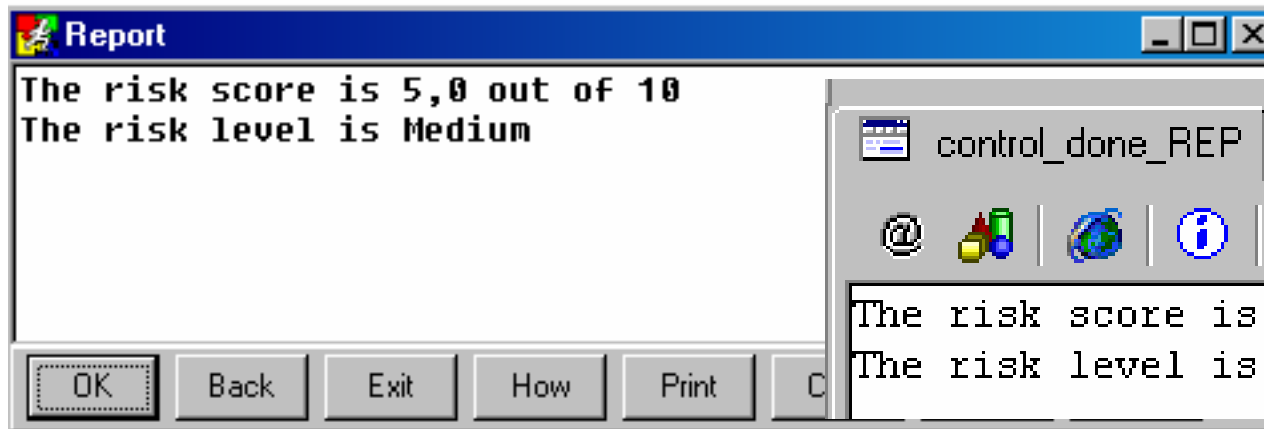
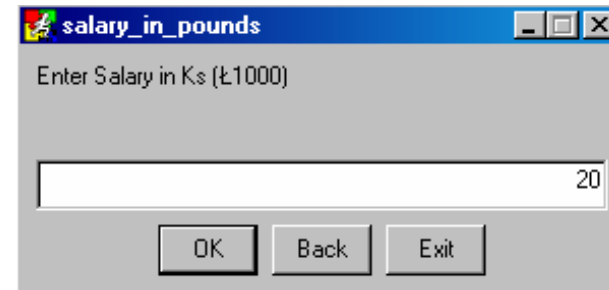
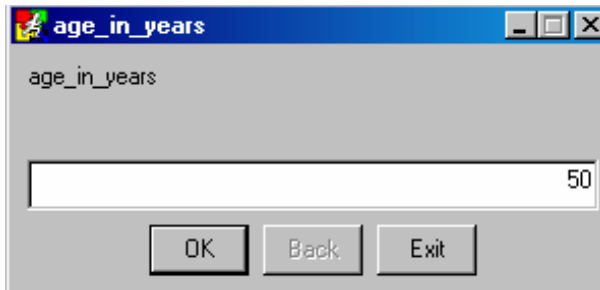


VBScript

```
@Do age_in_years
@Assign age.fuzzyValue = age_in_years
@Do salary_in_pounds
@Assign income.fuzzyValue = salary_in_pou
@Do credit_risk
@Do fuzzy_conversion
@Do control_done_REP
```

XpertRule, 3. gyakorló feladat

A futtatás lépései, megjelenő kérdések és riport pl.:





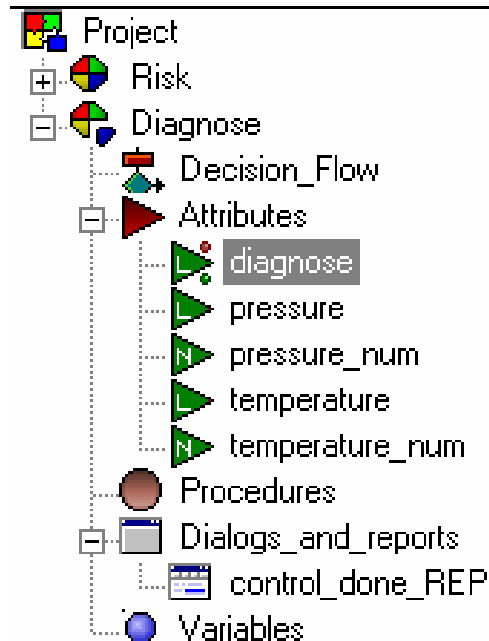
XpertRule, 4. gyakorló feladat

- A feladat: „Kazán diagnosztika”
 - A SZR három gyakori hiba „valószínűségét” prognosztizálja (vagy azt, hogy nincs hiba)
 - Döntési szempontok: nyomás, hőmérséklet
 - Körülbelüli értékekkel dolgozik: fuzzy objektumok a nyomás, a hőmérséklet
 - Diagnosztikai esetek: a fuzzy szabályok alapjai

XpertRule, 4. gyakorló feladat

A megvalósítás lépései:

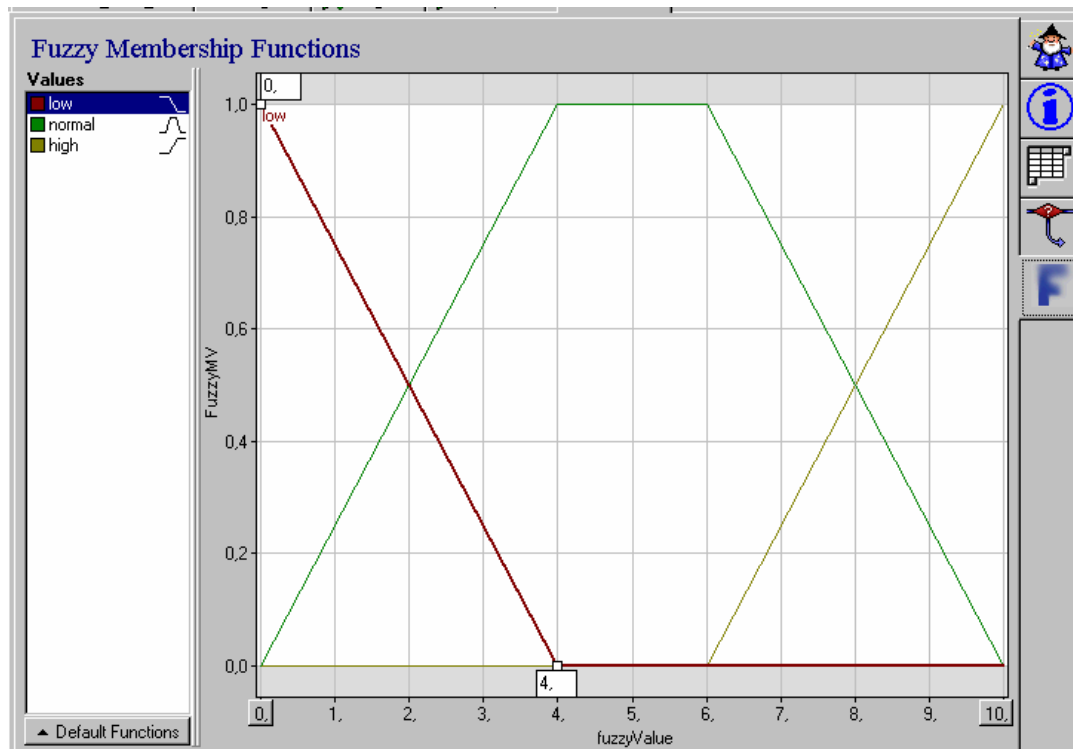
- Új projekt létrehozás: varázslóval
Access adatbázis hozzárendelés
(felhasználói névvel, jelszóval)
- Új tudás modul definiálás
(itt: kazán diagnosztika)
- Attribútumok definiálása:
nyomás (fuzzy)- és num.
hőmérséklet (fuzzy) - és num.
diagnózis (list+esetek)



XpertRule, 4. gyakorló feladat

A megvalósítás lépései:

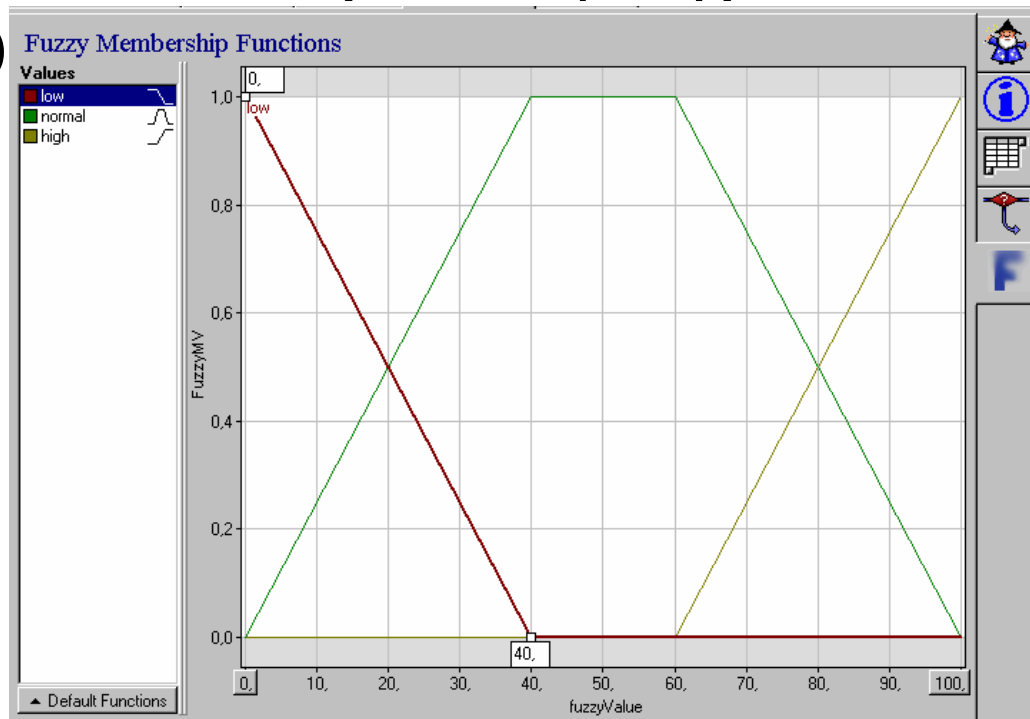
- Attribútum értékek definiálása (Ins. Property):
 - nyomás (fuzzy)



XpertRule, 4. gyakorló feladat

A megvalósítás lépései:

- Attribútum értékek definiálása (Ins. Property):
 - hőmérséklet (fuzzy)



XpertRule, 4. gyakorló feladat

A megvalósítás lépései:

- Attribútum értékek definiálása (Ins. Prop.)
 - Diagnózis: list + esetek –fuzzy szabályok

Cases

	temperature	pressure	diagnose	+
1	low	low	fault1	
2	high	high	fault3	
3	low	normal	fault2	
4	normal	low	fault2	

Rules View

1	If	temperature = low	And	pressure = low	Then	diagnose = fault1
2	If	temperature = high	And	pressure = high	Then	diagnose = fault3
3	If	temperature = low	And	pressure = normal	Then	diagnose = fault2
4	If	temperature = normal	And	pressure = low	Then	diagnose = fault2

Instance Properties

	Value	+
1	fault1	
2	fault2	
3	fault3	

Objects

- ▶ temperature
- ▶ pressure

XpertRule, 4. gyakorló feladat

A megvalósítás lépései:

- A végrehajtás szabályozása (főprogram):

Input nyomás,
hőmérséklet,
fuzzy értékek,
diagnózis értéke,
+riport

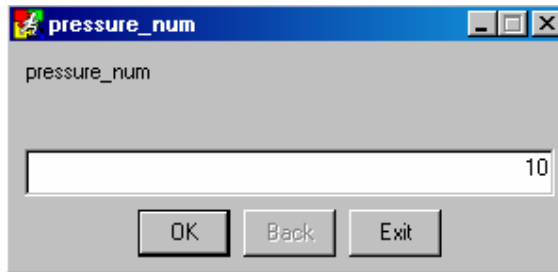
Main Agenda

@    VBScript

```
@Do pressure_num
@Do temperature_num
@Assign pressure.fuzzyValue = pressure_num
@Assign temperature.fuzzyValue = temperature_num
@Do diagnose
@Do control_done_REP
```

XpertRule, 4. gyakorló feladat

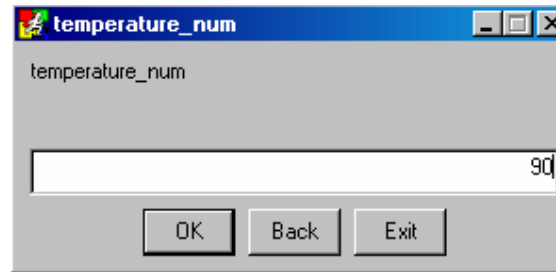
A futtatás lépései, megjelenő kérdések és riport pl.:



pressure_num

10

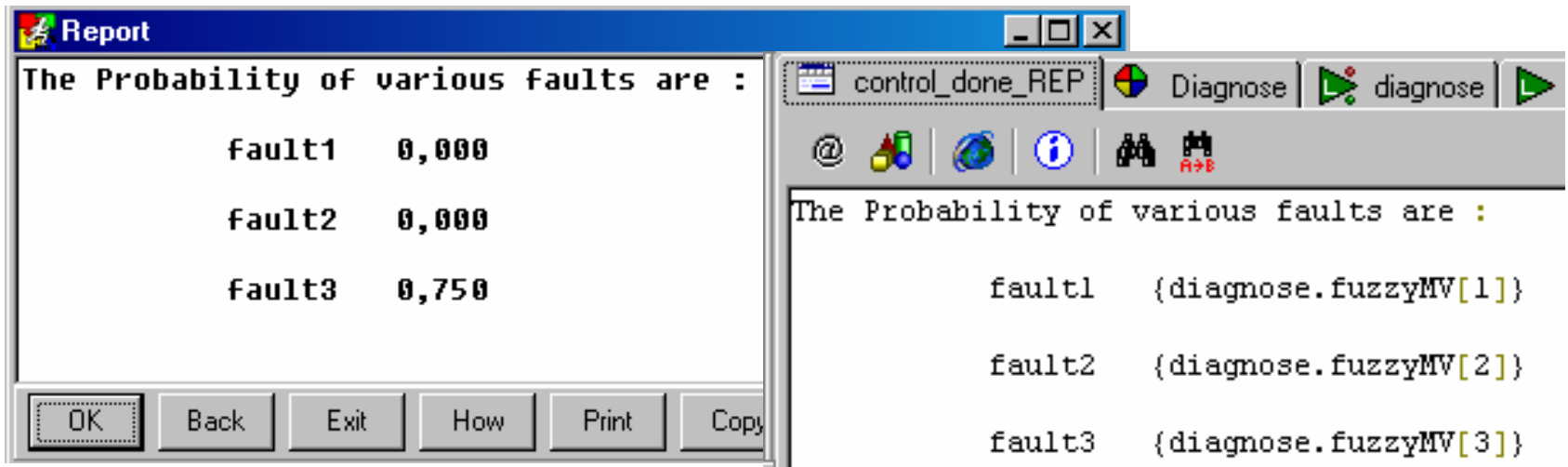
OK Back Exit



temperature_num

90

OK Back Exit



Report

The Probability of various faults are :

fault1	0,000
fault2	0,000
fault3	0,750

OK Back Exit How Print Copy

control_done_REP Diagnose diagnose

The Probability of various faults are :

fault1	{diagnose.fuzzyMV[1]}
fault2	{diagnose.fuzzyMV[2]}
fault3	{diagnose.fuzzyMV[3]}



Fuzzy-rendszerek

Alkalmazások I.

Fuzzy rendszerek alkalmazása



- **Gyakori alkalmazások** (közlekedés, autóipar, háztartási elektronika, ipari robot, gazd. élet,)
- **Problématípusok**
 - szabályozás (ipari)
 - közelítő köv. (fuzzy SZR)
 - döntések fuzzy környezetben
 - adatanalízis (osztályozás, klaszterképzés)
 - információ visszakeresés (adatbázisból)
 - optimalizálás (fuzzy aritmetika, ...)
 - képfeldolgozás, ...

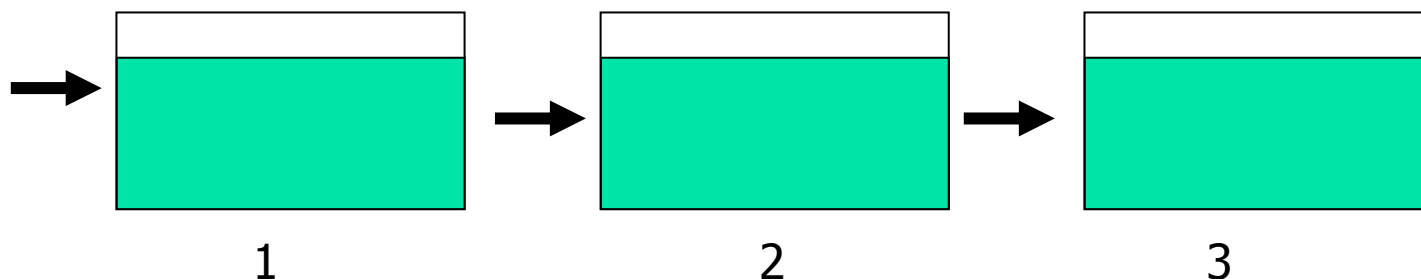
Fuzzy rendszerek alkalmazása

Szabályozás

- **Fuzzy szabályozó:** ipari szab. -(univ. függv. köz.)
- **Ipari példák:** (gőzgép, cementégető, metró, házt. gépek, szennyvíztisztító, blokkolásgátló, ...)
- **Víztisztító példája**
folyóvízből ivóvíz
3 tartály, 3-5 órás kezelés tartályonként

Fuzzy rendszerek alkalmazása

- **Vízisztító példája**
folyóvíz



Hipotézis: 1. tartály T1 vízmennyiségét kell szabályozni.
Jellemzők: AL(lúgosság), PH, TE (hőm.), kétféle szennyezettség fok (SZ1 - SZ2)
(fuzzy termek: kicsi, nagy minden változónál)

Fuzzy rendszerek alkalmazása

- **Vízisztító példája**
Sugeno-modell 8 szabállyal

R_i : IF A is x_1 AND B is x_2 AND C is x_3

THEN $T_1 = p_0 + p_1 * x_4 + p_2 * x_5 + p_3 * x_1 + p_4 * x_2 + p_5 * x_3$.

(ahol x_1, x_2, \dots, x_5 : PH, AL, TE, SZ1, SZ2)

p_i : becslés, függv. illesztés (40 elemű példasor ismert)

R_i	PH	AL	TE	p_0	p_1	p_2	p_3	p_4	p_5
1	K	K	K	8858	2664	-8093	11230	-1147	-2218
2	K	K	N	-7484	124	-427	761	52	-17

....

Fuzzy rendszerek alkalmazása

Döntések fuzzy környezetben

- **Döntéshozatal:** alapprobléma, többkrit. döntések
- **MCDM lépései:**
 1. probléma def.,
 2. Kritériumok
 3. Alternatíva - krit. kapcsolat (mátrix, súlyok)
 4. Aggregációs elj. , rendezés
(preferencia sorrend, súlyozás...)
eredmény: kiválasztás, osztályozás, rendezés
- **Fuzzy módszerek 3-4-ben**

Fuzzy rendszerek alkalmazása

- **Rendező módszerek**
- Yager "max-min" módszere

$$A = \{a_1, a_2, \dots, a_n\}, \quad K = \{k_1, k_2, \dots, k_m\}, \\ g_1, g_2, \dots, g_m \quad (\sum g_i = m)$$

$\mu_{kj}(a_i)$: milyen jó az a_i alternatíva a k_j szempontjából

Számolási lépések:

$$\sim\mu_{kj}(a_i) = [\mu_{kj}(a_i)]^{g_j} \quad \text{minden } a_i \in A\text{-ra.}$$

Legyen D a döntési tér:

$$\mu_D(a_i) = \min_{j=1,2,\dots,m} \sim\mu_{kj}(a_i) \quad (\text{aggregálás min művelettel})$$

$$\text{eredmény: } \mu_D(a^*) = \max_{a_i \in A} \mu_D(a_i)$$

Fuzzy rendszerek alkalmazása

■ Példa Yager módszerére

	k1	k2	k3	k4
a1	0.7	0.3	0.2	0.5
a2	0.5	0.8	0.3	0.1
a3	0.4	0.6	0.8	0.2

g1=2.32, g2=1.2, g3=0.32, g4=0.16, pl. $\mu_{k3}(a_2)=0.3$

$$\mu_D(\mathbf{a1}) = \min \sim \mu_{kj}(\mathbf{a1}) = \min \{0.44, 0.24, 0.6, 0.9\} = 0.24.$$

$$\mu_D(\mathbf{a2}) = \min \sim \mu_{kj}(\mathbf{a2}) = \min \{0.2, 0.76, 0.68, 0.69\} = 0.2$$

$$\mu_D(\mathbf{a3}) = \min \sim \mu_{kj}(\mathbf{a3}) = \min \{0.12, 0.54, 0.93, 0.72\} = 0.12$$

Az optimális megoldás:

$$\mu_D(\mathbf{a}) = \max \{ \mu_D(\mathbf{a1}), \mu_D(\mathbf{a2}), \mu_D(\mathbf{a3}) \} = 0.24 = \mu_D(\mathbf{a1}) \quad \mathbf{a} \in \mathbf{A}.$$

Fuzzy rendszerek alkalmazása

- **Még egy rendező módszer: "Osztályozó m."**

$A = \{a_1, a_2, \dots, a_n\}$, $K = \{k_1, k_2, \dots, k_m\}$,

$k_j = \{S_{j1}, \dots, S_{jpj}\}$, ahol S_{j1}, \dots, S_{jpj} nyelvi vált. értékek,
 g_1, g_2, \dots, g_m súlyok

(g_1^2) IF k_1 is S_{11} THEN $E = S_{11}$

(g_1^2) IF k_1 is S_{12} THEN $E = S_{12}$

.....

(g_j^2) IF k_j is S_{js} THEN $E = S_{js}$

.....

(g_m^2) IF k_m is S_{mpn} THEN $E = S_{mpm}$

$(s=1,2, \dots, p_j)$ $(j = 1, 2, \dots, m)$

Fuzzy rendszerek alkalmazása

- **Még egy rendező módszer: “Osztályozó m.”**

szabályozó: Σp_{ji} darab szabály

minden a_i -hez egy érték (“osztályzat”)

rendezi az alternatívákat

(eltérés a Yager max-min-től: nyelvi változó értékeket alkalmaz)

- **Mintapélda: Po-delta nemzeti park hasznosításának problémája.**

Fuzzy rendszerek alkalmazása



Alternatívák

- a. üzleti élet optimalizálása általában**
- b. mezőgazdaság optimalizálása**
- c. nagyobb terület vízzel elárasztása**
- d. részlegesen területek vízzel elárasztása, megtartva a jelenlegi mezőgazdasági termelést**
- e. részlegesen területek vízzel elárasztása, optimalizálva a mezőgazdasági termelést.**

Fuzzy rendszerek alkalmazása

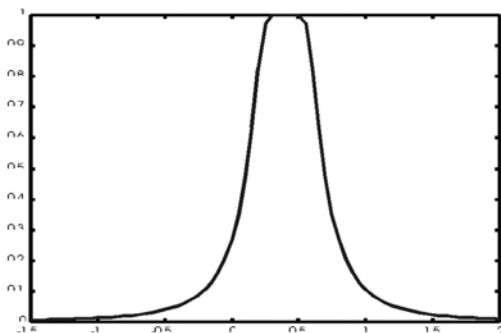


Kritériumok

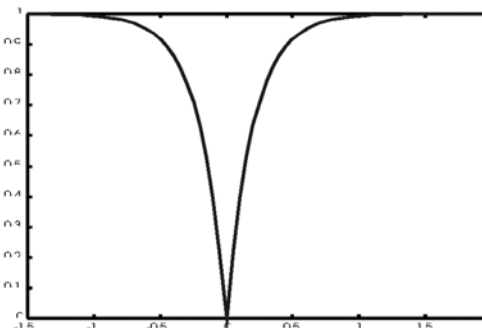
1. **nagy nyereség;**
2. **foglalkoztatás növelése;**
3. **turisztikai vonzerő növelése;**
4. **pihenésre, szórakozásra vonzerő növelés;**
5. **a táj ökológiai egyensúlya,**
6. **az ökológiai károk okozta veszély csökkentése.**

Fuzzy rendszerek alkalmazása

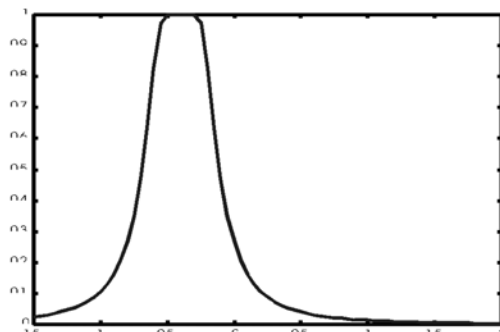
Tartalmazási függvények:



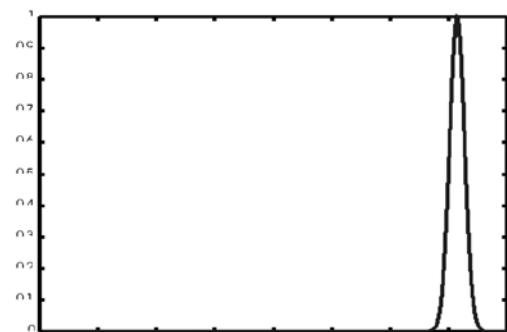
jó



mérsékelt



rossz



közelítőleg 143 mill.

Fuzzy rendszerek alkalmazása

- Adatok:

	a	b	c	d	e
k1	64. m.	159.m.	közel. 143.m.	közel. 95.m.	közel. 147.m.
k2	8	20	9	8	14
k3	rossz	rossz	jó	mérsékelt	mérsékelt
k4	mérs.	mérsékelt	jó	mérsékelt	mérsékelt
k5	rossz	rossz	jó	jó	mérsékelt
k6	mérs.	rossz	jó	rossz	rossz

- eredmény: $e > c > b > d > a$
lobbik hatása: c.

Fuzzy rendszerek alkalmazása

Függvényközelítés, prognózis

- **Példa: folyó árhullám előrejelzés (6 órával, Mosel)**
- **Adott: $d(t)$ vízmennyiség idősorok (11 áradás adatai)**
- **NH modell: vízmennyiség változás idősor időablak (n adat, n+1-dik becslése)
backpropagation modell (20-10-10-1)
csak átlagosan jó**

Fuzzy rendszerek alkalmazása

Függvényközelítés, prognózis

- **FR modell**

szakértői vélemény: $d_o(t) = d_o(t-6) + \Delta d_o(t)$,

$\Delta d_o(t) = f(\Delta d_1(t - \Delta t_1), \Delta d_2(t - \Delta t_2), \dots, \Delta d_n(t - \Delta t_n))$,

**ahol Δd_i az i-dik idősor vízmennyiség változása
3, 6 és 9 óránként.**

Sugeno modell $d_o(t)$ -re

Fuzzy rendszerek alkalmazása

Függvényközelítés, prognózis

- Sugeno modell $d_o(t)$ -re

R_i : if x_1 is A_{i1} and ... x_n is A_{in} then

$$y_i = p_{i0} + p_{i1} * x_1 + \dots + p_{in} * x_n$$

- Input tér particionálás: klaszterezéssel A_i -k
 - Külön szabályok minden idősorra (folyóra)
(p_0, p_1, \dots, p_n optimalizálása függvény illesztéssel)
 - Δt_i optimalizálása
- **Eredmény: jobb mint a szakértőké**



Fuzzy-rendszerek

Alkalmazások II.

Fuzzy rendszerek alkalmazása



- **Adatanalízis**
- **Fuzzy klaszteranalízis**
- **Fuzzy osztályozás**
- **Alkalmazási példa: függvényközelítés**

Fuzzy rendszerek alkalmazása

Adatanalízis

Lépések:

- **1. Gyakoriságok, szelektálás**
- **2. Mintafelismerés**
- **3. mat. modellezés (funk. kapcsolatok)**
- **4. Elemzés, értékelés**

Pl. 2. lépcső: Fuzzy klaszteranalízis

Pl. 3. lépcső: Fuzzy osztályozás, fuzzy shell klaszter algoritmusok, szabályfelismerés

Fuzzy rendszerek alkalmazása

Fuzzy klaszteranalízis

- Fuzzy c-Means-algoritmus

$o(\mathbf{x}, \mathbf{u}, \mathbf{v}) = \sum \sum u_{ik}^m d(\mathbf{v}_k, \mathbf{x}_i)^2$ minimum keresés

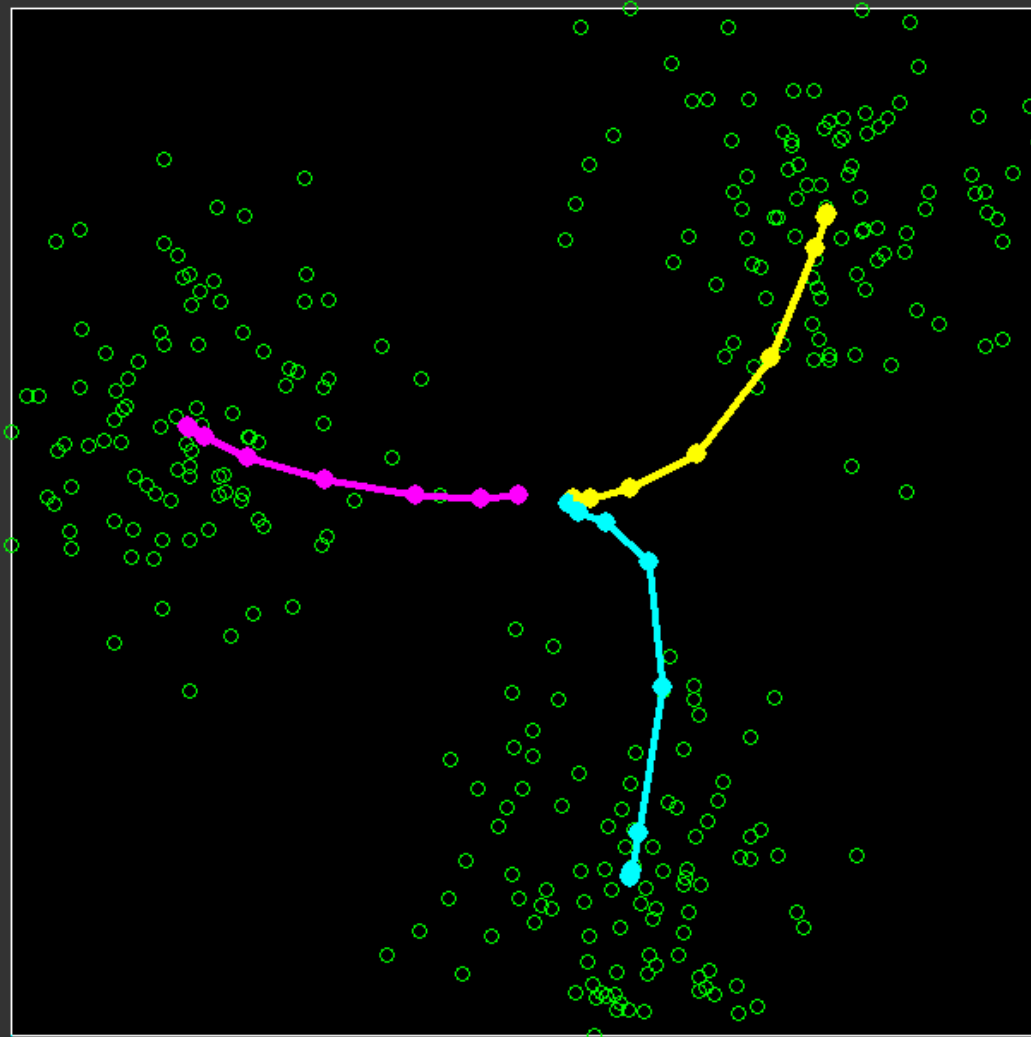
$$\sum u_{ik} > 0 \quad (k=1, \dots, n) \quad \sum u_{ik} = 1 \quad (i=1, \dots, c)$$

$$u_{ik} = 1 / (\sum (d(\mathbf{v}_i, \mathbf{x}_k) / d(\mathbf{v}_j, \mathbf{x}_k))^{1/(m-1)})$$

$$\mathbf{v}_i = \sum (u_i^m \mathbf{x}_i) / (\sum u_{ik}^m)$$

megállás: u_{ik} stabil

- Gustafson-Kessel alg., lineáris klaszter eljár., shell klaszter eljárások....



Data Set 2

3 Clusters

Label Data

Clear Traj.

MF Plot

Start

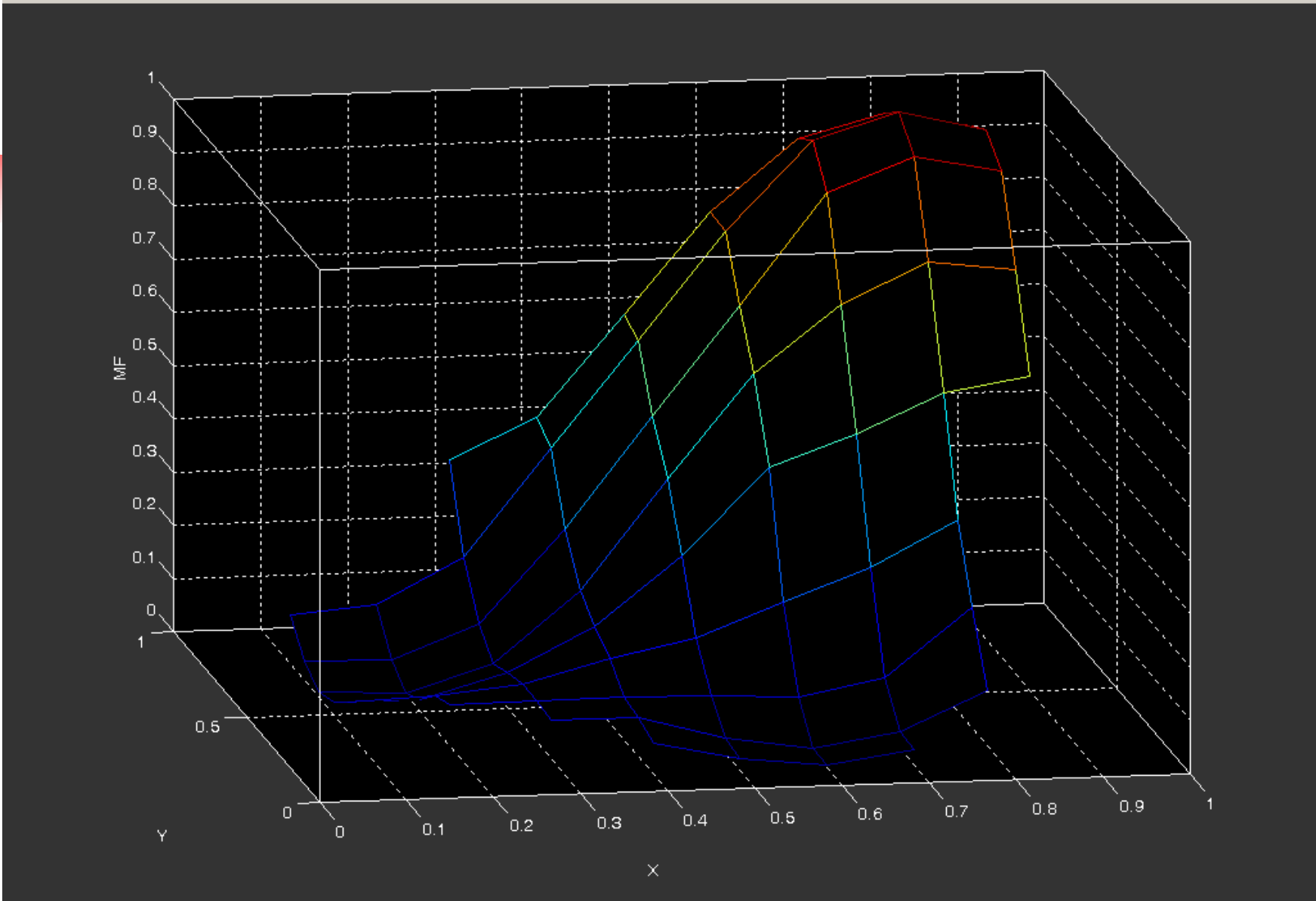
Expo.: 2

Iterat.: 100

Improv.: 1e-005

Help

Close



Fuzzy rendszerek alkalmazása

Fuzzy-osztályozás

Kitekintés: minta osztályozás - csoportosítás

- tudásábrázolás (explicit, implicit)
- tanulási mód (supervised, unsup.)
- minta tulajdonság (numerikus, heterogén)

Tudásábrázolás

Implicit: statisztika (diszkrim fg., kovariancia m., hasonlóság , val. érték) Bayes régió, potenciál fg.,NH

F(minta) →osztály tartalmazási fg érték.

Fuzzy rendszerek alkalmazása

Fuzzy-osztályozás

Tudásábrázolás Explicit:

If- then.., frame, log. kifejezések.

**Tudásalapú: szabályalapú (szabály – osztály kapcs.)
hierarchikus osztályozók**

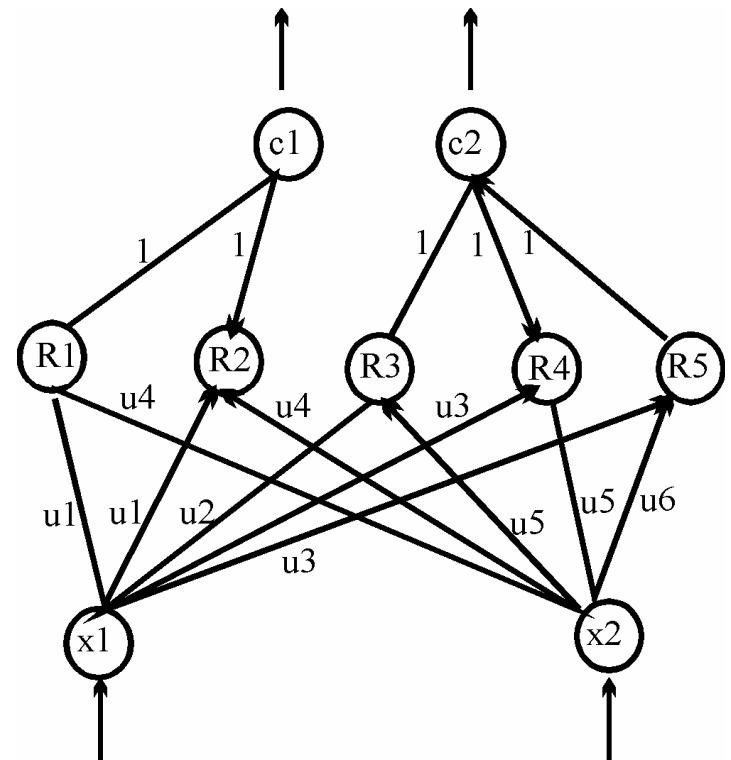
Optimalizálás - tanítás explicit esetről

- **evolúciós algoritmusok segítségével**
- **NH közreműködéssel**
- **fuzzy NH-val előállítás**
- **neurofuzzy rendszerrel**
- **Tanulóalgoritmussal finomítás**

Fuzzy rendszerek alkalmazása

NEFCLASS: egy neurofuzzy osztályozó

Pl. 2 input adat
x1 (u1,u2,u3)
x2 (u4,u5,u6)
c1, c2 osztályok



Fuzzy rendszerek alkalmazása

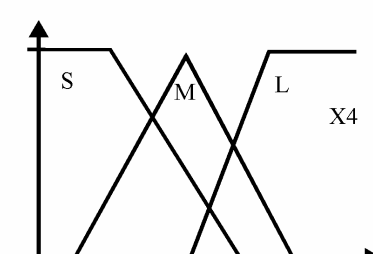
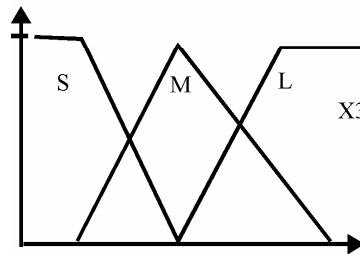
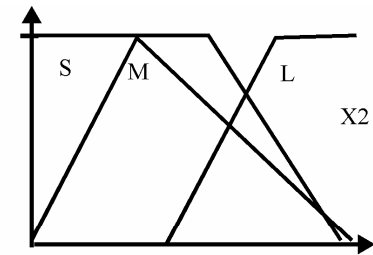
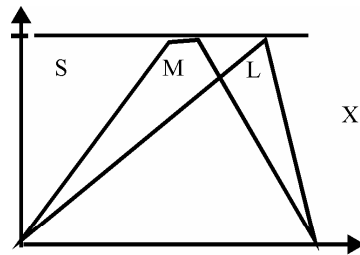
IRIS adatsor: a generált szabályok a következők:

R1: IF X1 is S AND X2 is M AND X3 is S AND X4 is S THEN osztály is 1;

R2: IF X1 is L AND X2 is M AND X3 is M AND X4 is M THEN osztály is 2;

...

R7: ...



Fuzzy rendszerek alkalmazása



Fuzzy szabályozó kialakítás adatok alapján

Yager-Filev módszere

- **mountain clustering eljárás**
- **két lépcső: induló szabályhalmaz
finomítás (optimalizálás)**
- **közelítési forma: bázisfüggvényekkel
(gauss)**

Fuzzy rendszerek alkalmazása

Fuzzy szabályozó kialakítás adatok alapján

- **Mountain Clustering algoritmus (Yager-Filev)**
 1. **2 dim. rács (térbeli): N_{ij} lehetséges centrumok**
 2. **M_i mountain fg. N_{ij} -ben**
$$M_1(N_{ij}) = \sum e^{(-a * d(N_{ki}, N_{jk}))}$$
 (a konstans)
 3. **Centrum választás: $M_1^* = \max M_1(N_{ij})$, (N_{jj}^* a rácspont)**
$$M_2(N_i) = M_1(N_i) - M_1^* e^{(-b * d(N_j, N_k))}$$

$$M_2 \rightarrow M_1$$

Vége ha $M_1^* < \delta$, különben újabb centrum választás
- Klaszterek: a kiválasztott centrumok**

Fuzzy rendszerek alkalmazása

Fuzzy szabályozó kialakítás adatok alapján Yager-Filev módszere

Input: x_1, x_2, \dots, x_n, y

minden klaszterre egy szabály:

IF x_1 is B_{r1} AND ... AND x_n is B_{rn} THEN y is D_r

$r=1,2,\dots,m$ és $(c_{r1}, c_{r2}, \dots, c_{rn}, c_r)$ az r -dik centrum

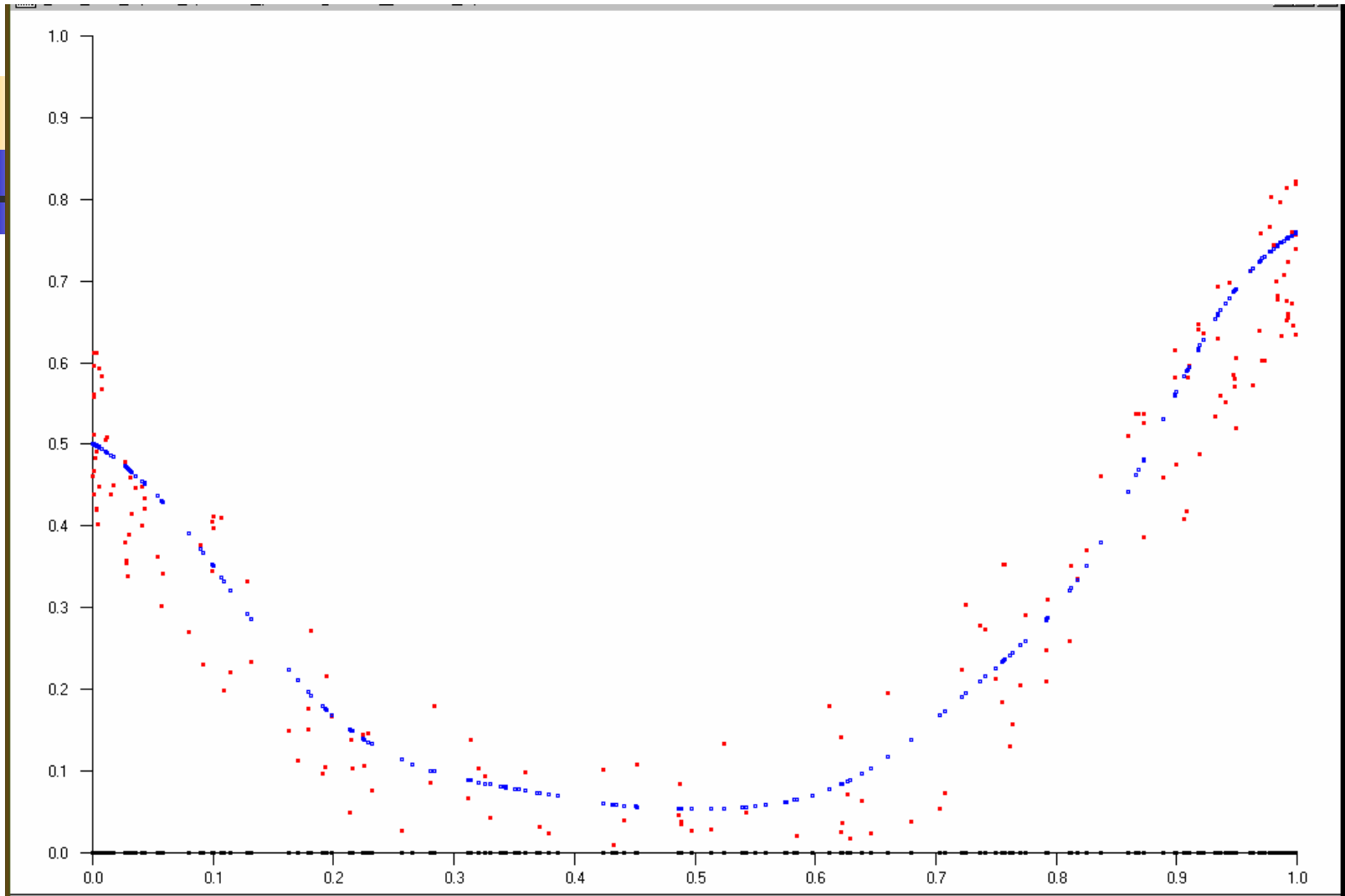
$$B_{ri}(x_{ij}) = \exp(-1/\sigma_{ri}^2 (x_{ij}-c_{ri})^2)$$

$$D_r(y) = \exp(-1/\sigma^2 (y-c_r)^2)$$

y -ra kifejezve egy adott input x_{1k}, \dots, x_{rk} esetén:

$$Y = \frac{(\sum_m c_i^* \exp(-\sum_r 1/\sigma_{ij}^2 (x_{jk}-c_{ij})^2))}{(\sum_m \exp(-\sum_r 1/\sigma_{ri}^2 (x_{jk}-c_{ij})^2))}$$

Finomítás: centrumok, szórások tanuló algoritmussal





Fuzzy mintapéldák

FuzzyTech 5.54 alkalmazása



Hitel kockázat becslés

Feladat:

Az FT Investment Bank ügyfelei adatbázisából megfelelő ügyfeleket kíván választani egy közvetlen levelezési kampányhoz. A kiválasztás marketing szakértők ismeretei alapján történik (hitelképes ügyfeleknek írnak).



Hitel kockázat becslés

Megoldás:

- Egy Access adatbázis menüből érhetőek el a szolgáltatások (fő űrlap)
 - Ügyfél adatok kezelése egy adatbázisban
 - Földrajzi-demográfiai jellemzők megadása, kezelése
 - Ügyfelek elemzése
- Ügyfél elemzés fuzzyTech segítségével:
 - Pénzügyi feltételek elemzése
 - Személyi feltételek elemzése
 - Levelezési kockázat elemzése



Hitel kockázat becslés

Ügyfél elemzés fuzzyTech segítségével:

- 3 szabálybázis
 - Pénzügyi felt. elemző (input: jövedelem költség, valós vagyon+teher/hitel)
 - Személyi felt. elemző (input: kor, gyerekek száma, házas)
 - Levelezési kockázat (input: pénzügyi elemzés értéke, személyi elemzés értéke, földrajzi-demográfiai jellemző)



Edit Customers

First Name: Last Name: Zip:

Address: City: State:

Phone: Day of Birth: Country:


Annual Income: Annual Spending: Mortgage:

Number of Children under 21: Real Estate Value: Married:

--> Fits Mailing (%): --> Will be mailed:

Rekord: összesen 30

Main Menu



FT Investment Bank

Database Actions:

-
-
-

Other Actions:

-
-

Edit Geodemographic Classes

GeodemographicClass:

Geodemographic Value:

Classification:

Descriptors:

ExtendedDescription:

Rekord: összesen 10

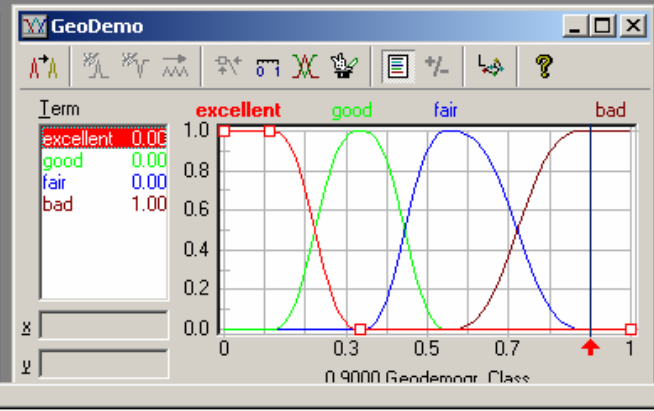
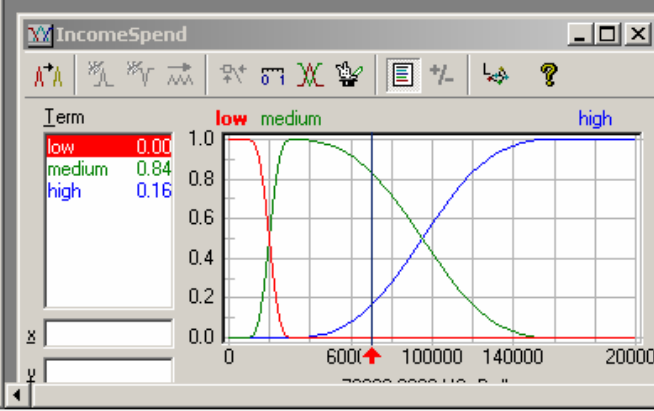
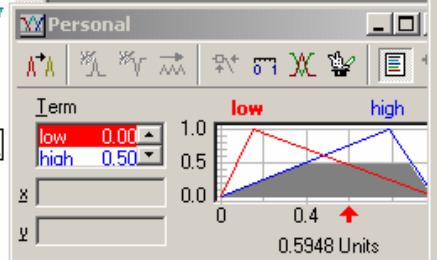
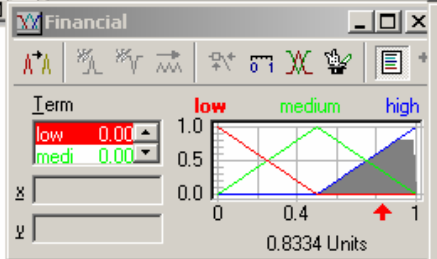
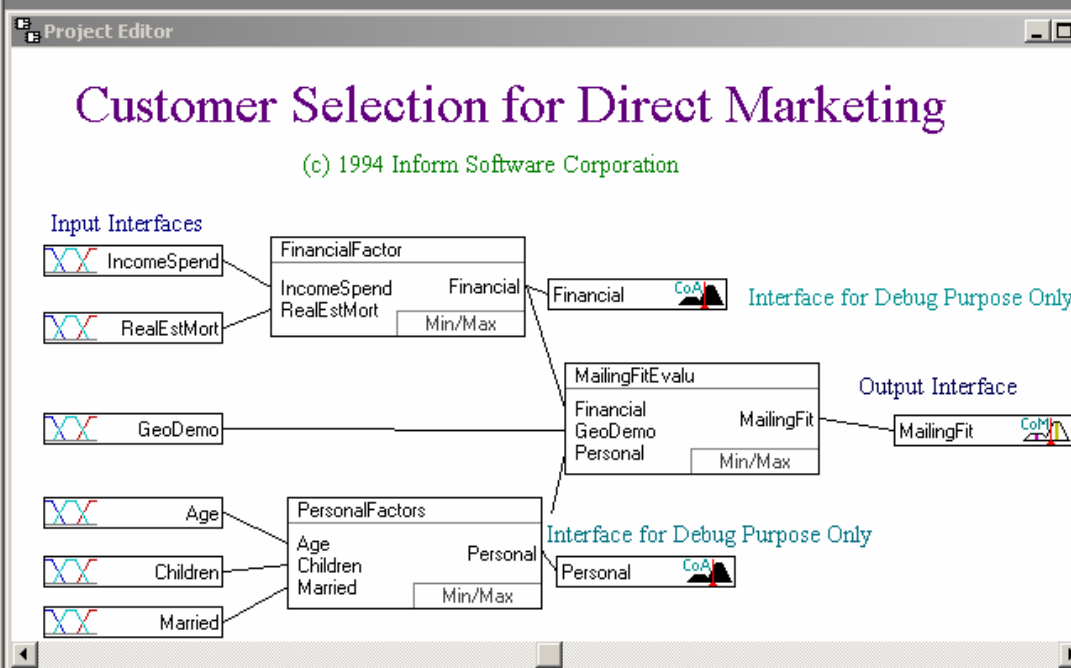
Extended Database Functions

-
-
-
-
-



FTINVEST

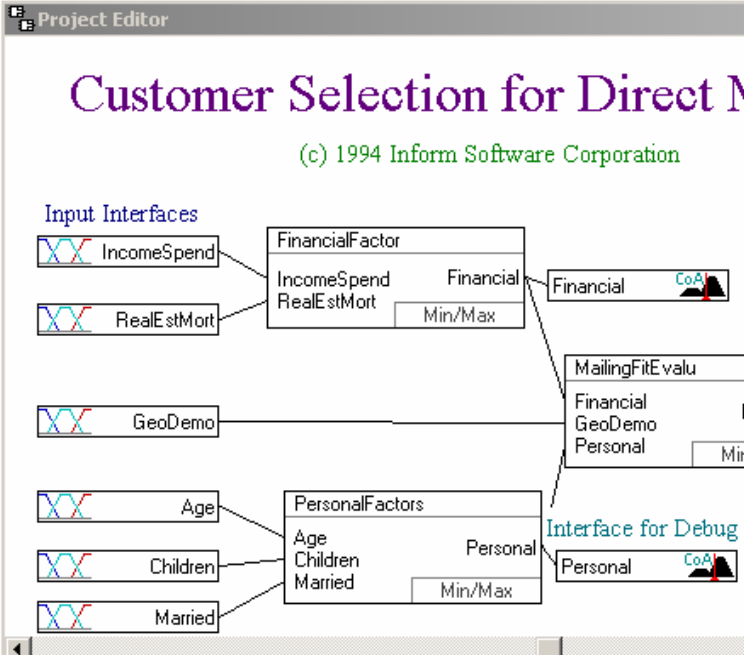
- Variable
 - Inp
 - Out
 - Inte
- Rule Blo
- Text
- Debug: I
- Online Conr





FTINVEST

- Variable
- Inp
- Out
- Inte
- Rule Blo
- Text
- Debug: I
- Online Conr



Spreadsheet Rule Editor - PersonalFactors

#	IF			THEN	
	Age	Children	Married	DoS	Personal
5		many	married	1.00	high
6	very_old		married	0.50	low
7		none		0.50	low
8	young		unmarried	1.00	low
9		none	unmarried	0.50	high

Spreadsheet Rule Editor - FinancialFactor

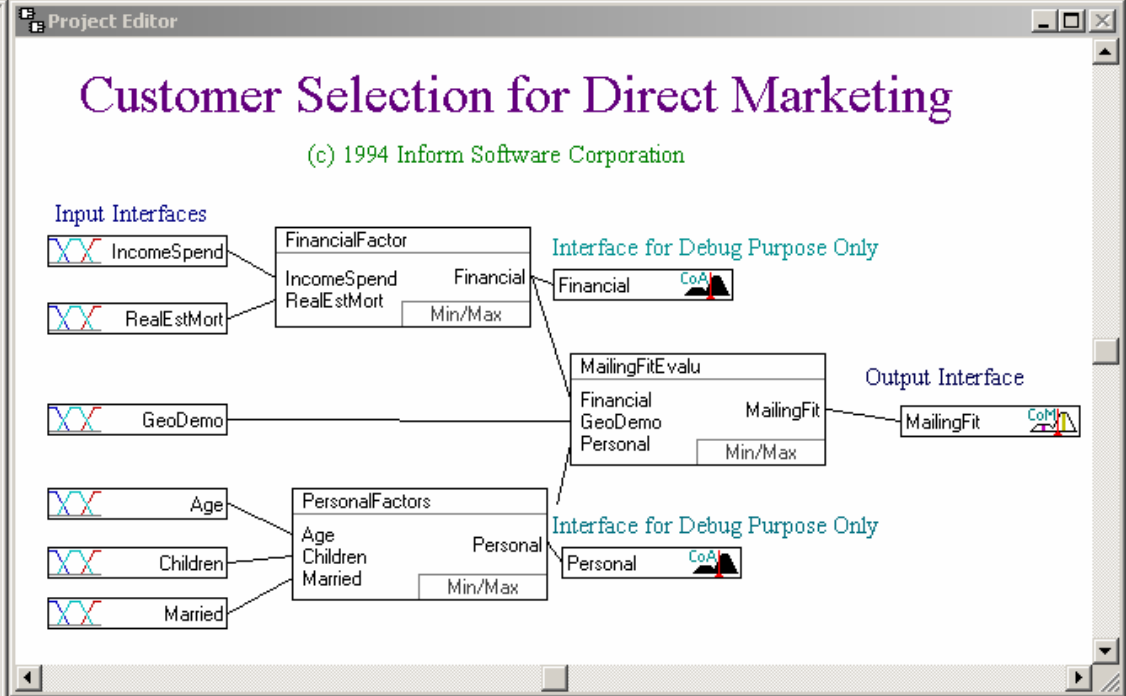
#	IF		THEN	
	IncomeSpend	RealEstMort	DoS	Financial
1	low	low	1.00	low
2	medium	low	1.00	low
3	high	low	1.00	medium
4	low	medium	1.00	medium
5	medium	medium	1.00	medium
6	high	medium	1.00	medium
7	low	high	1.00	medium
8	medium	high	1.00	high

Spreadsheet Rule Editor - MailingFitEvalu

#	IF			THEN	
	Financial	GeoDemo	Personal	DoS	MailingFit
1	low	bad	low	1.00	very_low
2	low	fair	low	1.00	very_low
3	low	good	low	1.00	low
4	low	excellent	low	1.00	low
5	medium	bad	low	1.00	very_low
6	medium	fair	low	1.00	very_low
7	medium	good	low	1.00	low
8	medium	excellent	low	1.00	medium

FTINVEST

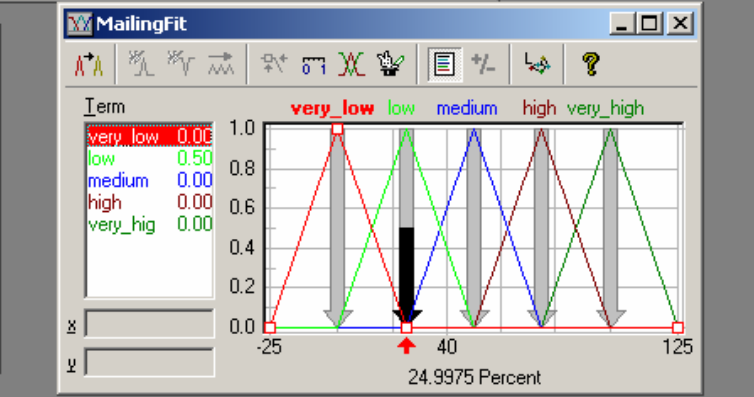
- Variable Groups
 - Inputs
 - Outputs
 - Intermediates
- Rule Blocks
- Text
- Debug: RCU
- Online Connections



Watch: RCU Debug Mode

50.5000

Inputs:		Outputs:	
Age	50.0000	Financial	0.8334
Children	2.0000	MailingFit	24.9975
GeoDemo	0.9000	Personal	0.5948
IncomeSpend	70000.0000		
Married	1.0000		
RealEstMort	500000.0000		

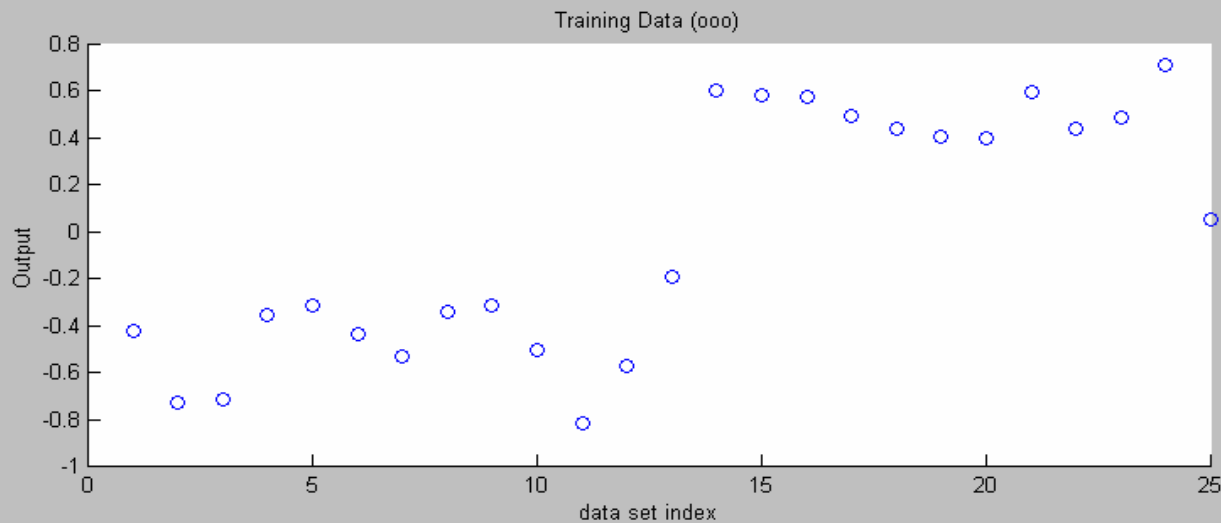




Fuzzy rendszerek generálása: ANFIS használat

Egyváltozós függvény illesztés
pontokra

Demo feladatok



ANFIS Info.

of inputs: 1
of outputs: 1
of input mfs: 0
of train data pairs: 25

Structure

Clear Plot

Load data

Type: Training
 Testing
 Checking
 Demo

From: disk
 worksp.

Load Data...

Clear Data

Generate FIS

Load from disk
 Load from worksp.
 Grid partition
 Sub. clustering

Generate FIS ...

Train FIS

Optim. Method: hybrid

Error Tolerance: 0

Epochs: 3

Train Now

Test FIS

Plot against: Training data
 Testing data
 Checking data

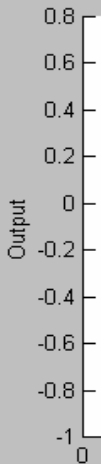
Test Now

train data loaded

Help

Close

Training Data (ooo)



INPUT

Number of MFs:

MF Type:

- trimf
- trapmf
- gbellmf**
- gaussmf
- gauss2mf
- pimf
- dsigmf
- psigmf

To assign a different number of MFs to each input, use spaces to separate these numbers.

OUTPUT

MF Type:

- constant
- linear**

ANFIS Info.

of inputs: 1
 # of outputs: 1
 # of input mfs: 0
 # of train data pairs: 25

Load data

Type: Training Testing Checking Demo

From: disk worksp.

Load from worksp.
 Grid partition
 Sub. clustering

Train FIS

Optim. Method:

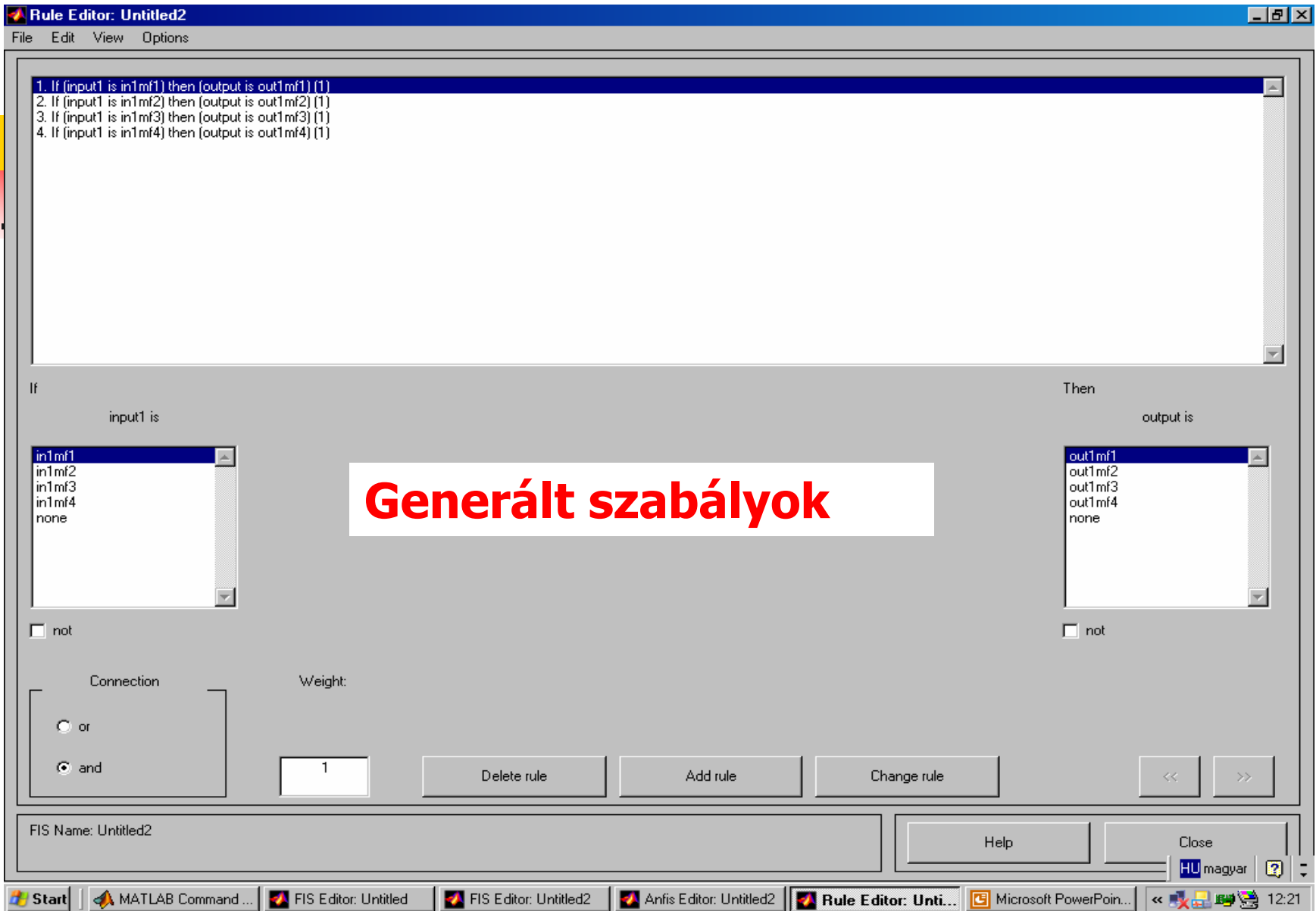
Error Tolerance:

Epochs:

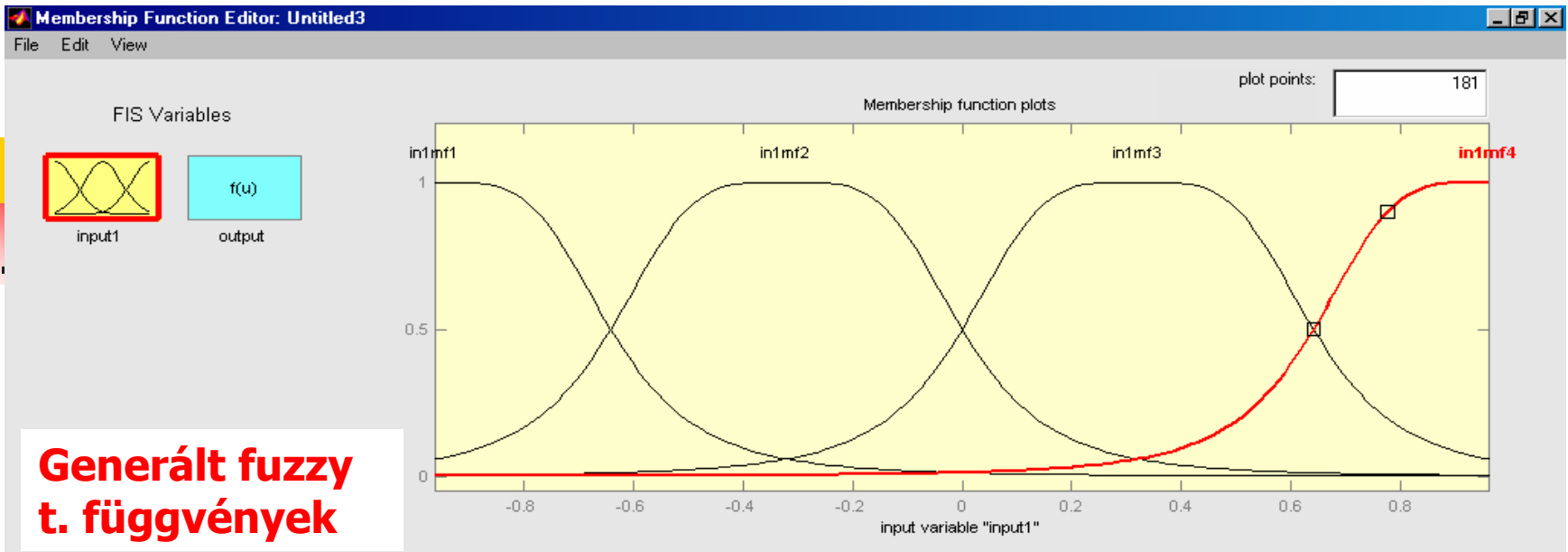
Test FIS

Plot against: Training data Testing data Checking data

train data loaded



Generált szabályok



Current Variable

Name: input1

Type: input

Range: [-0.96 0.96]

Display Range: [-0.96 0.96]

Ready

Current Membership Function (click on MF to select)

Name: in1mf4

Type: gbellmf

Params: [0.32 2 0.96]

Help Close

Windows taskbar: Start, MATLAB Comm..., FIS Editor: Untit..., Anfis Editor: Unti..., Membership ..., Microsoft Power..., 12:23

Anfis Model Structure

File Edit Window Help

input inputmf rule outputmf weighted sum output output

A hozzárendelt NH

— not
— or
— and

Close

Click on each node to see detailed information

Windows taskbar: Start, Microsoft PowerPoin..., MATLAB Command..., FIS Editor: Untitled, FIS Editor: Untitled2, Anfis Editor: Untitled2, Anfis Model Stru..., 12:27

Anfis Editor: Untitled2

File Edit View

Training data : o FIS output : *

Output

Index

ANFIS Info.

- # of input: 1
- # of outputs: 1
- # of input mfs: 4

Structure

Clear Plot

Load data

Type:

- Training
- Testing
- Checking
- Demo

From:

- disk
- worksp.

Generate FIS

- Load from disk
- Load from worksp.
- Grid partition
- Sub. clustering

Train FIS

Optim. Method:

hybrid

Error Tolerance:

0

Epochs:

3

Test FIS

Plot against:

- Training data
- Testing data
- Checking data

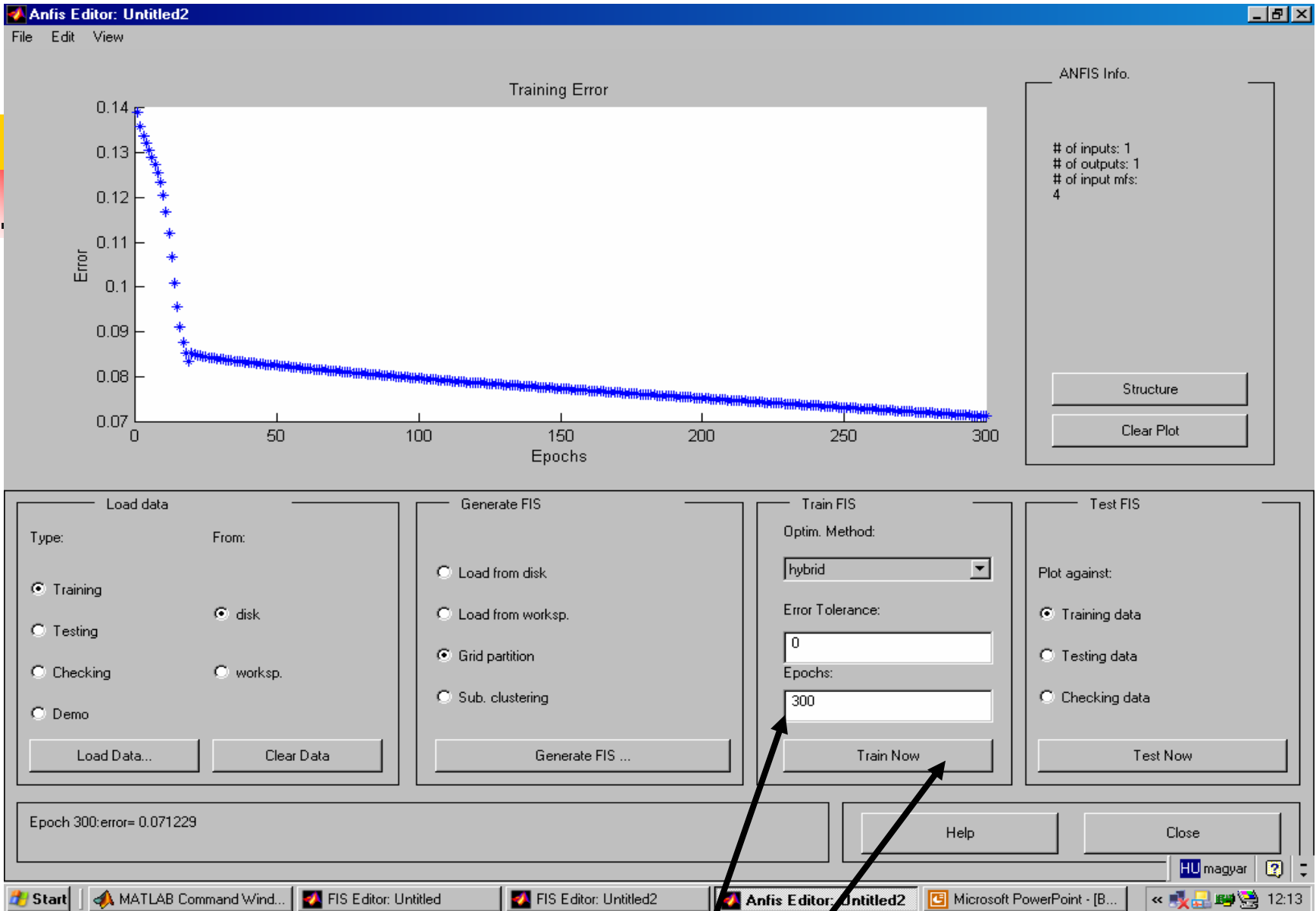
Test Now

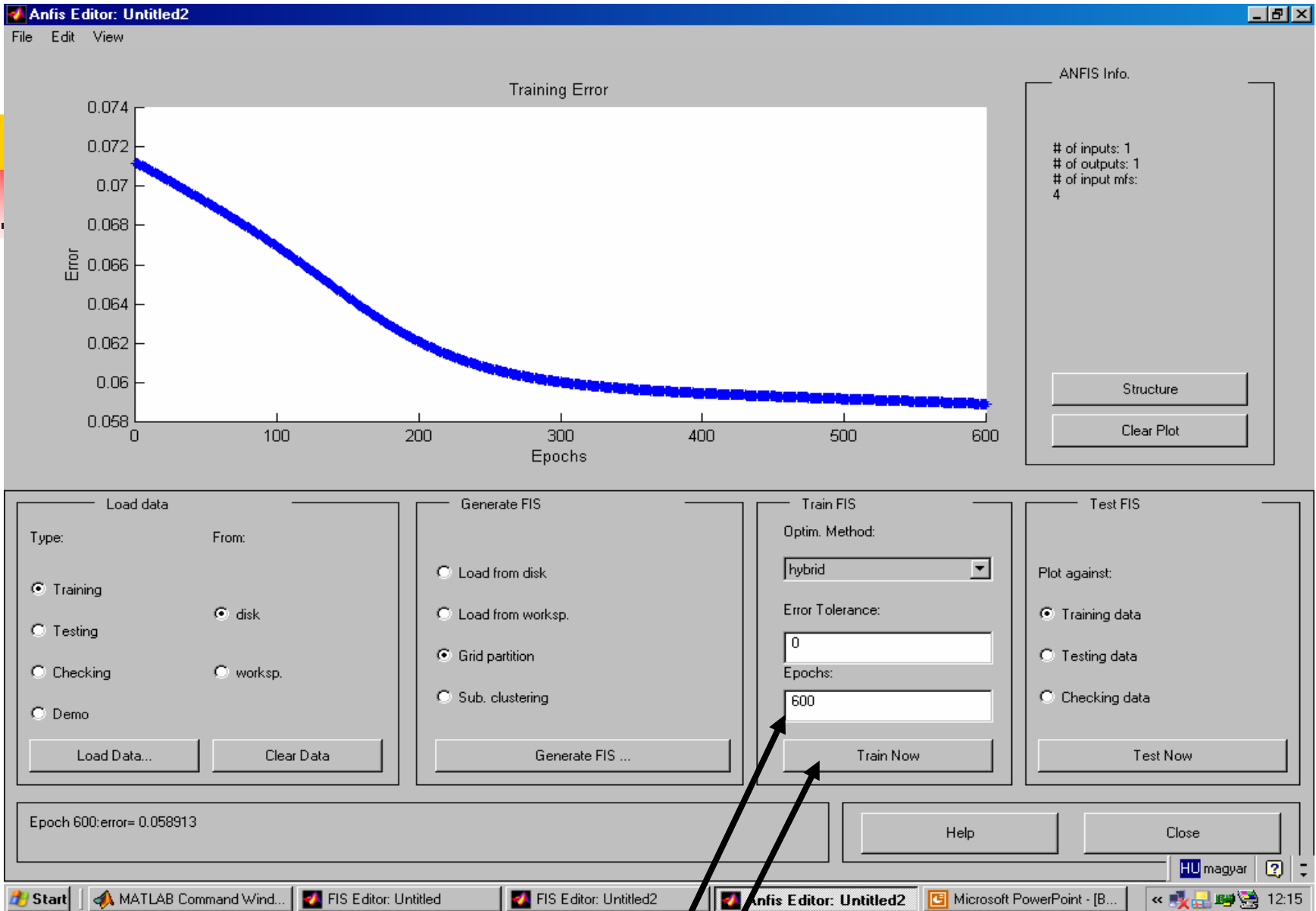
Close

A generált fuzzy rendszer eredménye a tanulás előtt.

Start Microsoft PowerPoint - [fu... MATLAB Command Wind... FIS Editor: Untitled FIS Editor: Untitled2 Anfis Editor: Untitled2

HU magyar 12:52





Anfis Editor: Untitled2

File Edit View

Training data : o FIS output : *

ANFIS Info:

- # of inputs: 1
- # of outputs: 1
- # of input mfs: 4

Structure

Clear Plot

Load data

Type: Training (selected), Testing, Checking, Demo

From: disk (selected), worksp.

Load Data... Clear Data

Generate FIS

Load from disk, Load from worksp., Grid partition (selected), Sub. clustering

Generate FIS ...

Train FIS

Optim. Method: hybrid

Error Tolerance: 0

Epochs: 600

Train Now

Test FIS

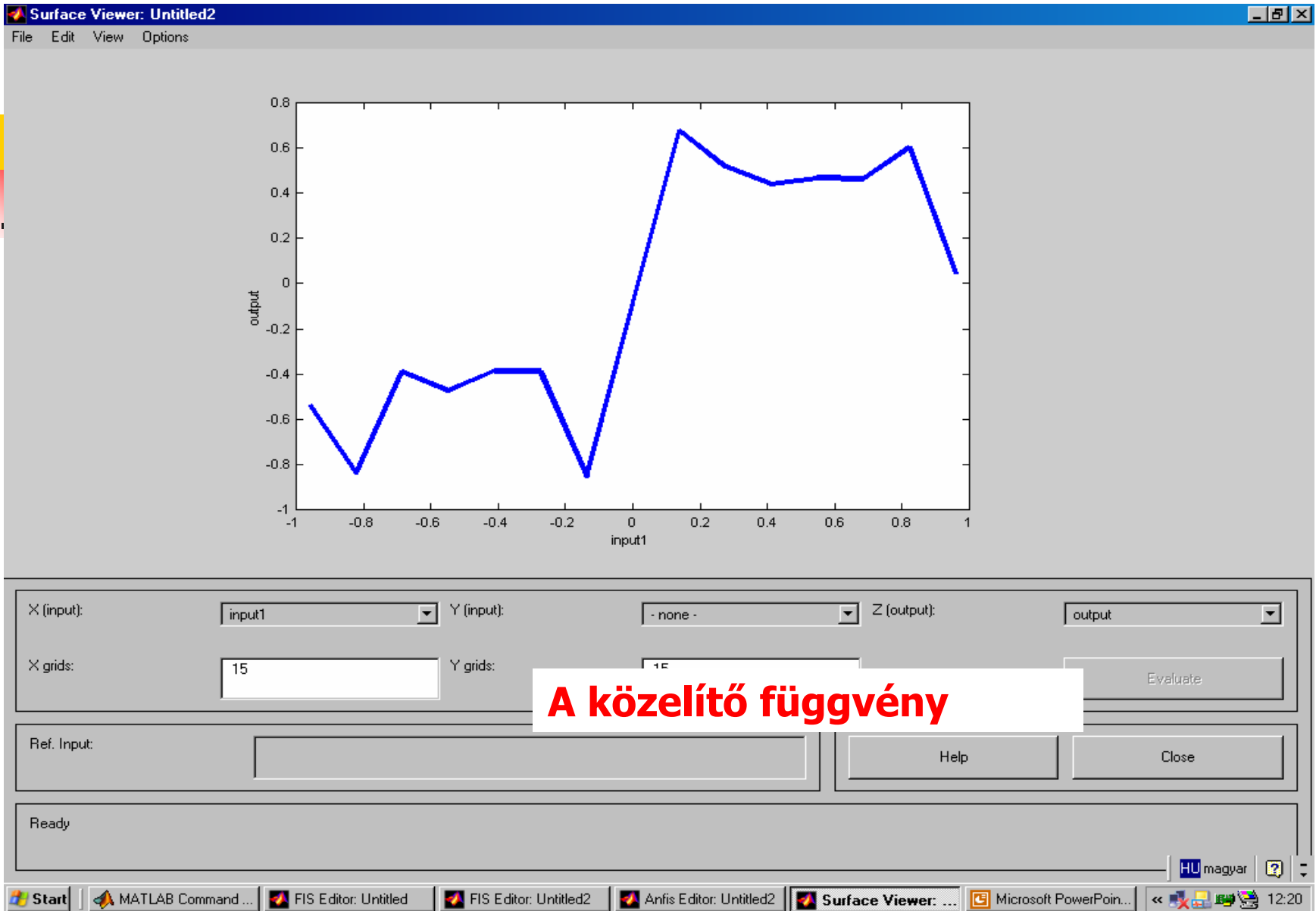
Plot against: Training data (selected), Testing data, Checking data

Test Now

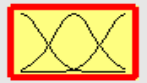
Average testing error: 0.058535

Help Close

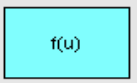
Windows taskbar: Start, MATLAB Command Wind..., FIS Editor: Untitled, FIS Editor: Untitled2, Anfis Editor: Untitled2, Microsoft PowerPoint - [B...], 12:17



FIS Variables

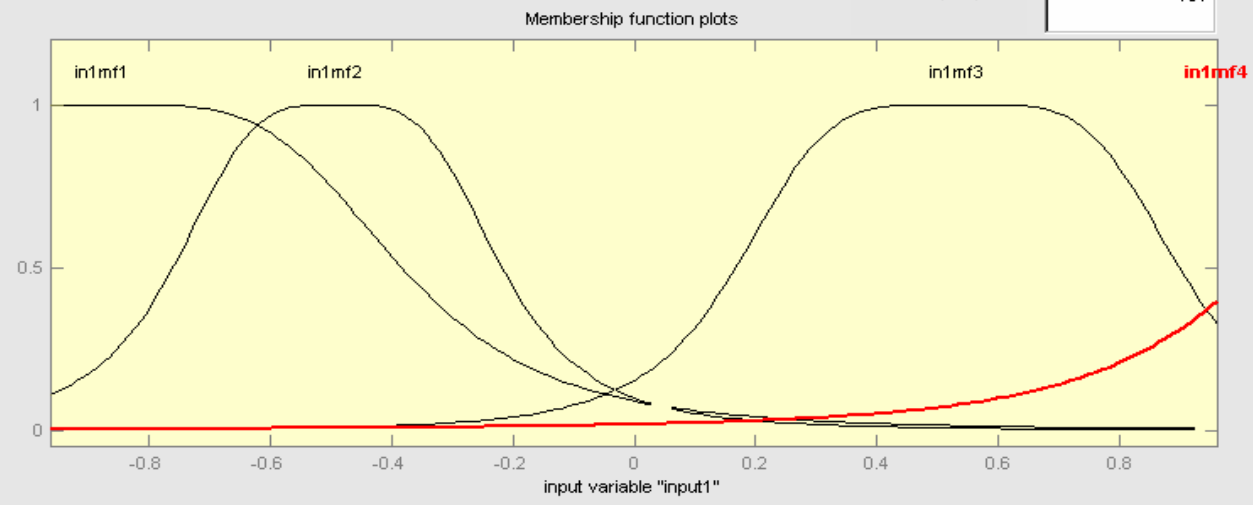


input1



output

Tanulás után



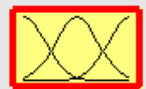
Current Variable

Name: input1

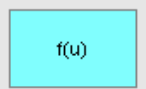
Current Membership Function (click on MF to select)

Name: in1mf4

FIS Variables

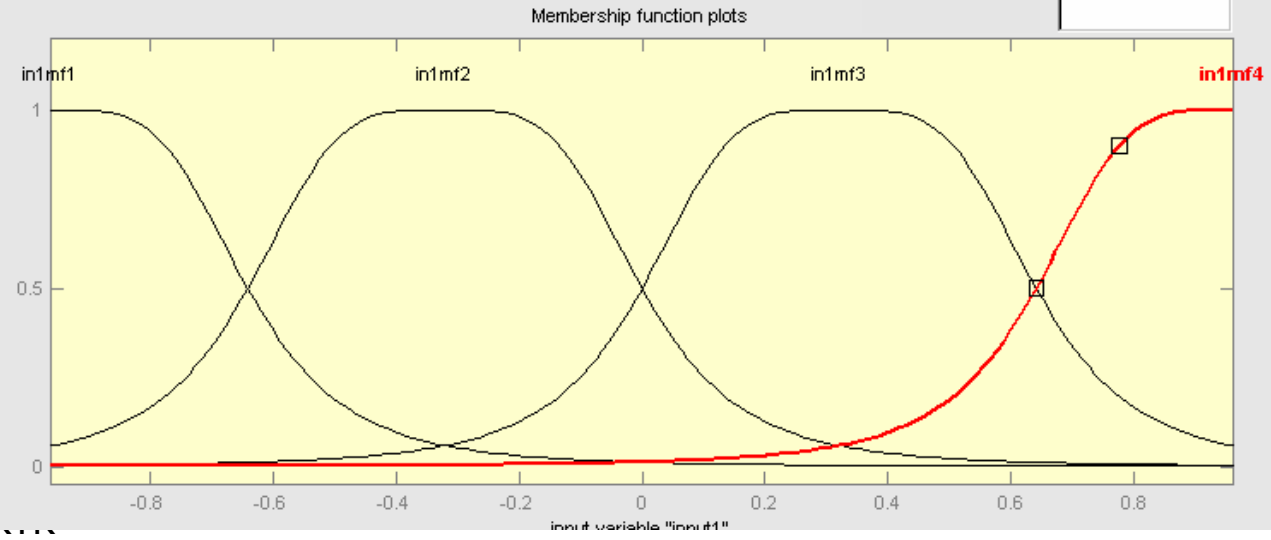


input1



output

Tanulás előtt



Újabb függvényillesztés pontokra

The screenshot displays the Anfis Editor software interface. The main window, titled "Anfis Editor: Untitled3", features a menu bar with "File", "Edit", and "View". The central plot area, titled "Training Data (ooo)", shows a scatter plot of "Output" (y-axis, ranging from -0.8 to 0.8) versus "data set index" (x-axis, ranging from 0 to 30). The data points are scattered, with a general trend of decreasing output as the index increases. To the right of the plot is the "ANFIS Info." panel, which displays the following statistics:

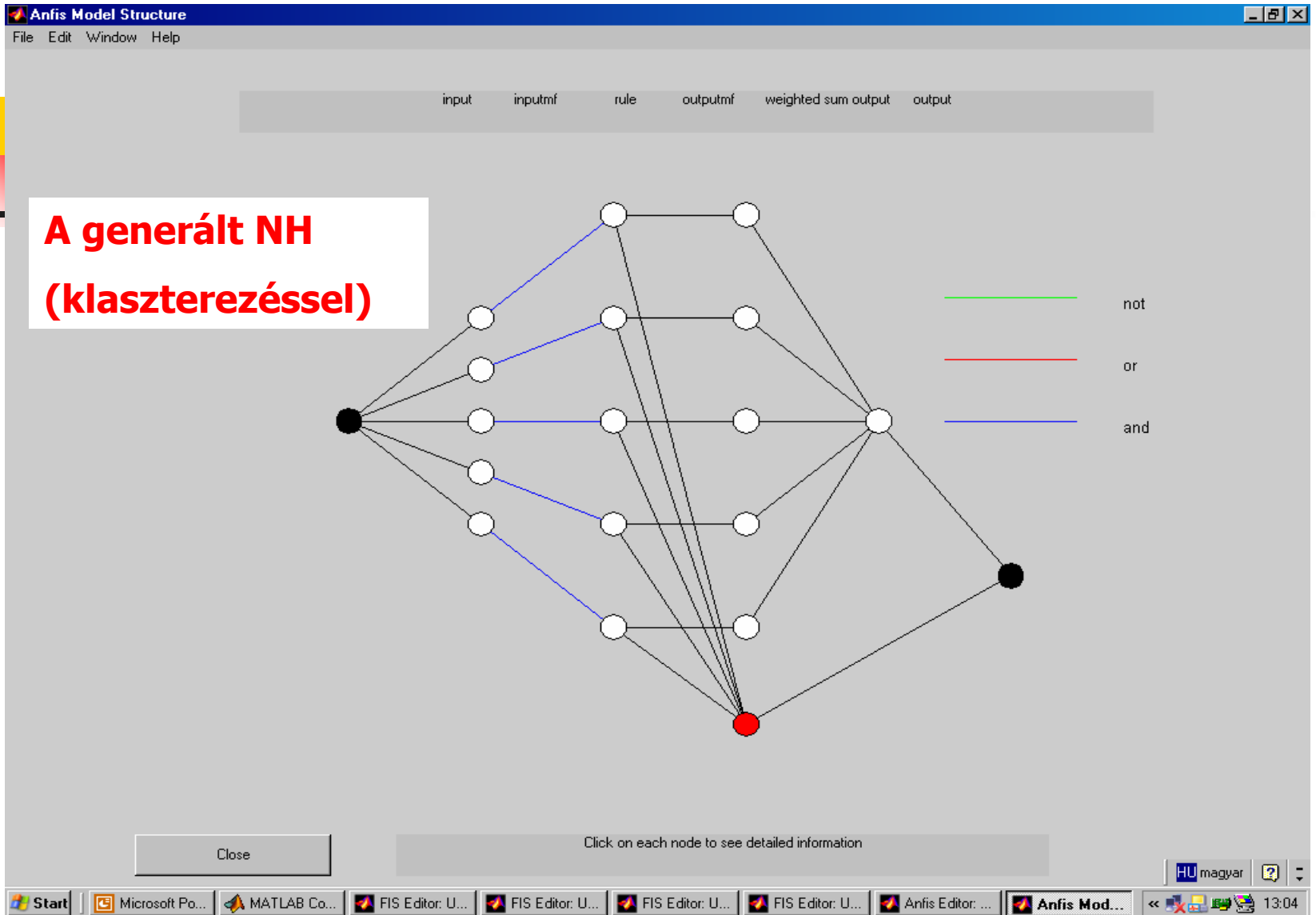
- # of inputs: 1
- # of outputs: 1
- # of input mfs: 4
- # of train data pairs: 26

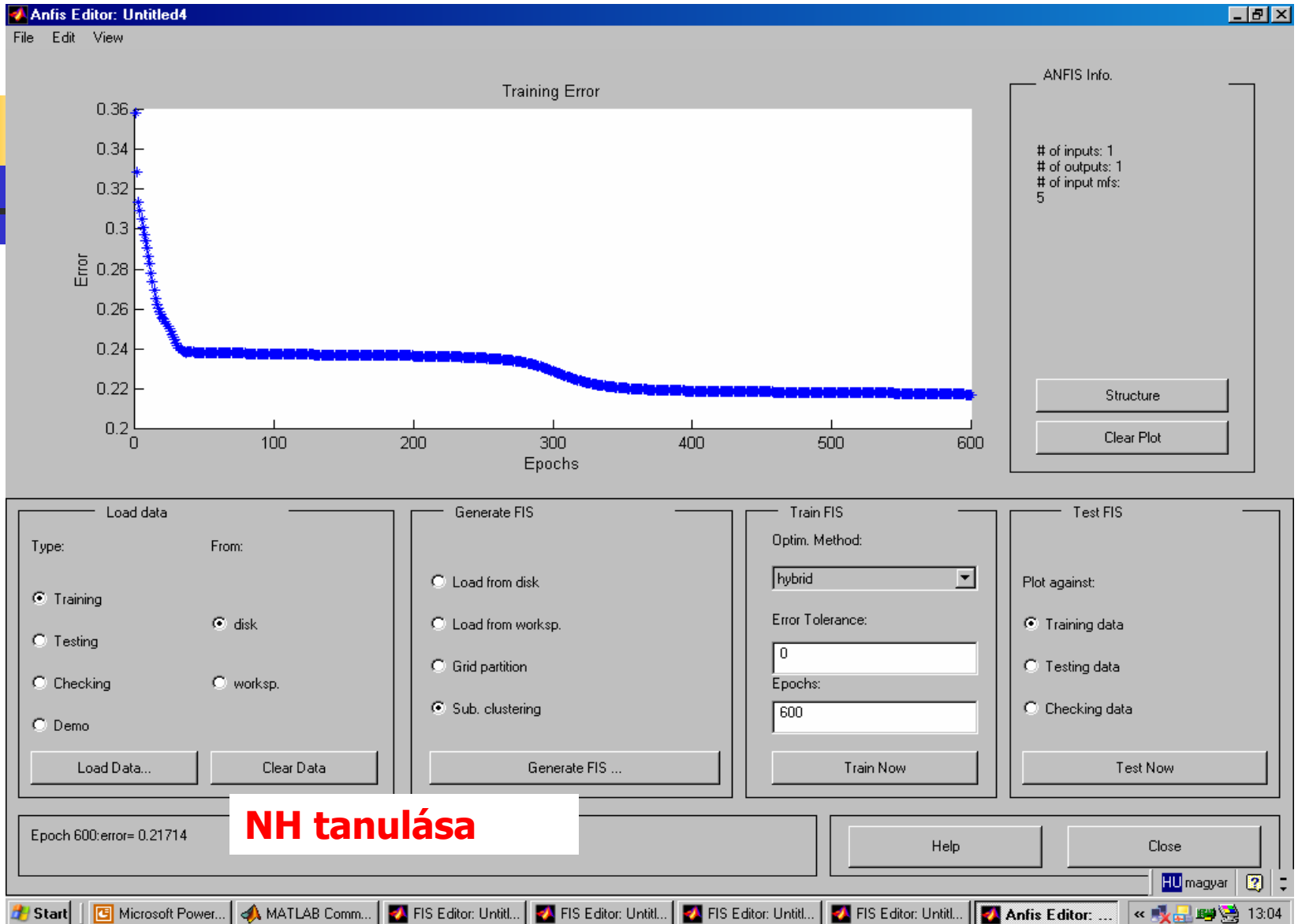
Below the statistics are two buttons: "Structure" and "Clear Plot".

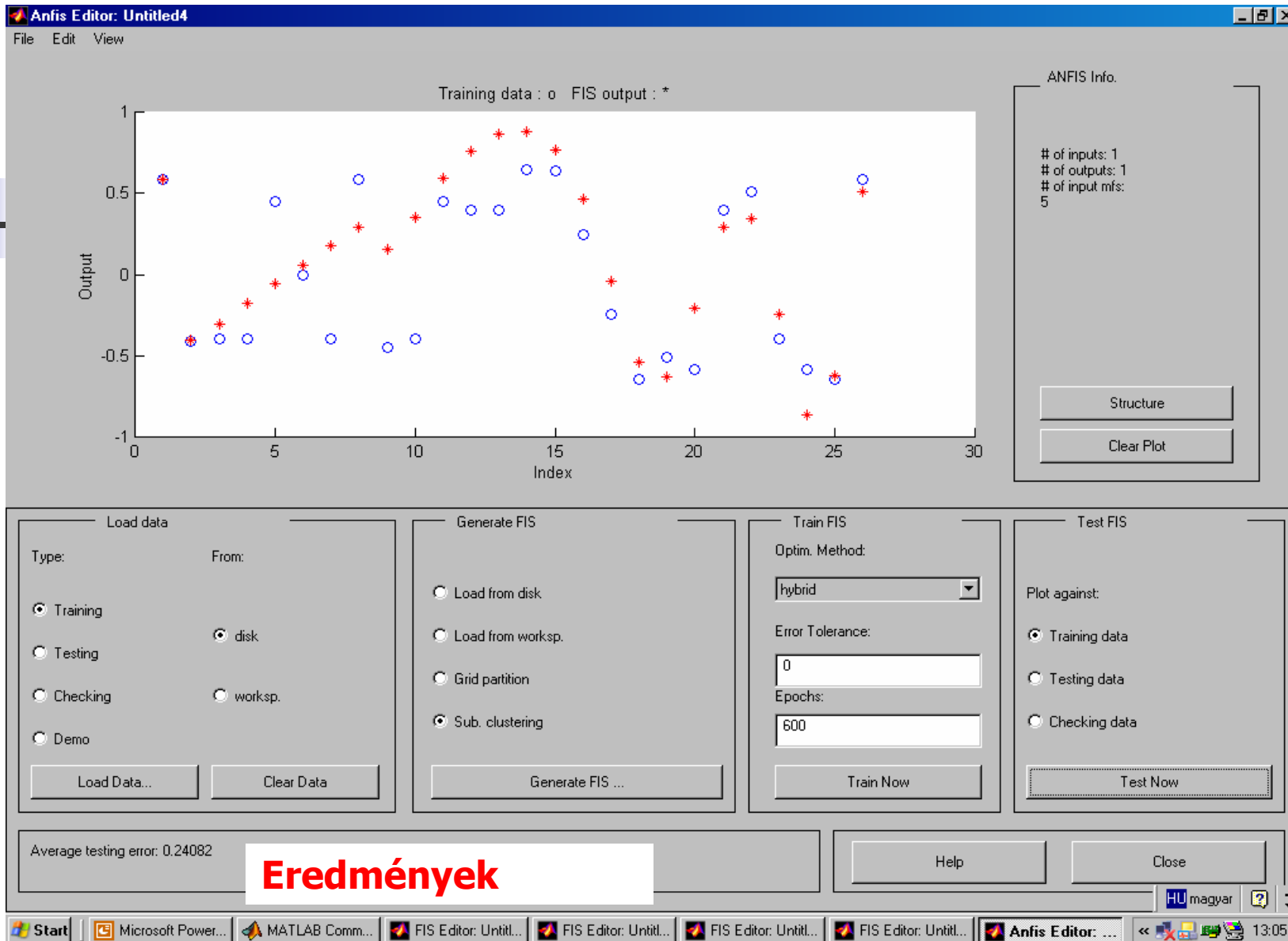
The bottom section of the interface contains four main control panels:

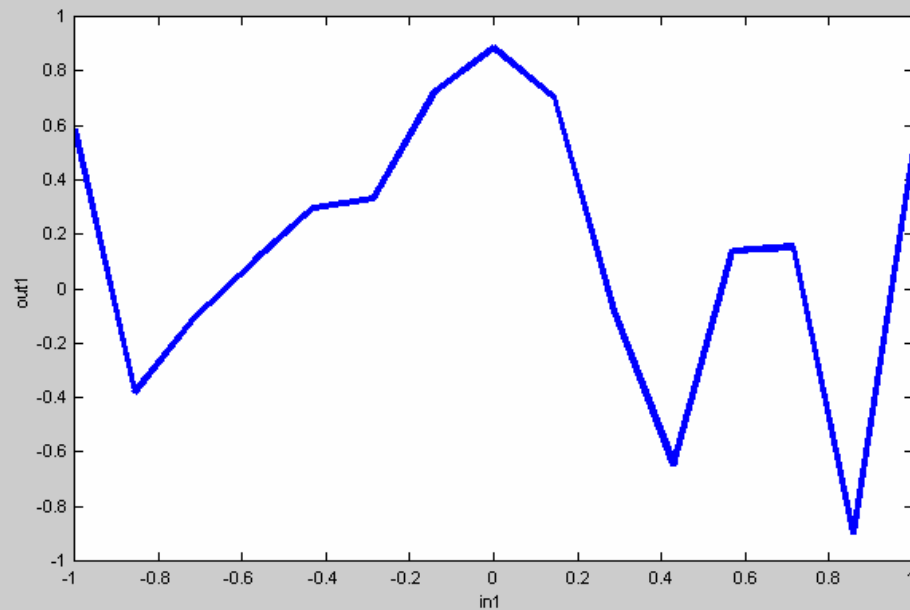
- Load data:** Includes radio buttons for "Type" (Training, Testing, Checking, Demo) and "From" (disk, worksp.). Buttons for "Load Data..." and "Clear Data" are at the bottom.
- Generate FIS:** Includes radio buttons for "Load from disk", "Load from worksp.", "Grid partition", and "Sub. clustering". A "Generate FIS ..." button is at the bottom.
- Train FIS:** Includes a dropdown for "Optim. Method" (set to "hybrid"), an "Error Tolerance" input field (set to "0"), and an "Epochs" input field (set to "600"). A "Train Now" button is at the bottom.
- Test FIS:** Includes radio buttons for "Plot against" (Training data, Testing data, Checking data). A "Test Now" button is at the bottom.

A status bar at the bottom left shows "train data loaded". The Windows taskbar at the very bottom displays the Start button and several open applications: Microsoft PowerPoint, MATLAB Command, FIS Editor: Untitled, FIS Editor: Untitled2, FIS Editor: Untitled3, and Anfis Editor: Untitled. The system clock shows 13:11.









X (input):	<input type="text" value="in1"/>	Y (input):	<input type="text" value="- none -"/>	Z (output):	<input type="text" value="out1"/>
X grids:	<input type="text" value="15"/>	Y grids:	<input type="text" value="15"/>	<input type="button" value="Evaluate"/>	

Ref. Input:	<input type="text"/>
-------------	----------------------

<input type="button" value="Help"/>	<input type="button" value="Close"/>
-------------------------------------	--------------------------------------

Ready

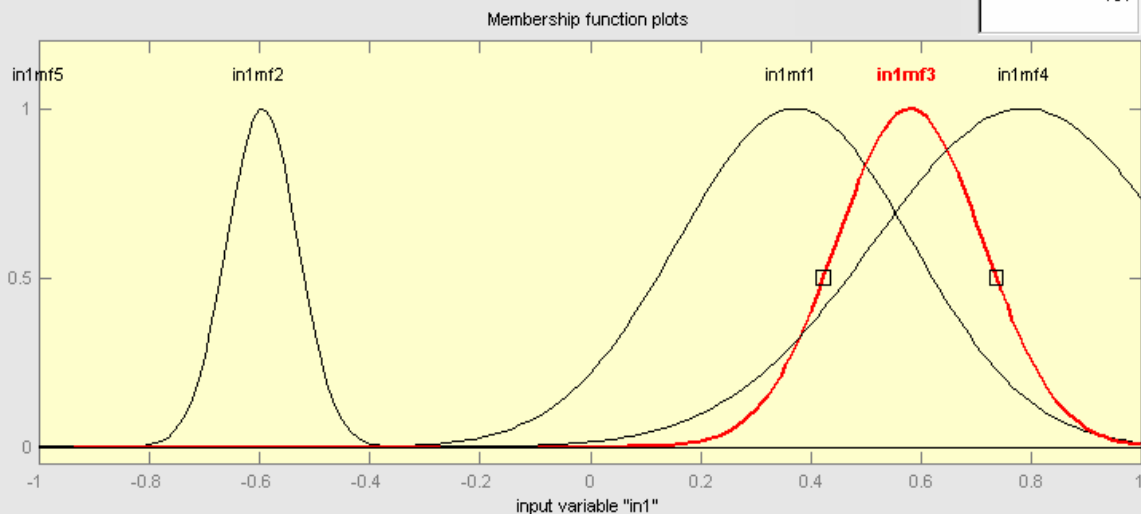
A közelítő függvény

HU magyar

FIS Variables



Tanulás után



Current Variable

Name in1

Type input

FIS Variables



Tanulás előtt

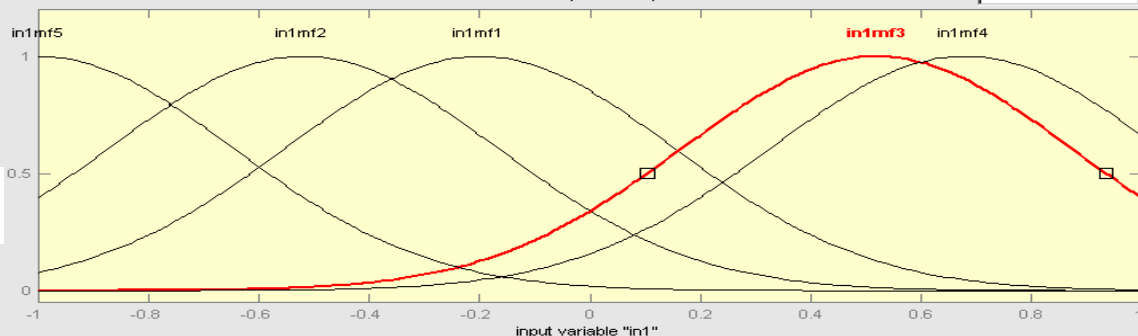
Current Membership Function (click on MF to select)

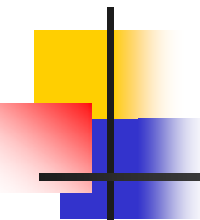
Name

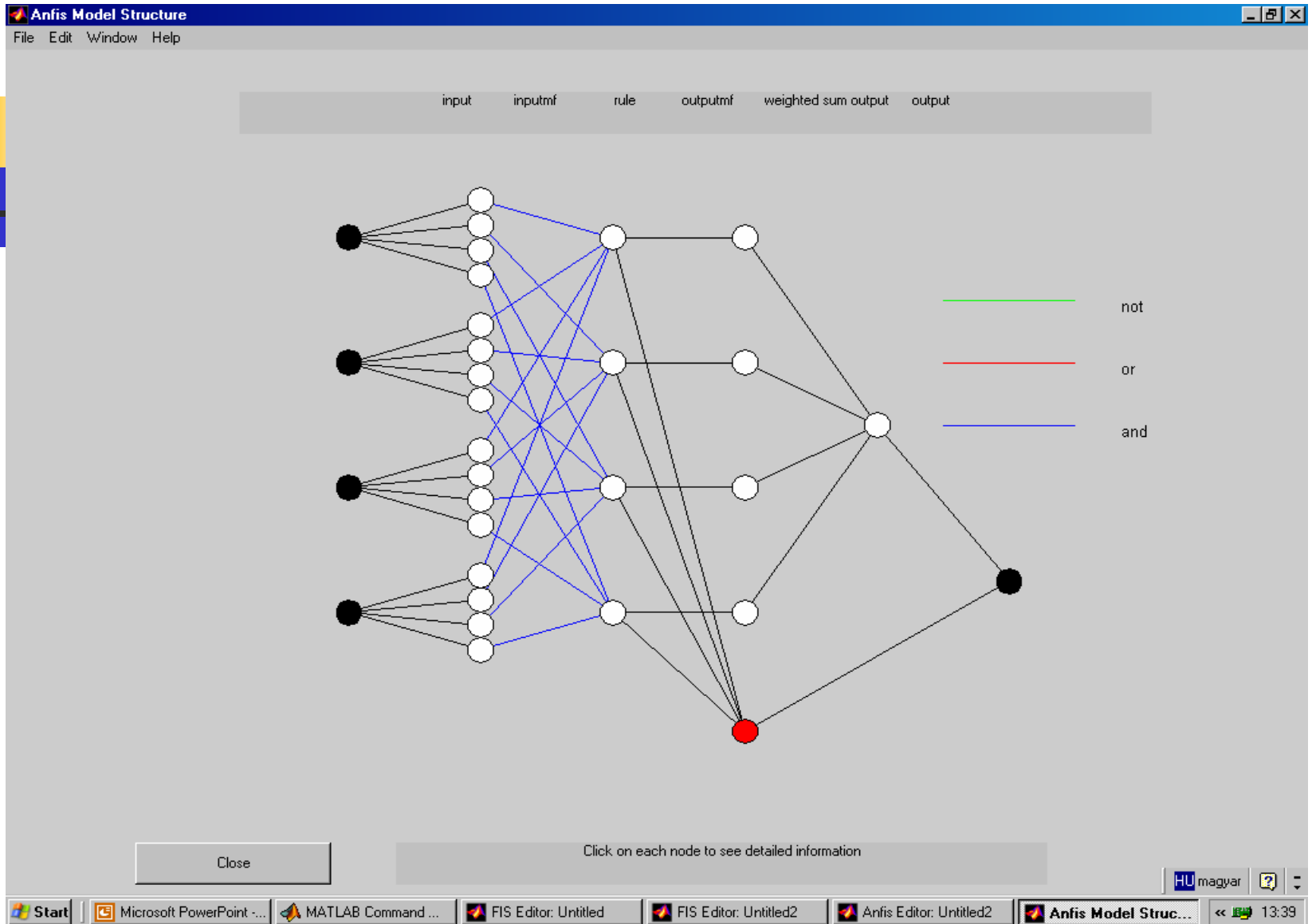
in1mf3

Type

gaussmf



- 
-
- IRIS adatok osztályozása - 3 osztály
 - Klaszterezéssel - 4 szabály



Membership Function Editor: Untitled2

File Edit View

plot points: 181

FIS Variables

in1 out1

in2

in3

in4

Membership function plots

in1mf1 in1mf2 in1mf3 in1mf4

1

0.5

0

45 50 55 60 65 70 75

input variable "in1"

Current Variable

Name in1

Type input

Range [43 79]

Display Range [43 79]

Current Membership Function (click on MF to select)

Name

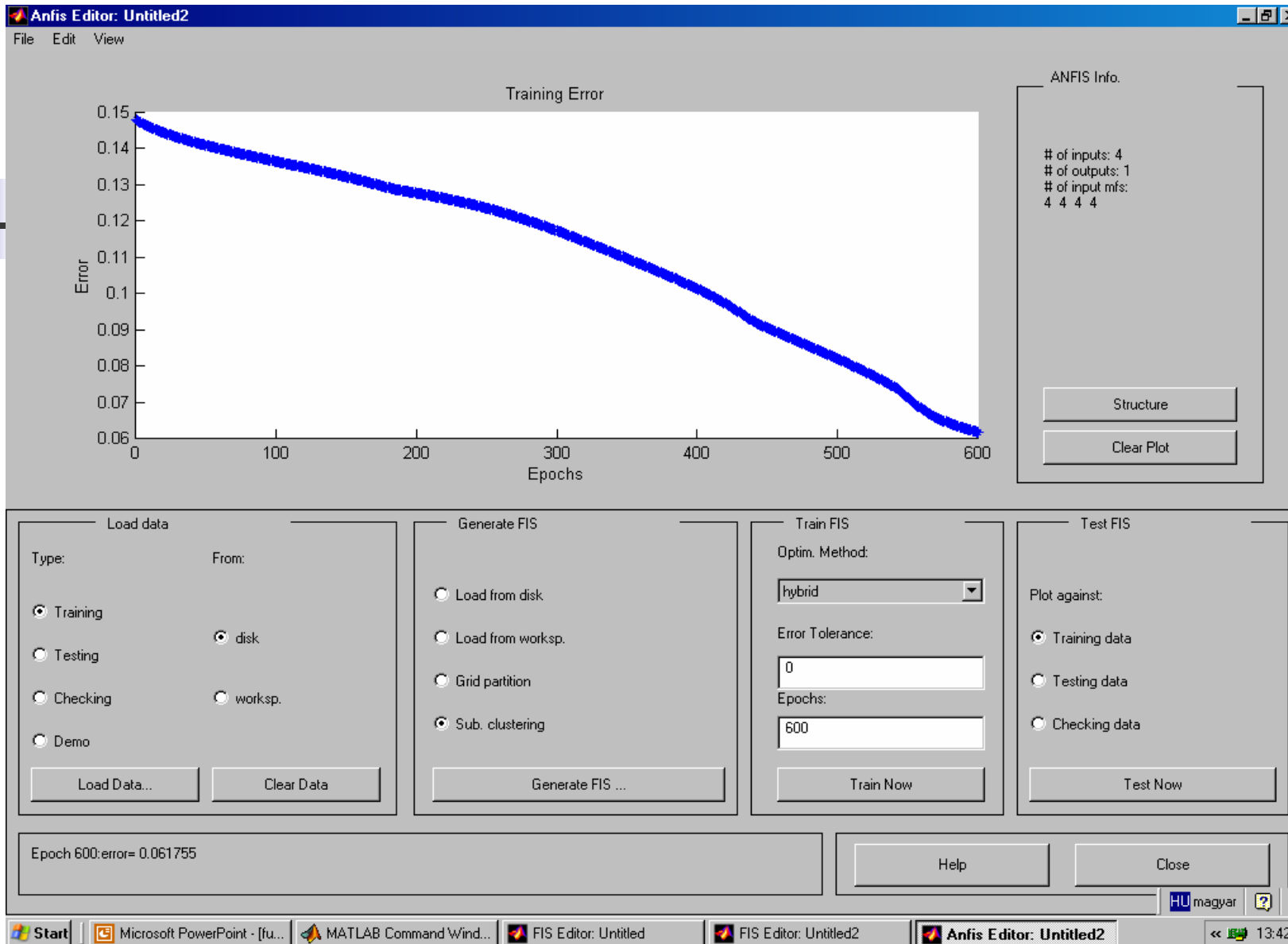
Type trimf

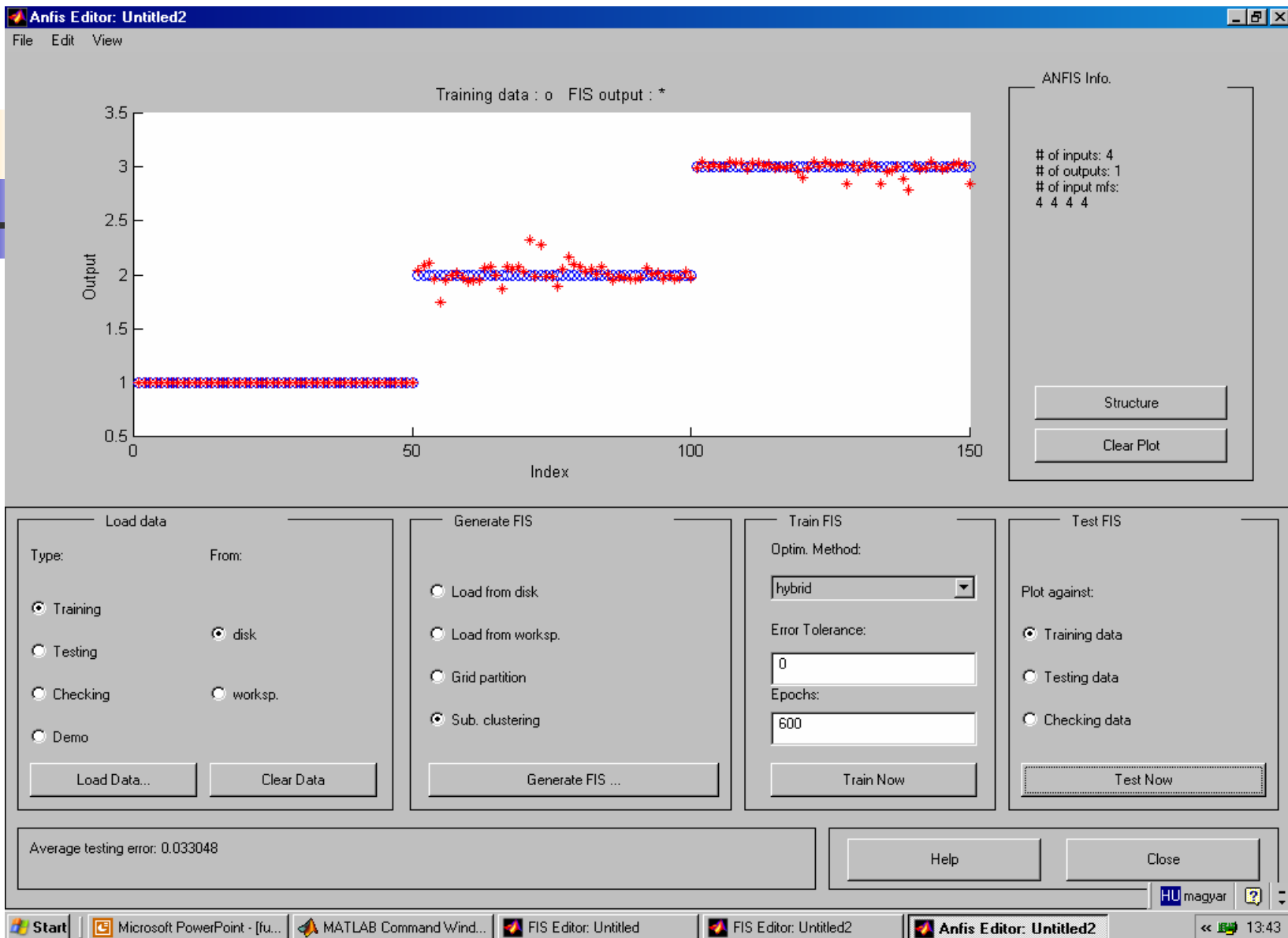
Params

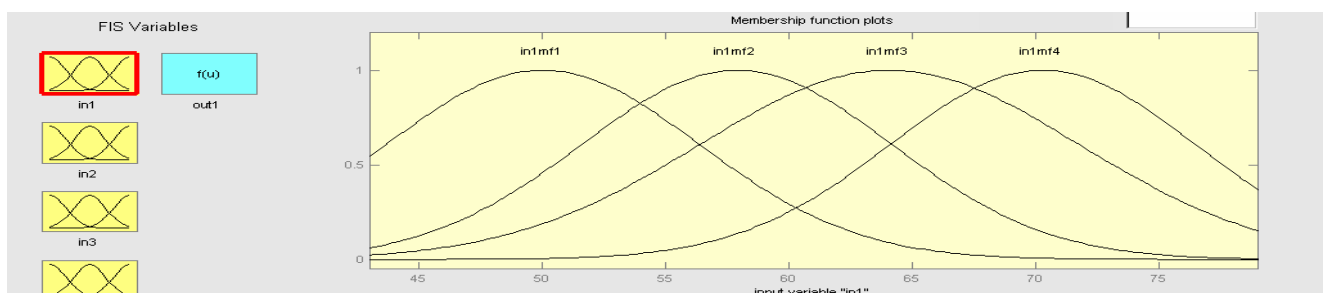
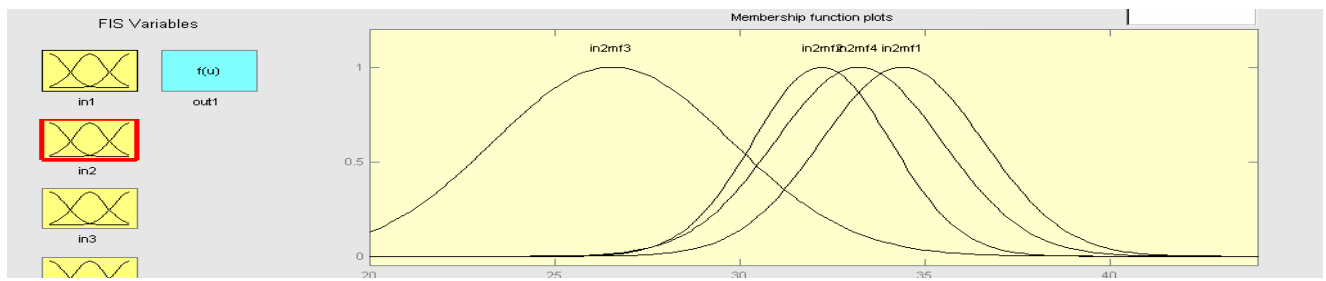
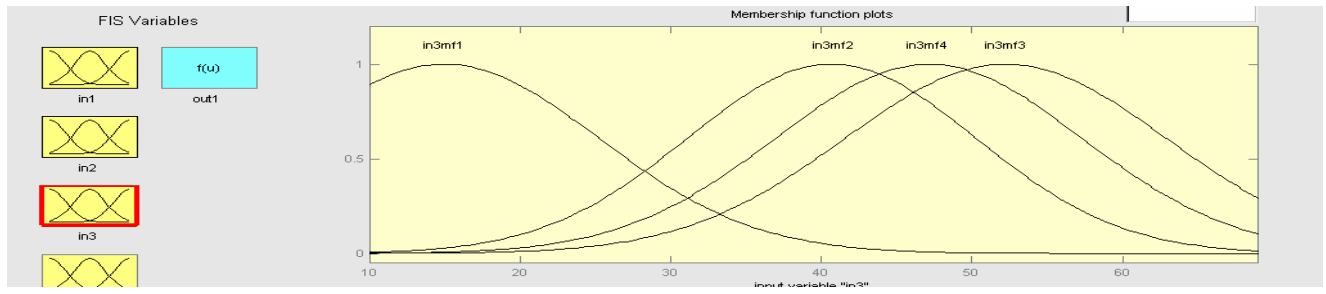
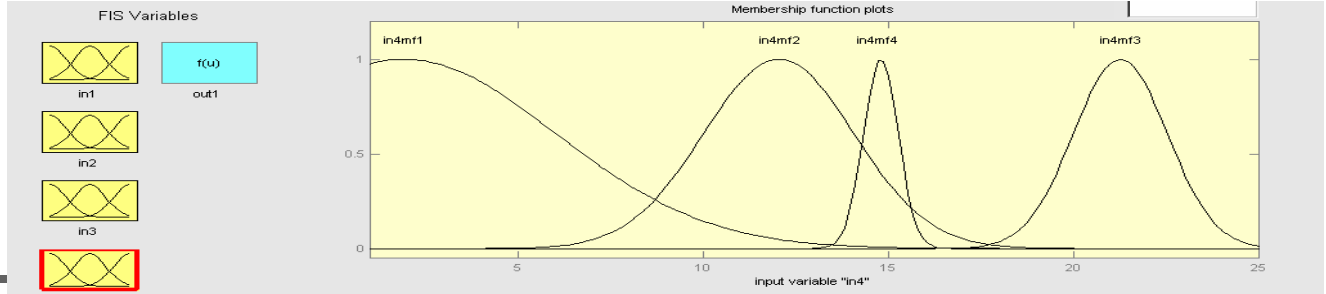
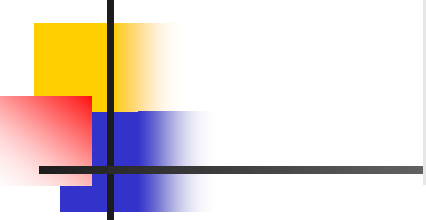
Help Close

Ready

Start Microsoft PowerPoint ... MATLAB Command ... FIS Editor: Untitled FIS Editor: Untitled2 Anfis Editor: Untitled2 Membership Funct... 13:40









Evolúciós algoritmusok



Evolúciós algoritmusok

- Bevezetés
- EA alapfogalmak, változatok
- GA, GP, ES, EP technikák
- Alkalmazások



Intelligens szoftverek

Ismeret

szimbólikus

numerikus

strukturált

szakértői r.

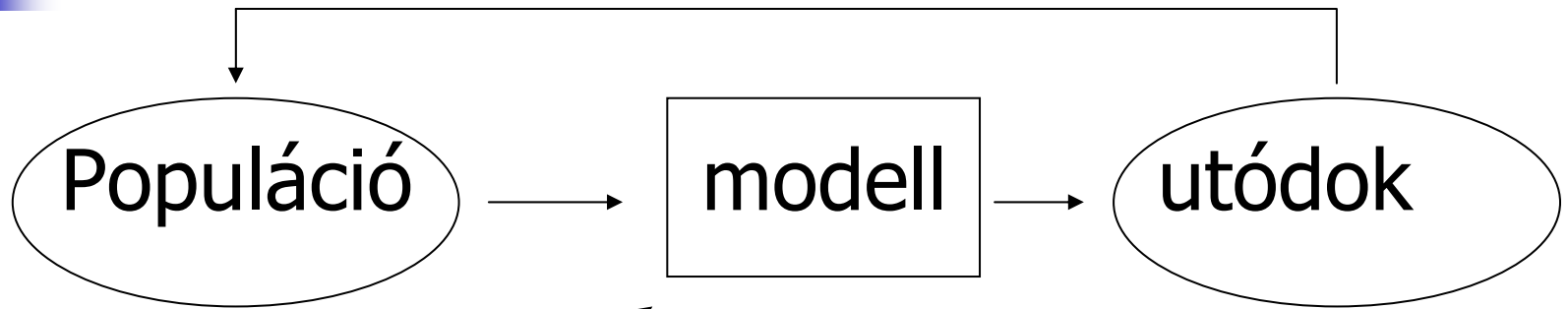
fuzzy r.

strukturálatlan

ev. alg

neur. háló

Evolúciós algoritmus konceptió



Modell koncepció alapja

- Biológiai evolúció,
- Biológiai rendszer, faj információ csere
- Matematikai modell

Evolúciós algoritmus koncepció



Modell koncepció alapja

- Biológiai evolúció,
 - Öröklés (szülő – utód), vírusok, baktériumok
- Egy biológiai rendszer
 - Immun rendszer
 - faj információ csere pl. méhek, hangyák
- Matematikai modell
 - Lineáris kombináció alapú
 - Valószínűségi modell a populáció alapján
 - Valószínűségi modell a korábbi populációk alapján



Evolúciós algoritmusok

- Az EA mint módszer
 - Sztochasztikus kereső eljárás
 - Tanuló algoritmus
 - Populáció alapú algoritmus, amely információ csereként értelmezi az evolúciót



Evolúciós algoritmusok

- EA: probléma megoldó metaheurisztika
- EA elméleti háttér (MI - Op.kut.)
 - Markov-láncok
 - statisztika döntéselmélete
 - Holland séma elmélet
 - statisztikai mechanika
- Számítási igény



Evolúciós algoritmusok

- EA alapfogalmak
 - individuum/kromoszóma (egy megoldás)
 - Populáció (megoldás halmaz)
 - szülő, utód (régi- új megoldás)
 - kereső operátorok (műveletek)
 - Rekombináció/crossover, mutáció, szelekció
 - Fitnessz (megoldás értékelésre)
 - generáció



Evolúciós algoritmusok

- EA általános ciklus
 - stratégiai par. választása
 - populáció inicializálás
 - individumok értékelése
 - generációs ciklus:
 - szülők választása, autódok generálása
 - utódok értékelése, új populáció előállítás
 - megállási feltétel
 - eredmény



Evolúciós algoritmusok

- EA változatok
 - genetikus algoritmusok (GA)
 - genotípus változtatás (kromoszóma vált.)
 - evolúciós stratégia (ES)
 - fenotípus vált. (mutáció, rekombináció, det szelekció-- legjobb túlélő)
 - evolúciós programozás (EP)
 - faj fenotip. vált. (mutáció, sztoch. szelekció)



Genetikus algoritmusok

- Alap GA

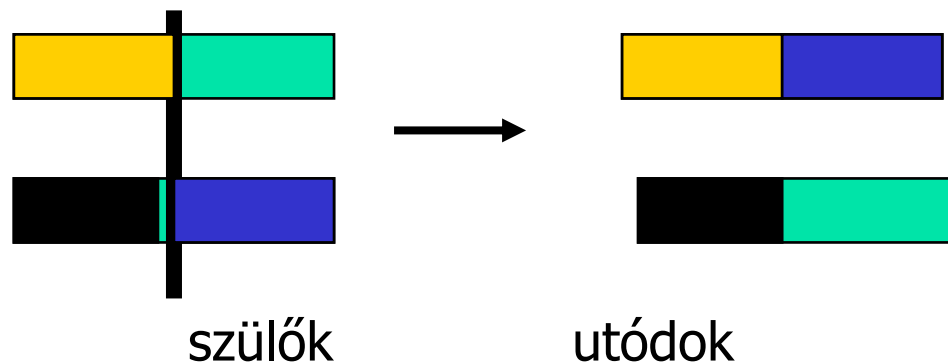
- Holland '60, genetikus mech. Optimalizálás
- individum: kromoszóma= bitsorozat (gensor.)
 - n-változó, kódolásuk

- Alap GA lépések

- inicializálás 30-500 individum, véletlen ért.
- értékelés fitnessz \sim célfüggvény
- generációs ciklus (teljesen új populáció)

Genetikus algoritmusok

- Alap GA lépések
 - generációs ciklus
 - szülők: sztoch. szelekció (szerencsekerék)
 - utódok: szülők sztoch. választása, rekombináció/crossover/ mutáció, értékelés



- Új populáció: az utódok
- megállási feltétel --- eredmény

Evulúciós stratégiák

■ ES

- Rechenberg, Schwefel '60 opt. módszer
- individum valós számokkal n pont (n dim) szórásokkal (normál eloszlás)
- μ -ből λ -utód $\mu \ll \lambda$ ($\mu + \lambda$) ES

■ ES lépések

- inicializálás μ pont szórásokkal
- értékelés célfüggvény = fitness

Evulúciós stratégiák

- ES lépések
 - generációs ciklus
 - szülők sztoch. választása (visszatevéses)
 - rekombináció pl.
 $(a\ b\ c\ d)\ (e\ f\ g\ h) \longrightarrow (a\ f\ g\ d)$ +szórás
 - mutáció pl. $\sigma_k' = \sigma_k * \exp(t1 * N(0,1))$
 $x_j' = x_j + \sigma_j' * N(0,1)$
 - értékelés
 - determinisztikus szelekció (μ legjobb - elit)
 - megállási felt., eredmény

Evulúciós programozás

■ EP

- C.J. Fogel, Owens Walsh '60 D. Fogel 92
- \sim ES, de csak mutáció (faj alkalmazkodása)

■ EP lépések

- inicializálás ($\mu > 200$), értékelés: fitness=célf.
- Generációs ciklus (μ utód)
 - replikáció: másolat
 - mutáció: pl. $x_j' = x_j + \text{fitness}(x_j)^{0.5} * N(0,1)$
 - sztochasztikus szelekció μ legjobb (győzelmek sz.)
 - megállási feltétel, eredmény



Genetikus programozás

■ GP

- Koza 1992, programkód aut. előállítás
- individum: generált program (fa-struktúra, változók, fg.-ek, konstansok), LISP
- programnyelv (műveletek, út., változó, ..)
- aut. definiálható függvények
- fitnessz érték: teszt halmazon helyes találat/hiba
- stratégiai par.: fa-srtruktúra szintek száma, ..

Genetikus programozás

■ GP lépések

- inicializálás rekurzív fa-strukturák
- értékelés fitness értékek (teszthalmazon)
- generáció ciklus
 - művelet választás: crossover, mutáció adott valószínűséggel

$(*a(+b c d) (-c 2)) \quad (*11 (+a b))$

$(*a(+b c d) (+a b))$

- értékelés, megállási feltétel, eredmény



Evolúciós algoritmusok

- Alkalmazási példák
 - fuzzy szabályozó előállítás
 - osztályozó r. generálás
 - multimodál fg. optimalizálás
 - áramkör tervezés, ...
 - csoportosítás (klaszter, repülő útvonal)
 - paraméter meghat. (NH, SZR, FR)
 - piac szegmentálás, kávépiac opt. árai, széria nagyság, prognózis, ...

Evolúciós algoritmusok

- Alkalmazási példák: géptelepítés tervezés
 - n gép, n munkahely, gép install. költs., anyag szállítási költség min.
 - jelölés C_{ij} i . hely, j . gép inst. költség
 B_{jm} $j - m$. gép közti anyag sz.
 A_{ik} $i - k$. hely közti anyag sz. költ.
 $f(i)$ a permutáció i . helyén a gép sor.
 - Célfg. $\sum C_{if(i)} + \sum \sum A_{ik} * B_{f(i)f(k)}$ ----- min

Evolúciós algoritmusok

- Géptelepítés

- (1+100) ES -el

- individuum (n=7)

4 3 5 6 2 1 7

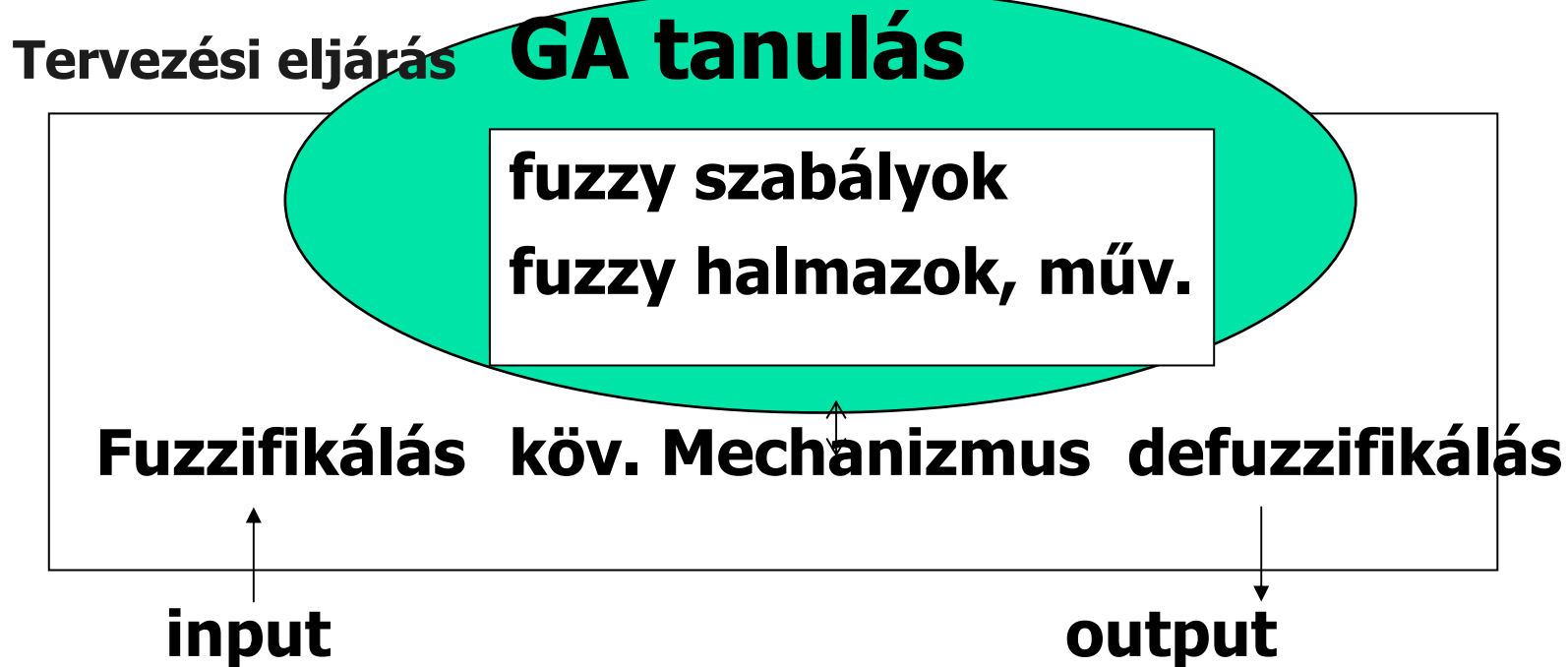
3. helyen 5. gép

- replikáció: 100 másolat

- mutáció 1-2 helyen gépcseré

Evolúciós algoritmusok

Alkalmazási példák: fuzzy szabályozó gen.



Alapgondolat : Hibrid GA - ES algoritmus

Evolúciós algoritmusok

- Alkalmazási példák: fuzzy szabályozó gen.
 - Tudásbázis:
 R_i : IF x_1 is A_{i1} AND ... x_n is A_{in} THEN y is B_i
 A_{ij} (a_{ij}, b_{ij}, c_{ij}) háromsz. term (n fuzzy halmaz)
 - individum (szabály): $C=C1C2$
 $= (c_1 c_2 \dots c_n$
 $a_{i1} b_{i1} c_{i1} a_{i2} b_{i2} c_{i2} \dots a_{in+1} b_{in+1} c_{in+1})$
 - inicializálás t példa t szabály (vázlatosan)
szabályonként n term

Evolúciós algoritmusok

- Alkalmazási példák: fuzzy szabályozó gen.
 - **Fitness: max**
 $Z(R_i) = R_i$, példák átl. kompatibilitási foka *
pozitív példák átl. " " *
büntető fg. (negatív példák $< \omega$) *
 R_i legkisebb elhelyezkedési interakció fok
 - **műveletek**
mutáció: C1 term cserék, C2 $c_k' = c_k + \Delta \text{ int.}$
crossover: C1C2 részek együtt mozognak
(1+1) ES: helyi opt. a legjobb szabálynál
C2 mutáció (term tuning)
 $\sigma_k' = \sigma_k * s_i, x_j' = x_j + s_i * \sigma_j'$



Evolúciós algoritmusok

- Fuzzy szabályozó gen. lépések
 - Hibrid GA - ES (m szabály generálás)
 - Szabálybázis egyszerűsítés (GA újabb fitnessz)
 - Szabálybázis optimalizálás (GA újabb fitnessz)



Evolúciós algoritmusok

Pl. multimodál fg.: $x_1, x_2 \in [-1, 1]$

$$F(x_1, x_2) = x_1^2 + x_2^2 - \cos(18 x_1) - \cos(18 x_2)$$

Teszt: 1681 adatpár,
hibrid GA (50 gener.) ES (25 gener.)
egyszerűsítés (500 gener.)
optim. (1000 gener.)

Eredmény: 232 szabály 0.19 a négyzetes hibák összege

O. Cordón, F. Herrera: A Hybrid Algorithm-Evolution Strategy Process for Learning Fuzzy Logic Controller Knowledge Bases In Herrera, Verdegay: Genetic Algorithms and Soft Computing Physica Verlag 1996. pp. 250-278.



Evolúciós algoritmusok

2. Alaptechnikák



EA alapternikák

- Reprezentáció, ábrázolási forma
- Szelekció
- Rekombináció
- Mutáció
- Visszahelyettesítés
- EA ciklus kialakítás



EA alapternikák

Ábrázolási forma (Változatos forma)

- Valós (egész) vektor $E = (x_1, x_2, \dots, x_n)$
- Permutáció $E = (\pi_1, \pi_2, \dots, \pi_n)$ i. pozíció: π_i objektum

$$\begin{array}{c} \pi_1, \pi_2, \dots, \pi_n \\ x_1, x_2, \dots, x_n \end{array}$$

- Bináris vektor

genotípus – fenotípus(szing)

1011101011010 x_1, x_2, \dots, x_n

x_i

kódolás: Gray-kód (Hamming távolság = 1)



EA alaptechnikák

Szelekció

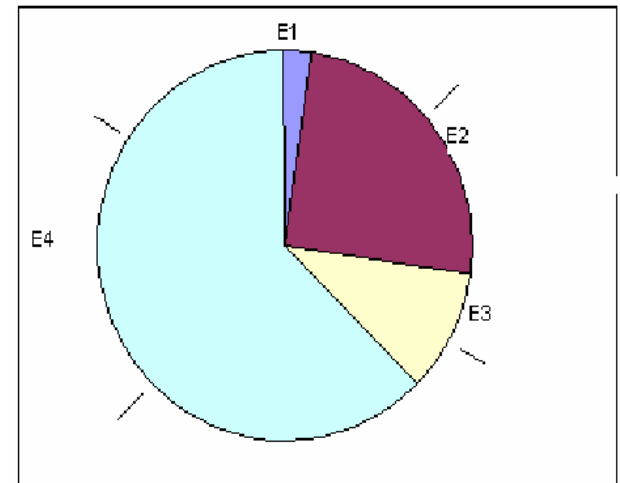
- Vizsgálható: pl. fitness értékek változása, varianciája
- Szelekciós állomány (selection pool), szülők állománya (matching pool) – köztes populációk
- **Rulett szelekció** (fitness arányos)
szelekciós valószínűség: $p(E_i) = f(E_i) / \sum f(E_i)$
rulett: $E_i \rightarrow p(E_i)$ ív egy szülőhöz μ -ször ismétlés
sok ismétlődés – szupermegoldások, javítás:
fitness skálázás (lineáris, logaritmikus, exp.)

EA alaptechnikák

Szelekció

- **Sztochasztikus univerzális mintavétel (fitness arányos)**
 $E_i \rightarrow \mu * p(E_i)$ ív egy szülőhöz (várható másolatok száma)
 μ számú mutató

Egyed	Várható másolatok száma	Kiválasztott egyedek száma
E1	0.1	0
E2	1.0	1
E3	0.4	1
E4	2.5	2





EA alapternikák

Szelekció

- **Versengő szelekció** (tournament sel.)
egyedek sorrendjét használja fel
 1. *tour* paraméter: *tour* számú egyedválasztás (rnd)
 2. legjobb kiválasztásaismétlés μ -ször (változatosság megmarad!)
- **Csonkolásos szelekció** (truncation sel.)
csak a legjobbakat választja ki
 1. növekvő sorrend fitnessz szerint
 2. P legjobb T-ed részét kijelöljük, μ -ször választás (rnd)

EA alapter technikák

Szelekció

- **Lineáris sorrend alapú szelekció** (lin, ranking sel.)
 1. növekvő sorrend fitness szerint
 2. sorszám osztás (1 a legrosszabb)
 3. egyed választás valószínűség szerint (lineárisan sorszám függő):
 $\eta \in [0, 1]$. $p_1 = \eta / \mu$ a legrosszabb egyed val. ($p_\mu = (2 - \eta) / \mu$)

Legyen J a fitness értékek alapján növekvőbe rendezett szelekciós halmaz.

$$S_0 = 0$$

For $i=1$ **to** μ **do**

$$S_i = S_{i-1} + p_i \quad \mathbf{od}$$

For $i=1$ **to** μ **do**

$$r = \text{Rnd}; \quad Ei' = J_z \text{ ha } S_{z-1} \leq r < S_z \quad \mathbf{od}$$



EA alapternikák

Rekombináció

Szülők állományából 1-több szülő - utód(ok)

Szülők által kijelölt keresési térben generál utódot.
örökölt tulajdonságok, változatosság fenntartás

- **Diszkrét rekombináció**

általános művelet, több típusra

keresési tér: néhány diszkrét érték (hiperkocka csúcsok)

(x_1, x_2, \dots, x_n) , (y_1, y_2, \dots, y_n) és (u_1, u_2, \dots, u_n) :

$$u_i = a x_i + (1-a) y_i \quad (i=1, 2, \dots, n) \quad a \in \{0, 1\}$$



EA alapternikák

Rekombináció

- **Egész és valós változók rekombinációja**

a hiperkocka „több” pontja választható

- köztes rekombináció: a hiperk. bármely pontja növelhető a kocka

$$u_i = a x_i + (1-a) y_i \quad (i=1,2,\dots,n) \quad a \in [-h, 1+h] \text{ (rnd!)}$$

- lineáris rekombináció: a hiperk. egy egyenesre, h. síkjára (mint a köztes rekombináció, de a állandó.)

EA alapternikák

Rekombináció

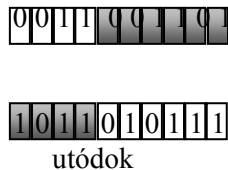
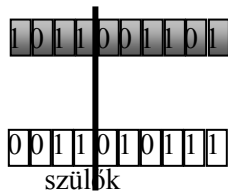
■ Bináris sztringek rekombinációja

GA genotípus szint: keresztezés (crossover)

változók bitpozíció határát nem figyeli Gray-kód!

Keresési tér: L-elemű bitsorozatok tere

- egyponos keresztezés





EA alapternikák

Rekombináció

- **Bináris sztringek rekombinációja**

- többpontos keresztezés
 - véletlen keresztezési pontok, növekvő sorrendben
- uniform keresztezés (diszkrét rek.)
- keverő keresztezés (shuffle crossover)
 - mindkét szülőnél véletlen bitpozíció keverés
 - egypontos keresztezés
 - eredeti sorrend visszaállítás

EA alapternikák

Rekombináció

- Permutációk rekombinációja**

helyes permutációt megtartó transzformációk

Uniform	Szülő1	1	2	3	4	5	Véletlen bitmaszk!
	szülő2	4	3	5	2	1	
	bitmaszk	1	0	0	1	0	
sorrendalapú rekombináció	közbülső állapot	1	0	0	4	0	Minél több sorszám egyezés legyen!
		0	3	5	0	1	
	utód1	1	3	5	4	2	
	utód2	2	3	5	4	1	

EA alapternikák

Mutáció

X szomszédsági környezetében új pont

$$Nhd(x, \varepsilon) = \{y \in S / d(x, y) < \varepsilon\}$$

Változtatás: 1-több változó, ε csökkenő, pontosság szélesebből – kisebb környezet felé

■ Valós/ egész változók mutációja

$$z_j = x_j \pm \varepsilon_j * \delta, \quad \delta = 2^{-k\alpha} \quad \alpha \in [0, 1]$$

pl. $\varepsilon_j 0.1 \rightarrow 10^{-7}$, $k = 4, 5, \rightarrow 20$

a változók számát p_m befolyásolja

EA alapternikák

Mutáció

- **Valós/ egész változók mutációja**

több változót érintő mutáció:

$$z_j = x_i \pm \varepsilon_j * \delta * \gamma^{|i-j|}, \quad \delta = 2^{-k \alpha} \quad \alpha \in [0, 1] \quad \gamma < 1$$

$$j = (i, i-1, i-2, \dots), \quad i = 1, 2, \dots, n$$

x_i -től távolodva csökken a változás

- **Bináris változók mutációja**

bitenként:

$$z_i = \begin{cases} x_i & \text{ha } Rnd > p_m \\ 1 - x_i & \text{ha } Rnd \leq p_m \end{cases}$$

EA alaptechnikák

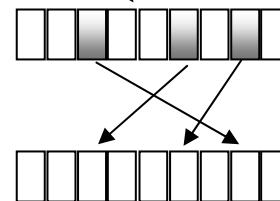
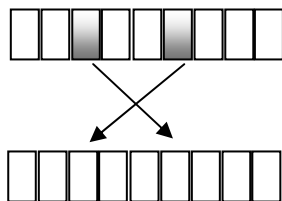
Mutáció

- **Permutációk mutációja**

változó: pozíción található érték

sorrend változtatás p_m alapján

- beszúrás, csere, inverz, scramble





EA alaptechnikák

Visszahelyezés (reinsertion)

- Utódképzési ráta(UR) (generation gap):
utódok száma hányszorosa μ -nek
- Visszahelyezési ráta (VR)
lecserélt egyedek aránya P-ben
 - $VR=1$ – teljesen új P
 - $VR<1$ – adott számút cserélünk (pl. elit szelekció)
 - $UR<VR$ – minden utód bekerül P-be
 - $UR>VR$ - utódok egy része kerül csak P-be
szelektálni kell: szelekciós műveletekkel
- Steady-state EA: 1-2 utód



EA alaptechnikák

EA ciklus kialakítás

Stratégiai paraméterek:

- A populáció mérete (az egyedek száma),
- A rekombináció alkalmazásának p_r valószínűsége,
- A mutáció alkalmazásának p_m valószínűsége
- Az utódképzési ráta értéke
- A visszahelyezési ráta értéke.

Kezdő populáció kialakítása:

előző feladatmegoldás felhasználása+véletlen egyedek,
módosított előző eredmények, többszörös újraindítás,
véletlenszerűen kialakítás



EA alaptechnikák

EA ciklus kialakítás

Megállási feltétel:

- max. generációszám elérése
- max. futási idő elérése
- adott idő alatt nem javul a megoldás minősége
- hasonlók az egyedek
- előre adott érték megközelítése
- P minősége megfelelő (mérték: célfüggvény értékek szórása/átlaga, legjobb- legrosszabb eltérése P-ben)
- kombinációk (nem teljesül, ...)



EA alaptertechnikák

EA ciklus kialakítás

Fitness kiértékelés:

szerep: egyedek sorrendje, keresési tér struktúrálás

célfüggvény adaptálás általában

bináris eset: bináris – valós kódolás

sorrend esetén:

 sorrend – sorszámok – fitness a sorszámra

 több célfüggvény – Pareto dominancia alapján

nincs célfüggvény: elvárások/korlátok megfogalmazása,
kombinálás – összetett fitnessfüggvény kialakítás



EA alaptechnikák

EA ciklus struktúra:

stratégiai paraméterek beállítása

kezdőpopuláció kialakítása

fitness kiértékelés

Repeat

szelekció, mutáció

fitness kiértékelés

visszahelyezés

until megállási feltétel teljesül



Evolúciós algoritmusok

3. Általános, standard módszerek



3. Általános, standard módszerek

Témakör

- Standard és származtatott módszerek
- GA (alap, változatok)
- ES és változatok
- EP és változatok
- Terjeszkedő (scatter) keresés



Alap GA

Jellemzők:

- Reprezentáció: bitsztring - változókhoz egy-egy rész
- Fitness kiértékelés: bitsztring részek dekódolása, transzformálása
- Szelekció: rulett szelekció
- Rekombináció: keresztezés – egypontos, uniform p_r valószínűséggel alk. ($\sim p_r=0.6$), fő művelet
- Mutáció: bitmutáció, ált. $p_m = 0.001$
- Visszahelyezés (UR=1, VR=1)



Alap GA

Jellemzők:

- EA ciklus
 - Kezdőpopuláció: véletlen (30-500 egyed)
 - Megállási feltétel: max. generációszám /futásidő
 - Stratégiai paraméterek (μ , p_m , p_r , bitsztring hossz, a_i , b_i)
- Főbb lépések
Kezdő P, fitness kiértékelés
Repeat
 szelekció, rekombináció, mutáció, visszahelyezés
Until megállási felt. teljesül



Alap GA

Bináris kódolás GA-nál

A valós egyed: (x_1, x_2, \dots, x_n)

x_i eleme a $[b_i, c_i]$ intervallumnak $i=1, 2, \dots, n$

$[b_i, c_i]$ felbontása részekre, ahol $1/db_i$ rész kellő pontosság már és

db_i egy kettő hatvánnyal egyenlő (pl. $db_i \sim 30$ akkor 2^5 és $h_i = 5$)

Kezdőpont: b_i --- 00000...000 (h_i db)

Végpont: c_i -----11111...111 (h_i db)

(ez ritkán azonos a standard bináris kódolással)

Egyed hossz: $\sum h_i$

Dekódolás: intervallumonként

$$D(a_{i1}, \dots, a_{ih_i}) = b_i + \frac{c_i - b_i}{2^{h_i} - 1} \left(\sum_{z=1}^{h_i} a_{iz} (h_i - z + 1) * 2^{z-1} \right) = x_i$$



Alap GA

Pl.: $-1 \leq x \leq 2$ valós szám és 0.1 pontosságot szeretnénk.

- 30 egyenlő részre kell bontani
- Legközelebbi 2 hatvány 2^5 5 hosszúságú bitszring kell
- Kezdőpont 00000 végpont 11111
- Pl. 11001 értéke 1.4

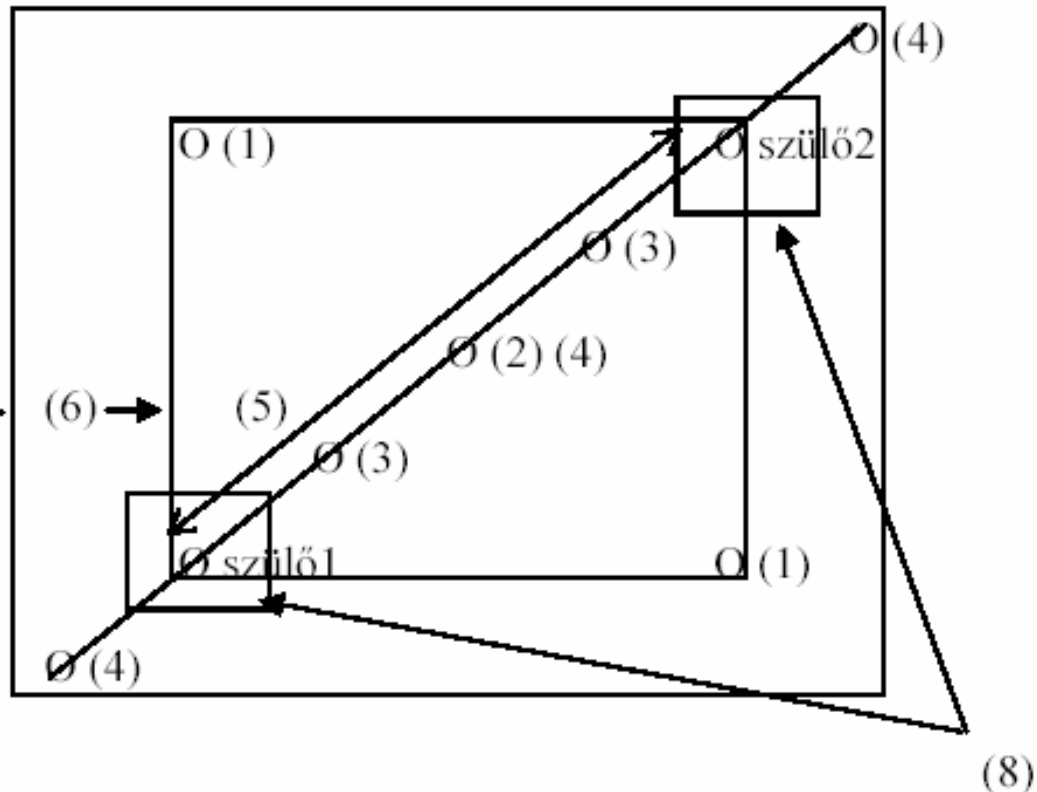
Pl. $0 \leq x \leq 63$ egész szám, melyet pontosan akarunk használni

- 64 egyenlő részre kell bontani
- Legközelebbi 2 hatvány 2^6 6 hosszúságú bitszring kell
- Kezdőpont 000000 végpont 111111

Alap GA

Változatok: folytonos GA – csak fenotípus szint.

diszkrét rekombináció (1-es)
köztes rekombináció (2-es)
aritmetikai keresztezés (3-as)
lineáris keresztezés (4-es)
véletlen köztes rekombináció
(5-tel jelölt egyenes pontjai) (7) →
globális köztes rekombináció
(6-tal jelölt téglalap pontjai)
kiterjesztett köztes
rekombináció (7-tel jelölt)
fuzzy rekombináció (8-al jelölt)





Alap GA

Változatok: steady-state GA – q utód (1-2)

Kezdő populáció generálása: $t=0, P(t) = \{E1, E2, \dots, En\}$
fitness kiértékelés $\Phi(Ei) \ i=1,2,\dots,n$

Repeat

$t=t+1$

Repeat

Szelekció: két szülő választás, (lin. sorrend alapú szel.)

Rekombináció (utódok: Ei', Ej'), Mutáció ($\Phi(Ei'), \Phi(Ej')$),

Duplikált utód törlése Fitness kiértékelés ($\Phi(Ei'), \Phi(Ej')$)

Until az utódok száma $\geq q$

Visszahelyezés: $P(t) = P(t-1)$, rendezés, törlés, + q utód

Until megállási feltétel teljesül



Általános ES

Jellemzők:

- Reprezentáció: $E = (x_1, \sigma_1, \dots, x_n, \sigma_n)$.
 σ_n (lépésköz): állandó / különböző és önadaptív
- Fitness kiértékelés: \sim célfüggvénnyel azonos
- Szelekció: véletlen egyed választás (ismétléssel)
- Rekombináció: két szülő – egy utód
pl. diszkrét rekombináció + szórások: átlag
(köztes, globális köztes)



Általános ES

Jellemzők:

■ Mutáció

- Azonos szórások esetén (hipergömb pontjai x_i körül)

$$\sigma' = \sigma \exp(\tau_0 N(0, 1))$$

$$x_i' = x_i + \sigma' N_i(0, 1). \quad \tau_0 = n^{-0.5}$$

- Változónként eltérő szórások (hiperellipszoid pontjai x_i körül)

$$\sigma_i' = \sigma_i \exp(\tau' N(0, 1) + \tau N_i(0, 1))$$

$$x_i' = x_i + \sigma_i' N_i(0, 1).$$

$$\tau' \sim (2n)^{-1/2} \text{ és } \tau \sim (2n^{1/2})^{-1/2}$$



Általános ES

Jellemzők:

- **Visszahelyezés:** két változat (μ egyed λ utód)
 - (μ, λ) –szelekció új P: a legjobb μ számú utód
diszkrimináló: a rosszabb utódnak nincs esélye
előny: állandóan változik, mutáció lépésköz mindig változik
 $\mu / \lambda \sim 1/7$ és $\mu \sim 15$
 - $(\mu + \lambda)$ –szelekció új P: a legjobb μ számú a szülők+
utódok közül
elit szelekció: a legjobbak tovább „élnek”



Általános ES

Jellemzők:

- EA ciklus

kezdő populáció generálása: $t=0, P(t) = \{E1, E2, \dots, E\mu\}$
fitness kiértékelés $\Phi(Ei)$ $i=1,2,\dots,\mu$

Repeat

$t=t+1$

Rekombináció (utódok: $E1', E2', \dots, E\lambda$), Mutáció ($E1', E2', \dots, E\lambda$)

Fitness kiértékelés ($\Phi(Ei')$ $i=1,2,\dots, \lambda$),

If (μ, λ) szelekció **then** $P(t) = \mu$ legjobb utód **fi**

If $(\mu + \lambda)$ szelekció **then**

$P(t) = \mu$ legjobb $P(t-1) \cup \{E1', E2', \dots, E\lambda\}$ közül **fi**

Until megállási feltétel teljesül



Általános ES

Változatok: korrelált mutáció alkalmazása

legjobb irányba történő módosítás: rotációs szögek

egyed: $E_i = (\underline{x}_i, \underline{\sigma}_i, \underline{\alpha}_i)$, $\underline{\alpha}_i$ szintén változik (~ 5 fok)

Változatok: (1+1) ES

1 egyed (vektor), rekombináció: másolat, azonos szórások

Rechenberg-szabály: szórás megválasztására

sikeres mutációk aránya $\sim 1/5$ –nél megfelelő

$< 1/5$: csökkenteni kell σ -t

$> 1/5$: növelni kell σ -t



Standard EP

Jellemzők:

- Reprezentáció: valós vektor + $2n$ mutáció paraméter
- Fitness kiértékelés: \sim célfüggvény,
transzformálás >0 -ra: $\Phi(E_i) = F(f(E_i), z_i)$.
- Rekombináció: másolat
- Mutáció: dinamikusan változó szórás, k_i , z_i paraméterek

$$\sigma_i' = \sqrt{k_i * \Phi(E_i) + z_i}$$

$$x_i' = x_i + \sigma_i' * N_i(0,1)$$



Standard EP

Jellemzők:

- Visszahelyezés: verseny a szülők és utódok közt
„sztochasztikus szelekció” ~ versengő szelekció
minden szülőhöz, utódhoz súly rendelés
lépések: - minden E_i -hez q egyed választás (rnd)
- súly = győzelmek száma (jobb a $\Phi(E_i)$)
- új P: μ egyed a legjobb súlyokkal
($q \sim 0.05$ μ és $\mu > 200$)



Standard EP

- EA ciklus

kezdő populáció generálása: $P(t) = \{E1, E2, \dots, E\mu\}$

fitness kiértékelés $\Phi(Ei) \ i=1,2,\dots,\mu$

Repeat

$t=t+1$, Rekombináció (utódok: $E1', E2', \dots, E\mu'$)

Mutáció ($E1', E2', \dots, E\mu'$), Fitness kiértékelés ($\Phi(Ei')$ $i=1,2,\dots, \mu$)

Visszahelyezés:

$P(t-1) \cup \{E1', E2', \dots, E\mu'\}$ súlyok meghatározása, rendezés

$P(t) =$ az első μ egyed a rendezettek közül.

Until megállási feltétel teljesül



Standard EP

- **Változatok: önadaptív, meta-EP**

meta-EP \sim ES

egyed: $E = (x_1, \sigma_1, \dots, x_n, \sigma_n)$,

mutáció:

$$\sigma_i' = \sigma_i + \alpha \sigma_i N(0, 1)$$

$$x_i' = x_i + \sigma_i' N(0, 1).$$

α stratégiai szerepe (kisebb, nagyobb változások)



Terjeszkedő (scatter) keresés

Alapgondolat: *korábbi megoldások lineáris kombinációi közt keressük a jobb megoldásokat*
megoldások populációja – de nem evolúciós műveletek

Főbb lépések:

1. Kezdő megoldás halmaz generálása, és egy megoldás halmaz (Refset: reference set) elkülönítése.
2. Új megoldások képzése a Refset elemeinek kombinálásával.
3. Az új megoldások javítása helyi kereséssel.
4. A legjobb új megoldások kiválasztása és a Refset-be illesztése.
5. Folytatás a 2. ponttól, ha a Refset még változott, vagy max. generációszámot nem érte el.



Terjeszkedő (scatter) keresés

Részletek:

- Kezdő P: generátor programmal (minden régióból..)
- Javító eljárás: helyi kereső elj. minden egyednél
- Refset módosítás ($\text{Refset} = \text{Refset}' + \text{Refset}2$):
Refset1 – a legjobbak, Refset2 – a legrosszabbak
vizsgálat, hova helyezhető (Resfset1-2)
- Részhalmaz generálás: $2, 3, 4, \dots | \text{Refset} |$ elemű halmazok
pl. kételemű +a legjobb kimaradó, ...
- Megoldás kombináló eljárás: a részhalmaz egyed-párok
lineáris kombinációi (többféle típus)



Evolúciós algoritmusok

5. Genetikus programozás (GP)



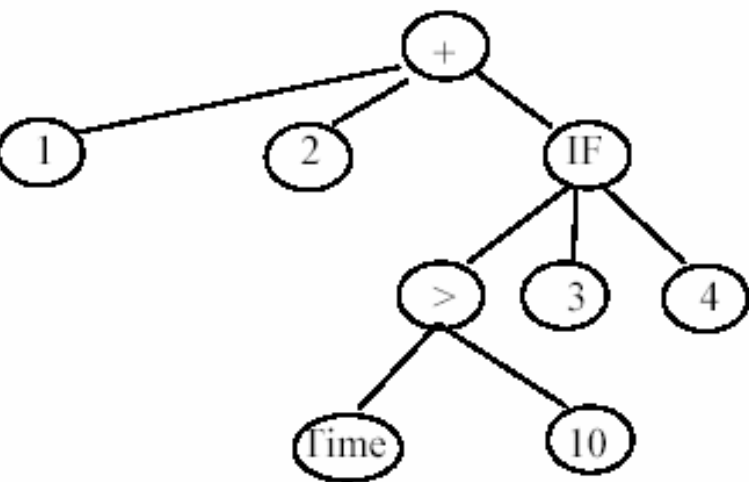
Genetikus programozás

- Alapfeladat
- GP alapkoncepció
- Az általános GP módszer
- Fejlettebb GP technikák
- Példa ADF-el
- Alkalmazások

Genetikus programozás

■ Alapfeladat

- Automatikus programírás
- Koza (1992): LISP programok írása
- Programfa írásmód
(*+1 2 (IF(> Time 10) 3 4)*).





Genetikus programozás

■ GP alapkonceptió

Főbb lépések, definiálni kell:

1. a végpontok halmazát (*terminal set*)
2. a függvények halmazát (*function set*)
3. a fitnessfüggvényt
4. a stratégiai paramétereket
5. a megállási feltételt és
6. a program felépítését.



Genetikus programozás

- **GP alapkonceptió**
 - T halmaz: konstans, változó, függvény()
 - F halmaz: aritmetikai, matematikai, logikai, szelekciós, ciklus utasítások, spec. függvények (problémás: osztás, ciklus, különböző adattípusok)
 - Fitnessfüggvény:
hibafüggvény / sikeres tesztek száma
több cél kezelés: büntető fg. / Pareto dominancia



Genetikus programozás

- **GP alapkonceptió**
 - Stratégiai paraméterek: programhossz, p_r , p_m , $|P|$ (több ezer), szintek száma (max. 7)
 - Megállási feltétel: pontos/közelítő eredmény, max generációszám (50)
 - Program felépítés (egy egyed): egy fa (főprogram), több fa (eljárások/ akciók)



Genetikus programozás

■ Az általános GP módszer

- Kezdőpopuláció: véletlen, irányított véletlen (half-ramping). Csomópontokban F elemek, leveleknél T elemek
- Szelekció: fitness arányos, versengő szelekció, versengő + Pareto dominancia (több célnál-
eddig legjobb lista)
- Rekombináció: egy pontos keresztezés; (0.9 val.
belső csomópontnál kijelölés), vagy reprodukció



Genetikus programozás

■ Az általános GP módszer

■ Mutáció:

- részfa lecserélés (rnd),
- pont mutáció (F elem csere),
- részfa képzés (önálló fa lesz a részfa)
- rövidítés(részfa helyett T elem),
- konstans mutáció,
- szisztematikus konstans mutáció (input változó – konstans csere)

Genetikus programozás

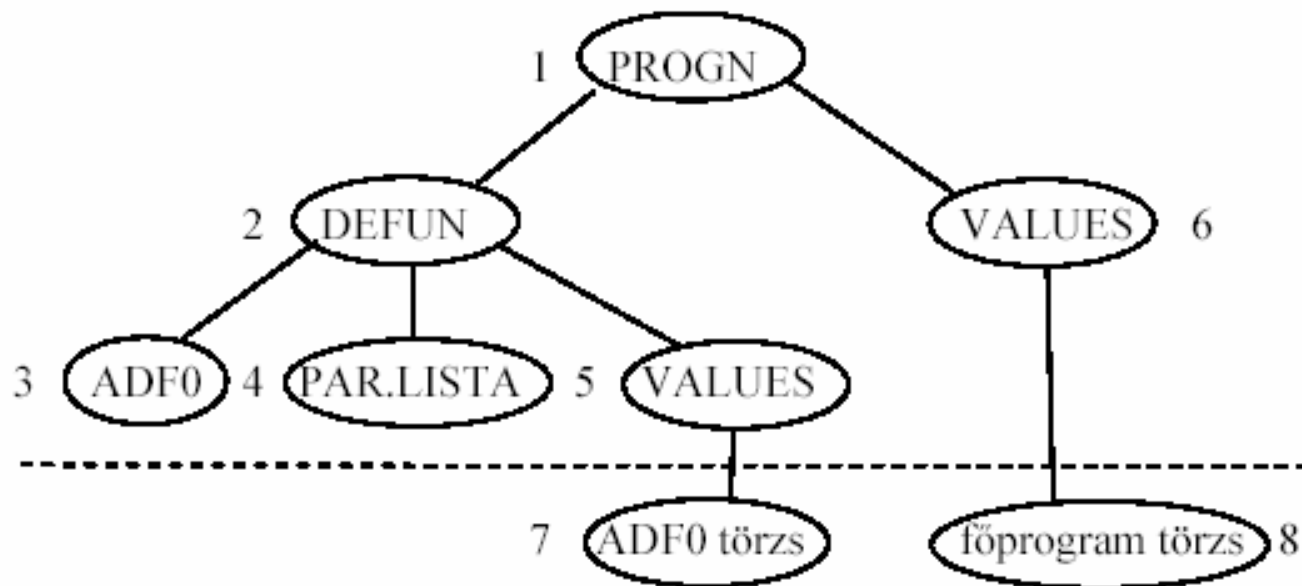
GP definíció

Cél	Program megadás
Architektúra	Egy fa
Tesztadatok	Tesztadatok leírása
Végpont halmaz	$T = \{ \dots \}$
Függvény halmaz	$F = \{ +, -, *, \% \}$
fitnessfüggvény	A teszhalmazon a találatok száma
Szelekciós eljárás	Versenyző (7 elemmel)
Kezdőpopuláció kialakítása	Half-ramping
Paraméterek	Populáció méret 4000-16000
	Max. generációs szám 51
	$p_r = 0.9$
	$p_{rep} = 0.1$
	$p_m = 0.1$
	Max. szint a végeredményben 6
	Max. szint futás közben 17

Genetikus programozás

■ Fejlettebb GP technikák

- Automatikusan definiálható függvény - ADF



Genetikus programozás

■ Fejlettebb GP technikák

- Memória használat: skalár, read/write indexelt (-k,k), read(index), write(index, érték),
- Elkülöníthető tevékenységek/ akciók (multi-tree)
 - minden akciónak külön belépési pont,
 - minden akciót külön tesztelünk,
 - rekombináció, mutáció kiválasztott akcióra



Genetikus programozás

Alkalmazási példa ADF-el

Feladat: Két téglatest különbsége $D = a1 * b1 * c1 - a2 * b2 * c2$

13. generációnál a jó végeredmény:

```
(PROGN (DEFUN ADF0 (PAR0 PAR1 PAR2)
(VALUES (-(*PAR2 PAR0) (*(+ PAR0 (* PAR0 PAR1)
(% PAR2 (% PAR2 PAR2))))))
(VALUES (-(ADF0 a2 b2 c2 ) (ADF0 b1 c1 a1))))))
```

GP definíció módosítás:

F halmaz a főprogramnál $Ff = \{+, -, *, \%, ADF0\}$

T halmaz az ADF0-nál $Ta = \{PAR0, PAR1, PAR2\}$

F halmaz az ADF0-nál $Fa = \{+, -, *, \%\}$

Genetikus programozás

Alkalmazási példa: veremkezelés

- push down verem 5 tevékenységgel:

Inicializálás (*makenull*)

mutató := maxméret + 1

Üresség vizsgálat (*empty*)

üres := (mutató > maxméret)

Csúcs olvasás (*top*)

csúcs := verem (mutató)

Csúcs olvasás és törlés

törölt := verem (mutató)

(*pop*)

mutató := mutató + 1

Újadat elhelyezés

mutató := mutató - 1

(*push*)

verem(mutató) := újadat

Genetikus programozás

Alkalmazási példa: veremkezelés

- definiálások:

- T halmaz:

pl. +index memória,
verem mutató, maxméret konstans

- F halmaz: spec. függvények (read(mutató), write, mutató kezelő műveletek)

- Fitnessz: pontszám (helyes teszteredmények száma) tevékenységenként külön teszt





Genetikus programozás

GP alkalmazások

- Fuzzy osztályozó rendszer fejlesztés (biztosító, anyag felhasznál.)
- Képfeldolgozás (röntgen, MRT) pl. filter tervezés
- Tervezés (áramkör, repülő részek, auto karosszéria)
- Kereskedelem: piacmodellezés (fuzzy rendszer)
- Mesterséges élet: mozgás, ágesnek
- Művészet: videó film, jazz, 3D kép



Evolúciós algoritmusok

4. Fejlettebb EA technikák I.



Fejlettebb EA technikák

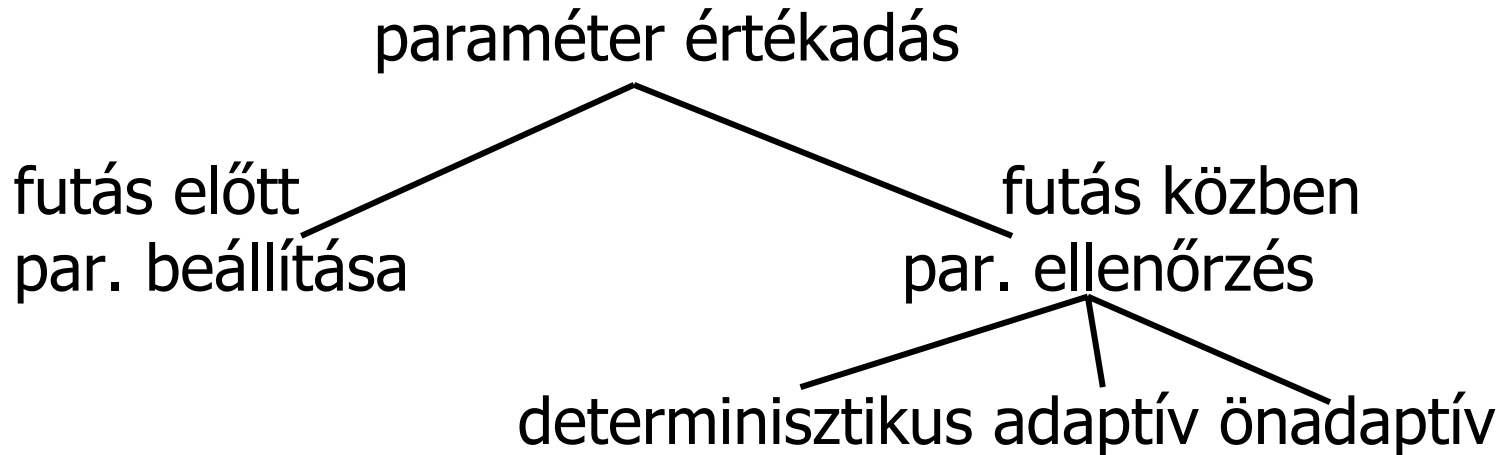
Témakörök

- Paraméterek beállítása
- Csoportosítás (niching)
 - Fitness megosztás (fitness sharing)
 - Tömörítés (crowding)
 - Ritkítás (clearing)
- Memóriát alkalmazó módszerek
 - Virtuális vesztés
 - Hangya kolónia
 - Kulturális algoritmus

Fejlettebb EA technikák

Paraméterek beállítása

- Megfelelő kombináció kiválasztás, módosítás kölcsönösen függnnek egymástól
- Beállítási technikák:





Fejlettebb EA technikák

Paraméterek beállítása

- Futás előtti par. beállítás: kézi beállítás, változatok próbálgatása
 - csak néhány kombináció próbálható ki
 - időigényes
 - bizonytalan a legjobb megtalálása
- Futás közbeni változtatás (minden változtatható!)
 - Determinisztikus par. ellenőrzés: det. szabályt alkalmaz
 - Adaptív par. ellenőrzés: visszacsatolás a hatékonyságról
 - Önadaptív par. ellenőrzés: evolúciós folyamat változtatja

Fejlettebb EA technikák

Paraméterek beállítása

- Pl. futás közbeni mutáció par. változtatás ES-nél
 - Determinisztikus: $x_i' = x_i + N(0, \sigma)$ és $\sigma(t) = 1 - 0.9 t/T$
 - Adaptív: p_s sikeres mutációk aránya (Rechenberg)

If ($t \bmod n = 0$) **then**

$$\sigma(t) = \sigma(t-n)/c \quad \text{ha } p_s > 1/5$$

$$\sigma(t) = \sigma(t-n) * c \quad \text{ha } p_s < 1/5$$

$$\sigma(t) = \sigma(t-n) \quad \text{ha } p_s = 1/5$$

else

$$\sigma(t) = \sigma(t-1)$$

fi ahol $c \in [0.817, 1]$

- Önadaptív mutáció ellenőrzés: általános ES-nél

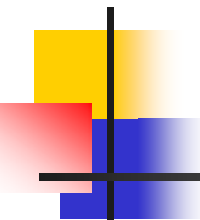


Fejlettebb EA technikák

Paraméterek beállítása

Alap GA-nál vizsgálatok:

- Önadaptív mutáció: p_m 0.001-0.15 közt (tárolják p_m -t)
- Önadaptív keresztezés: p_r változik (tárolják p_r -t) p_r -től függően
 - uniform keresztezés (mindkét szülőnél $RND < p_r$)
 - Másolás (egyik szülőnél sem $RND < p_r$)
 - csak mutáció (egy szülőnél $Rnd < p_r$)

- 
- Adaptív populáció méret: egyedhez max élettartam, amely csökken generációnként a fitness függvényében
 - Afit (átlagos fitness), HET (hátralévő élettartam), Rofit (legrosszabb f), Jofit (legjobb f), $\eta = 1/2(\text{MAXHET} - \text{MINHET})$ MINHET – MAXHET intervallumban változhat

$$HET_i = \begin{cases} \text{MINHET} + \eta \frac{\text{Rofit} - \text{fitness}_i}{\text{Rofit} - \text{Afit}} & \text{ha } \text{fitness}_i > \text{Afit} \\ 1/2(\text{MINHET} + \text{MAXHET}) + \eta \frac{\text{Afit} - \text{fitness}_i}{\text{Afit} - \text{Jofit}} & \text{ha } \text{fitness}_i < \text{Afit} \end{cases}$$

Fejlettebb EA technikák

Csoportosítás (niching)

Ha több lehetséges megoldást keresünk

Stabil csoportok kialakítása párhuzamos keresésnél.

Technikák: fitness megosztás, tömörítés, ritkítás.

■ Fitness megosztás (fitness sharing)

cél: a csoportokban megmaradjanak a legjobbak,
fitness érték csökkentés a csoportban $f_i = f_i / m_i$

(sh a sharing függvény, σ_s becsült küszöbérték (niche sugár))

$$m_i = \sum_{j=1}^{\mu} sh(d_{ij}) \quad sh(d_{ij}) = \begin{cases} 1 - (d_{ij} / \sigma_s)^\alpha & \text{ha } d_{ij} < \sigma_s \\ 0 & \text{különben} \end{cases}$$



Fejlettebb EA technikák

Csoportosítás (niching)

■ Tömörítés (crowding)

csoportok fenntartása: új egyed a hasonlót cserélheti le

■ Standard tömörítés:

csak néhány utód,

visszahelyezés: CF számú (rnd) egyed közül a leghasonlóbb szülő helyére kerül egy-egy utód

■ Determinisztikus tömörítés: standard tömörítés, de visszahelyezésnél verseny az utód és a legközelebbi, csoportbeli szülő közt. (utódonként)

■ Korlátozott versengő szelekció (standard változat)

visszahelyezés: CF számú (rnd) egyed közül a leghasonlóbb szülővel versenyez



Fejlettebb EA technikák

Csoportosítás (niching)

■ Ritkítés (clearing)

Fitness változtatás a csoporton belül, csak a legjobbak maradnak.

A mechanizmus:

- egy csoportban csak k egyed lehet
- csoporthoz tartozás: σ_c -nél kisebb a távolságuk
- k számú legjobb egyed változatlan, többi fitness értéke nulla.



Fejlettebb EA technikák

Memóriát alkalmazó módszerek

Hatékonyság növelés „problémamegoldó tudással”;
Sikeres / sikertelen lépések, ismeretek memorizálása.

Kollektív memória (EC-memory)

Használható: szelekciónál, művelet par. beállításnál, stb.

Típusok: numerikus és szimbolikus

- Numerikus:
műveletek valószínűségeivel, virtuális egyed mint valószínűségi vektor/mátrix a legjobb egyedekből
- Szimbolikus:
keresési intervallumok, szabályok tanulása



Fejlettebb EA technikák

Memóriát alkalmazó módszerek

Virtuális vesztes

- Korábbi sikertelen egyedekből levonható tanulságok
- Valószínűségi vektor bitérték választására (VL)
- Képzése: legrosszabb egyedek bitpozíció átlagai
- Aktualizálás: $VL^{t+1} = (1 - \alpha) VL^t + \alpha \Delta VL$ ($\alpha \sim 0.2$)
 ΔVL a t-dik generációban képzett VL
- Alkalmazás: egy X egyed x_i bitjénél
 $p_i = 1 - |VL_i - X_i|$ valószínűséggel kell 1 érték



Fejlettebb EA technikák

Memóriát alkalmazó módszerek

Hangya kolónia (AC: Ant colony)

- Hosszútávú memória diszkrét értékeknél (permutációk)
- Populáció egyedei „hangyák”
párhuzamosan több útvonalat bejárnak, a jó útvonalat feromonnal megjelölik, feromon intenzitás nőhet...
- Feromon csík koncepció átvétele
Állapottér leírás gráffal, (i,j) élhez τ_{ij} feromon csík érték
feromon mátrix, sikeres út aktualizálja

Fejlettebb EA technikák

Memóriát alkalmazó módszerek

Hangya kolónia

- Lépésenkénti útvonal kialakítás
a k -dik hangya p_{ij} valószínűséggel lép i -ből j állapotba
 p_{ij} függ:- az átmenet „vonzerejétől” η_{ij}
 - τ_{ij} feromon csík értéktől
 - a lehetséges átmenettől (tabu _{k} lista)
- Feromon aktualizálás: $\tau_{ij}(t) = \tau_{ij}(t-1) + \Delta\tau_{ij}$
($\Delta\tau_{ij}$: hangyák száma amelyek i - j -t sikeresen alkalmazták)
- Nincs szelekció, rekombináció, mutáció



Fejlettebb EA technikák

Memóriát alkalmazó módszerek

Hangya kolónia lépések:

Véletlen kezdőértékek a τ feromon csík mátrixnál.

Repeat

For $k=1$ to $|P|$

Kezdőállapot választás; legyen i .

Repeat

Minden j -re η_{ij} kiszámolása. Egy állapot átmenet kiválasztása $tabu_k$ lista bővítése az új állapot átmenettel.

Until k -dik megoldás kész

endfor

For a hangyák minden i - j állapot átmenetére.

$\Delta\tau_{ij}$ kiszámítása. Feromon mátrix aktualizálás.

endfor

Until megállási feltétel teljesül.



Fejlettebb EA technikák

Memóriát alkalmazó módszerek

Kulturális algoritmus

- A kultúra által motivált modell, EA újabb elemekkel - egy mechanizmussal az ismeretek kinyerésére
- Két populáció: - szociális P (pl. normál ES)
- hihető BLF (ismereteket tárol)
BLF részek: -SZ legjobb példák halmaza
- N intervallum halmaz
- Műveletek: szelekció BLF-hez P-ből (Accept), BLF aktualizálás (Adjust), paraméter módosítás önadaptíven (influence)



Fejlettebb EA technikák

Memóriát alkalmazó módszerek

Kulturális algoritmus lépések:

Kezdőpopuláció kialakítása: $t=0$, $P(t)$, $BLF(t)$.

Fitness kiértékelés: $P(t)$ -n.

Repeat

$t=t+1$

$P(t)$ evolúciója az *influence* művelettel.

$BLF(t)$ aktualizálása:

Accept ($P(t)$),

Adjust ($BLF(t)$).

Until megállási feltétel teljesül.



Evolúciós algoritmusok

4. Fejlettebb EA technikák II.



Fejlettebb EA technikák

Témakörök

- Párhuzamos EA változatok
 - Globális modell
 - Regionális modell
 - Konkurens modell
 - Kooperáló modell
 - Lokális modell



Fejlettebb EA technikák

- Párhuzamos EA változatok
 - Nehéz problémák megoldására
 - Multiprocesszoros rendszerek, nagyobb populáció, elfogadható futási idő párhuzamos számításokkal
 - Különböző modellek: populáció struktúrák, szelekció művelet alapján
 - Globális: párhuzamos számítások
 - Regionális: régiók, migráció
 - Regionális modell változatok: koevolúció (szimultán evolúció) - konkurens, kooperáló
 - lokális



Fejlettebb EA technikák

■ Globális modell

- Populáció struktúra változatlan
- Párhuzamosan végrehajtható:
rekombináció, mutáció, célfüggvény számítás
- Master-slave processzor struktúra:
 - master osztott memóriával tárolja P-t
 - slave-ek számolnak



Fejlettebb EA technikák

- **Regionális modell (island, coarse grained modell)**
 - Egyenlő nagyságú alpopulációk (régiók)
 - Alpopulációk önálló EA-ként működnek, egy-egy mikroprocesszorral
 - Nagy populáció kialakításhoz: migrációs művelet
 - Hatékonyságát befolyásolja:
alpopulációk száma /mérete
topológia, migrációs egyedek száma
egyedkiválasztás mód



Fejlettebb EA technikák

■ Regionális modell

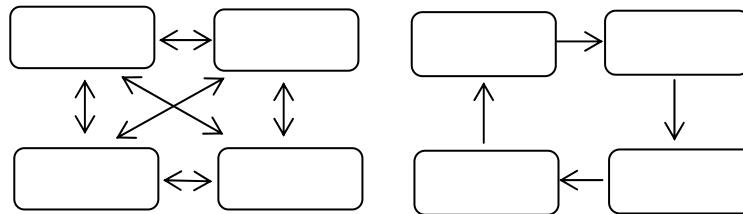
- alpopulációk száma /mérete
tapasztalati képlet: alpopulációk száma= $n^{0.5}$
alpop. méret= $20+n/10$
(processzorszám fontos)
pl. $n=150$ esetén 13 régió, 35-35 egyed (420)
- Alpopulációk közti topológia: irányított gráf a populációk közt, migráció használja.
Lehet:
teljes kapcsolat, egydimenziós gyűrű (+távolság),
szomszédsági (2-dimenziós rács + távolság)

Fejlettebb EA technikák

■ Regionális modell

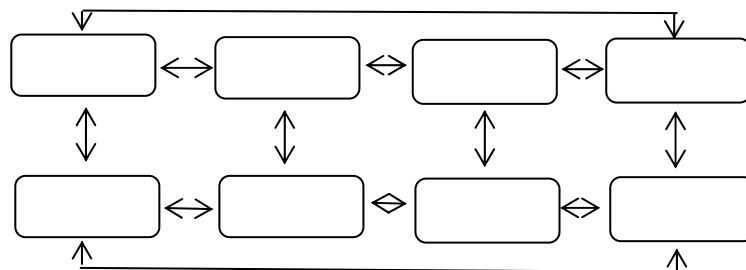
Topológia:

teljes



gyűrűs

szomszédsági





Fejlettebb EA technikák

■ Regionális modell

■ Migrációs ráta és periódus

egyedek száma amit kicserélünk, cserék gyakorisága.

tapasztalati képlet:

Migrációs ráta= régióméret 20%-a /*egyedszám

Migrációs periódus= régióméret /*generációszám

■ Egyedek kiválasztása, lecserélése: migrációs műv.

1. egy régióhoz - régiók kiválasztása (topológia alapján)
2. egyedek kiválasztása régiónként → migrációs halmaz
3. migrációs halmazból egyedcsere a régiónál

Fejlettebb EA technikák

Kezdőértékek a P_1, P_2, \dots, P_k alpopulációknál.

Fitness kiértékelés a P_1, P_2, \dots, P_k alpopulációknál.

Repeat

For $j=1$ **to** *migrációs_ periódus*

Párhuzamos végrehajtás minden P_i -re ($i=1,2,\dots,k$):

Szekvenciális_EA(P_i)

endfor

For $i=1$ **to** k

For i minden j szomszédjára

Migráció (P_i, P_j)

/ csak két populáció közt!*

Fitness kiértékelés P_i -nél

endfor

endfor

until megállási feltétel teljesül



Fejlettebb EA technikák

■ **Regionális modell**

■ Felhasználás:

- Nagy populáció kialakításhoz,
nagyobb keresési tér lefedés (megfelelő izolálás)
- Különböző stratégiai paraméterek alkalmazása
sikeresebbek befolyásolják a többi EA-t
- Ismeretlen stratégiai par. keresése
(sikerességet mérni kell!)



Fejlettebb EA technikák

■ **Konkurens modell (competing subpopulation)**

- Regionális modell változat – különböző stratégiákkal
- Sikeresség függvényében változhat:
régió méret, számítások biztosítása a sikeresek javára.
- Konkuraló régiók rangsorolása sikeresség alapján. Pl.:
 1. reprezentáns egyedek választása régióként
 2. lineáris rendezés alapú fitness kiértékelés
 3. Régiók értékelése: reprez. egyedeik átlagos fitness értéke
 4. rendezzük , majd sorszámozzuk a régiókat.
 5. **pozíció** számolás: $pozíció_t = 0.9 pozíció_{t-1} + 0.1 sorszám$



Fejlettebb EA technikák

■ Konkurens modell

- Források felosztása pozíciók alapján. Változatok:
 - Legjobb stratégia térnyerése: a legjobb kap csak forrásokat
 - Sikeres / sikertelen régiók: első 25% sikeres. Csak a sikeresek kapnak újabb forrásokat
 - Erőforrások súlyozott elosztása: pozíciók, sorszámok függvényében felosztás (minimális régióméret biztosított)
- Konkurencia ráta és periódus (migráció átfogalmazás)
konkurencia_ráta = 10 % alpopuláció méret
Konkurencia_periódus = 20% alpopuláció_méret



Fejlettebb EA technikák

■ **Kooperáló modell**

- Nagy számítási igénynél, EA helyi szélsőértékhez konvergál
- Alkalmazható:
 - Összetett problémáknál,
 - részekre bontható a probléma,
 - a megoldásrészek erősen függnék egymástól,
 - a megoldásrészek erősen befolyásolják a fitnesszt
 - Pl. fuzzy rendszer (szabályok, halmazok)
n-dim fgv. optimalizálás (dimenzióként...)



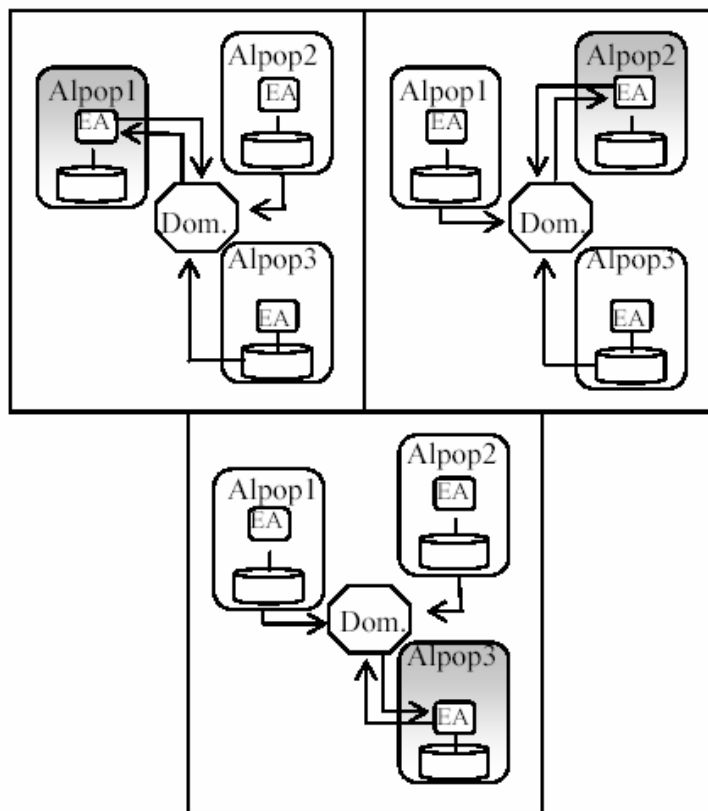
Fejlettebb EA technikák

■ **Kooperáló modell**

- A régiók kooperálnak egymással, de függetlenek
- Egy régió a megoldás egy részletét fejleszti, közös adatstruktúrában lépnek kapcsolatba
- A fitness a közös adatstruktúrán értelmezett, az együttműködés milyenségétől függ. Ehhez:
 - reprezentatív egyedek választása régióként,
 - megoldások összeállítása,
 - tesztelése
 - (minden régiónál és egyednél)

Fejlettebb EA technikák

- **Kooperáló modell** Potter et al. (2000. modellje)



Közös adatstruktúra a domain modellben

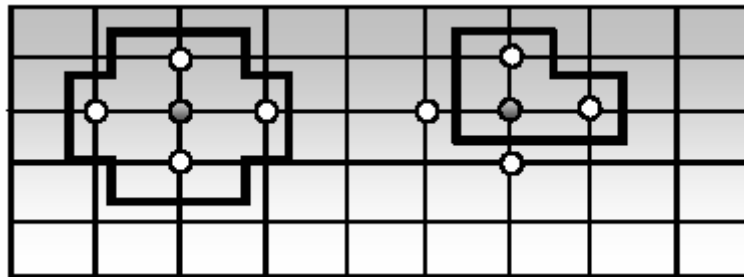
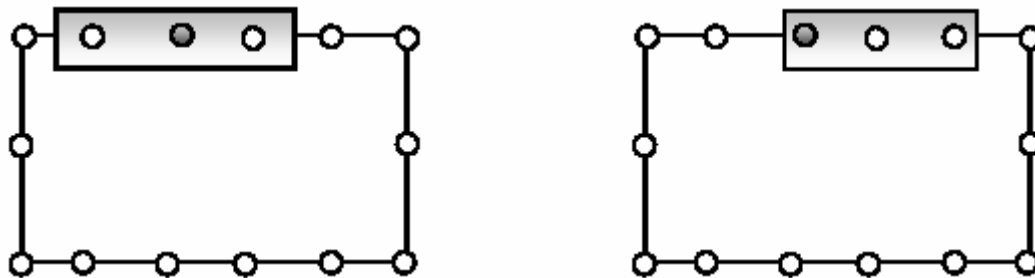


Fejlettebb EA technikák

- **Lokális modell (diffúz, cella modell)**
 - Izolált egyedek (egy-egy mikroprocesszor), csak a szomszédossal van kapcsolatuk
 - Párhuzamos számítások minden kiválasztott egyeden/környezeten
 - Alapja a topológia:
 - 1-több dimenziós, „távolság”
pl. egész/ fél gyűrű, négyzetes-átlós rácson egész/fél kereszt, csillag, kör, stb.

Fejlettebb EA technikák

- **Lokális modell** - topológiák





Fejlettebb EA technikák

■ **Lokális modell**

- Topológia jellemzők:
 - minden egyed csomópont,
 - közvetlen szomszéd távolsága 1,
 - a szomszédsági topológia az egyed/centrum körül,
 - a szomszédság méret növelhető. (2, vagy 1,2 ...)



Fejlettebb EA technikák

■ **Lokális modell** – műveletek

- Szelekció:
 - centrumok,
 - szomszédságból szülők választása (legjobb egyed / szelekció)
- Visszahelyezés: a szomszédságba.
 - utódok kiválogatása,
 - Szülők / egyedek válogatása cserére
- Modell hatékonyság:
 - jobb utódok, legrosszabb szülők válogatása visszahelyezésnél,
 - szomszédság kiterjedése: kisebb távolság jobb eredmény